Dissertation

Model-Based Leak Localization in Water Distribution Systems

Dipl.-Ing. David Bernhard Steffelbauer, BSc.

Vienna, 2018

Institute of Urban Water Management

Graz University of Technology

Supervisor/First reviewer: Assoc.-Prof. Dipl.-Ing. Dr. techn. Daniela Fuchs-Hanusch

Second reviewer: Assoc.-Prof. Dr. Ramon Pérez Magrané

To my grandmother—thanks for all the love, coffee and cigarettes.

Abstract

Fast detection and localization of leaks is of great importance to reduce losses in water distribution systems. Common leak detection and localization techniques (e.g. step-testing or acoustic logging) are labor-intensive, time-demanding and require frequent implementation. Alternatively, model-based approaches—coupling hydraulic measurement data with mathematical models of the systems—can be used to detect and locate leaks automatically. This thesis aims to develop such model-based approaches with a special focus on small leaks (<1 L/s) and is subdivided in three distinct although interconnected parts: (i) leak detection, (ii) sensor placement and (iii) leak localization.

In the first part, new leak detection methods are presented that combine time series analysis with stochastic event detection algorithms. Using these methods, real-world flow and pressure signals are deseasonalized and the remainder is modeled with low-order ARMA processes to extract trends from noise. This approach shows excellent forecast performances. Subsequently, different event detection techniques (e.g. CUSUM or Bayesian detectors) are applied on the ARMA model results. These novel methods are successful in automatically detecting a small exemplary leak within two hours—also whilst exclusively using pressure information—and are robust against false alarms. Other methods that have been developed, effectively estimate the full probability distributions of the leak's start time, magnitude and induced pressure drops—necessary for successful leak localization.

The second part addresses optimal sensor placement. The efficiency of leak localization depends on the pressure measurement positions in a system. Uncertainties such as unknown customer demand may affect those measurement locations. A new sensor placement method is presented that incorporates uncertainties to retrieve more robust measurement points. Moreover, cost-benefit analysis is performed to examine the ideal sensor numbers for specific systems. For the first time, the power-law behavior is shown for the localization performance as a function of sensor numbers . This behavior stays true under incorporation of uncertainties. Furthermore, different optimal sensor placement algorithms are benchmarked with this novel algorithm in a real-world field study.

As a third part, a new model-based leak localization algorithm—formulated as an inverse optimization problem—is presented. The fitness landscape is examined for different metrics and parameter space orderings and reveals a multi-modal nature. Yet, only meta-heuristic algorithms (e.g. Differential Evolution) reach the global optimum. The novel insights from fitness landscape analysis are used to improve the algorithm's convergence properties. Leak localization is tested in simulations as well as in a real-world case study. Throughout the case study, artificial leaks of different sizes are introduced at different locations. The developed methods effectively locate small leaks with pressure and flow measurements in the real-world. Through a two stage optimization approach based on fitness landscape insights, the algorithms additionally locate leaks using pressure information only.

In conclusion, through combining the three distinct parts presented within this thesis, an automated real-time detection and localization of leaks proved to be attainable, prospectively contributing to minimize water losses and their accompanying costs in water distribution systems.

Kurzfassung

Eine schnelles Auffinden von Lecks in Trinkwassernetzen ist von enormer Bedeutung, um Wasserverluste zu reduzieren. Weitverbreitete Techniken zur Leckdetektion und -lokalisierung (z.B. Stufentest, Geräuschlogger) sind arbeitsintensiv, zeitaufwendig und erfordern eine regelmäßige Anwendung. Eine Alternative bieten modellbasierte Ansätze, in denen die Informationen aus hydraulische Messungen mit mathematischen Modellen der Systeme kombiniert werden. Diese Techniken können ebenfalls verwendet werden, um Lecks automatisch zu finden. Die vorliegende Arbeit beabsichtigt, solche modellbasierten Ansätze, mit speziellem Fokus auf kleine Lecks (<1 L/s), zu entwickeln. Die Arbeit ist in drei eigenständige, jedoch thematisch miteinander verbundenen Teile gegliedert: (i) die Leckdetektion, (ii) die Sensorplatzierung und (iii) die Lecklokalisierung.

Im ersten Teil werden neu entwickelte Verfahren vorgestellt, die Zeitreihenanalyse mit stochastischen Ereignisdetektionsalgorithmen kombinieren, um Lecks frühzeitig zu detektieren. Druck- und Durchflussdaten werden zunächst von Saisonalitäten befreit. Der Rest wird mit ARMA Prozessen niedriger Ordnung modelliert, um Trends aus dem Rauschen zu extrahieren. Dieser Ansatz zeigt hervorragende Prognoseeigenschaften. Anschließend werden verschiedene Ereignisdetektionsalgorithmen (z. B. CUSUM- oder Bayes-Detektoren) auf die Ergebnisse des ARMA Modells angewendet. Die neuen Verfahren erwiesen sich als erfolgreich in der automatischen Detektion eines kleinen Lecks: Das Leck wurde in unter zwei Stunden entdeckt, einzig durch Informationen aus den Drucksensoren. Darüber hinaus erwies sich die Methode robust gegenüber Fehlalarmen. Andere Methoden, die zusätzlich entwickelt wurden, schätzen effektiv die vollständigen Wahrscheinlichkeitsverteilungen der Startzeit, der Größe des Lecks und der durch das Leck verursachten Druckabfälle im System ab. Das ist notwendig für eine erfolgreiche Lecklokalisierung.

Der zweite Teil befasst sich mit der optimalen Sensorplatzierung. Die Effizienz der Lecklokalisierung hängt von den Druckmesspositionen in einem System ab. Unsicherheiten, wie unbekannte Verbräuche, können diese Messorte beeinflussen. Eine neue Sensorplatzierungsmethode wird vorgestellt, die Unsicherheiten berücksichtigt und so robustere Messpositionen liefert. Darüber hinaus wird eine Kosten-Nutzen-Analyse durchgeführt, um die ideale Sensoranzahl für spezifische Systeme zu ermitteln. Zum ersten Mal wird gezeigt, dass die Qualität der Lokalisierung als Funktion der Sensoranzahl einem Potenzgesetz folgt. Dieses Verhalten bleibt auch unter Berücksichtigung von Unsicherheiten erhalten. Zusätzlich werden in einer Feldstudie verschiedene Algorithmen zur optimalen Sensorplatzierung mit diesem neuartigen Algorithmus verglichen.

Im dritten Teil wird ein neuer, modellbasierter Lecklokalisierungsalgorithmus, der als inverses Optimierungsproblem formuliert ist, vorgestellt. Die Fitnesslandschaft wird auf unterschiedliche Metriken und Parameterraumordnungen untersucht und zeigt die Multimodalität des Optimierungsproblems. Daher können nur metaheuristische Algorithmen (z. B. Differential Evolution) das globale Optimum erreichen. Die neuen Erkenntnisse aus der Analyse der Fitnesslandschaft werden verwendet, um die Konvergenzeigenschaften des Algorithmus zu verbessern. Die Lecklokalisierung wird sowohl in Simulationen als auch in einer realen Fallstudie getestet. Während der gesamten Fallstudie werden Lecks unterschiedlicher Größe an verschiedenen Orten künstlich erzeugt. Die entwickelten Methoden lokalisieren selbst kleine Lecks effektiv mit Hilfe von Druck- und Durchflussmessungen. Durch einen zweistufigen Optimierungsansatz, der auf Erkenntnissen aus der Fitnesslandschaft basiert, können die Algorithmen Lecks auch unter der alleinigen Verwendung von Druckinformationen finden.

Zusammenfassend ermöglicht die Kombination der drei Teile, die in dieser Arbeit präsentiert werden, eine automatisierte Echtzeiterkennung und -lokalisierung von Lecks. In Zukunft könnte diese Methode daher zur Minimierung von Wasserverlusten und den damit verbundenen Kosten verwendet werden.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, ______________________ ____________________________

Place, Date Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am ______________________ ____________________________

Ort, Datum Unterschrift

Contents

List of Figures

List of Tables

Chapter 1

Introduction

“Don’t panic!"

— Douglas Adams, *The Hitchhiker’s Guide to the Galaxy*

1.1. Why finding leaks is relevant all over the world

The world is steering into a freshwater crisis and the only way out is a sustainable management of our precious water resources. Across the globe, Water Utilities (WUs) face exceptional challenges (Moe and Rheingans 2006; Niemczynowicz 1999) as communities are running out of water and new resources are ill-equipped to meet rising water demands. Already today, more than two billion people live in highly water-stressed areas while growing urbanization and climate change is expected to exacerbate this tragedy (Oki and Kanae 2006). Once water is scarce, the true value of this precious good is revealed and countermeasures are taken, but unfortunately often too late. Furthermore, shortsightedly acting on water scarcity leads to unsustainable decisions, both, economically and environmentally. However, there exists an underestimated comparably easy solution: Make our water systems more efficient by reducing leakages through model-based approaches that emerged through the use of new information and communication technology in the water related sector.

Three regions on three continents facing the same problem A well-known example of a drinking water crisis is the Millennium drought in **Southern Australia** (Grant et al. 2013), which started in 1997 and lasted more than a decade. Low (below-average) rainfall in combination with high (above-average) temperatures dried out the city’s reservoirs resulting in extreme pressure on urban water supply in Southern Australia’s major cities. In the city of Melbourne, for instance, the storage volumes fell to a historic low capacity of 25.6% in 2009 during this drought (Low et al. 2015)—despite retaliatory actions increasing water supply and decreasing demand as, e.g., demand-side water restriction and voluntary conservation programs, massive introductions of greywater systems and distributed rainwater harvesting or use of recycled waste water. Concerning bad decisions, Melbourne invested approximately €4 billion in the largest desalination plant in the Southern Hemisphere—completed in 2012 which has been three years after the drought had ended (Grant et al. 2013). Because the cost and carbon footprint of water produced by desalination is significantly higher than producing water from conventional sources and nowadays in the absence of an immanent shortage of water, not a single drop of water left the brand-new plant until today. But not only in the Melbourne area, in total six major desalination plants were build during this drought in Australia (IWA 2017).

Another recent example of dry periods impacting urban water management was the drought in **California** from 2011 to 2017. During these years the state of California faced the driest period since record keeping started in the late 1800s (Hanak et al. 2015). The situation is very similar to the Southern Australia drought, although this water scarcity was amplified in California by a more dramatic population growth (Grant et al. 2013). Furthermore, the drought has been intensified by anthropogenic warming (Williams et al. 2015). Since California experienced heavy droughts in the past, heavy investments in water infrastructure were undertaken long before 2011—cushioning the impact of this drought on the major cities. Since the early 1990s, WUs invested in new interconnections that enabled sharing water supply with neighboring communities and water companies introduced more-diversified water resource portfolios (Hanak et al. 2015). "These innovations focused largely on augmenting supply and only rarely on attenuating demand or encouraging more efficient water use" (Grant et al. 2013). The drought lead to dropping reservoir levels and communities started to exploit their groundwater resources massively and in an unsustainable way. Additionally, since May 2015 a statewide urban water conservation mandate has been launched—forcing all communities to save 25 % of water compared to 2013 (Hanak et al. 2015). Although previous management decisions prevented a serious water crisis in the big cities, rural areas were hit hard by the drought as wells ran dry. Additionally, small rural poorly financed WUs—already facing water quality problems due to contaminants such as nitrate—have been further compromised by water scarcity. New pipes and deeper wells had to be constructed to limit the economic impacts of the drought (Hanak et al. 2015). But as a stopgap, bottled water had to be trucked to consumers or temporary holding tanks had to be filled.

Certainly, the most vivid example at the moment—as writing this thesis—is the **Cape Town** water crisis that began 2015. South Africa experienced between 2015 and 2017 the driest three years period since comparable record data, statistically only occurring once every 300 years (Wolski 2018). The current crisis is preceded by a strong population growth of 79 % in the last 23 years, while the reservoir capacity had been only increased by 15 % (Bohatch 2017). Currently, the government introduced restrictions on the first of February—limiting water usage per person per day to 50 liters (CNN 2018). If the conditions do not change, Cape Town will face "day zero" on August the 27^{th} when drinking water supply will be switched off (BBC 2018). From this time on residents will only be able to get their daily ration of water of 25 liters at 200 water collection points distributed over the city. This will be the first time that a major city will run out of water in the world, ever (Mulligan 2018). The proposed solution—big desalination plants—have been delayed due to bureaucracy and inefficient negotiations, leaving the City of Cape town under criticism by desalination companies (Morris 2017).

All of the above examples have in common that water scarcity was planned to be solved through a combination of (i) increasing the availability of water by exploiting new water resources which resulted in high-cost investments and (ii) painfully restricting water consumption.

Another more sustainable and often overseen solution for the water scarcity problem is increasing the water distribution efficiency by decreasing water losses. Whereas well maintained Water Distribution System (WDS) only loose 3 - 7 % of the inputed water, this number can go up to 50 % and more (Colombo, Lee, et al. 2009) in developing countries. Most of the lost water originates from leaks. That is why locating and repairing leaks in WDS is becoming highly important. Especially in South Africa, where on average 37 % of water is lost due to leaks in the distribution system (Vuuren 2014), water restrictions could have been prevented by solely reducing the amount of lost water through leaks.

A positive example of increasing efficiency—although driven by acute water scarcity and accompanied by bad management decisions—can be found in the handling of the water crisis in South-East Australia. In fact, Melbourne has shown that high pressure on water resources increases the urge for more efficiency in distribution systems as the levels of Non-revenue water (NRW) were reduced from over 40 % at the beginning of the crisis in 1995 (Scolnicov 2013) to a level of 9 % in 2015 (Leeuwen 2017), in addition, to other innovative solutions. A vivid example of the famous saying: "*No pressure, no diamonds*".

What about Europe? Europe is not an exception. In many European cities, years of stringent financial constraints on WUs, unoptimized operations and the unaffordability for WUs to maintain and replace aging infrastructure have resulted in leakage levels growing dramatically, especially in regions already under high water stress, e.g. south of Spain and Italy (González-Gómez et al. 2012). Even today, regions in Europe exist where WUs are not able to supply their customers twenty four seven—switching from continuous supplied systems to intermittent supply. Contemporary, in European countries an average of 23 % of treated water is lost in WDSs before it reaches the customer (EurEau 2017), which costs society around €80 billion per year. Certainly, this represents a big potential to use water in a more efficient and sustainable way. This fact makes leak detection and localization of high relevance even in Europe's WDSs.

It is not only about water savings Reducing leaks and, hence, water losses in WDS have additional benefits besides savings on valuable water. Especially, in countries that possess vast quantities of water (e.g. Austria), water is not the major economic driver for reducing losses. Besides saving on valuable water, lower distribution losses also result in savings on energy—used for treating and pumping water—hence, decreasing the carbon footprint of urban water systems in total. Leaks can also increase the risk of contaminant intrusion from surrounding groundwater or soil (Colombo and Karney 2002). Last, leaks harm the surrounding infrastructure—either the escaping water causes flooding of roads, houses and basements, or undermine roadways eroding the underlying soil or potentially recharge underlying aquifers in cities which poses risks to foundations of buildings (Colombo and Karney 2002). The latter impacts can easily cause damages resulting in millions of € per leak.

1.2. The language used for describing water losses

Before discussing leak detection and localization, the standard terminology for the components of the famous International Water Association (IWA) water balance are defined here following Lambert and Hirner 2000. This sets the leaks into context of other terms for describing water losses (e.g. NRW, real losses, physical losses). The water balance can be found in Figure 1.1. The **system input volume** is the starting point of the water balance and describes the volume of the water input to the transmission or WDS. It is the most important part of the water balance and, hence, has to be measured accurately. It splits up into authorized consumption and water losses. The **authorized consumption** branches out in billed and unbilled, respectively, metered or unmetered consumption. It includes—besides domestic, commercial and industrial use—the volume of water for fire fighting,

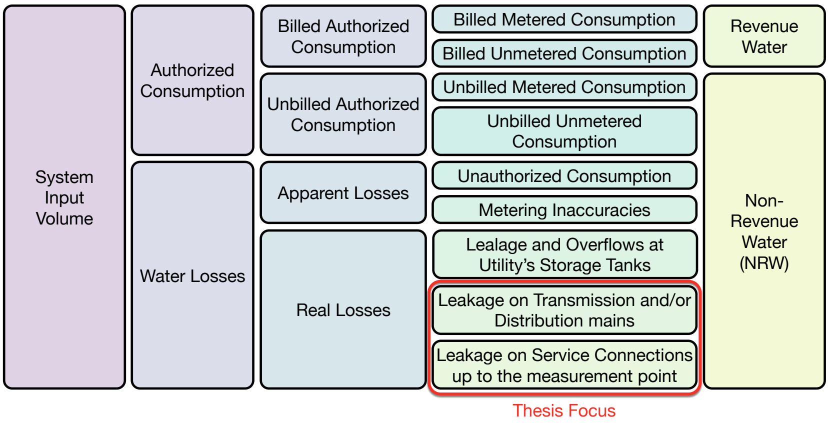

System Input Volume
Authorized Consumption
Water Losses
Billed Authorized Consumption
Unbilled Authorized Consumption
Apparent Losses
Real Losses
Billed Metered Consumption
Billed Unmetered Consumption
Unbilled Metered Consumption
Unbilled Unmetered Consumption
Unauthorized Consumption
Metering Inaccuracies
Lealage and Overflows at Utility's Storage Tanks
Leakage on Transmission and/or Distribution mains
Leakage on Service Connections up to the measurement point
Revenue Water
Non-Revenue Water (NRW)
Thesis Focus

Figure 1.1.: IWA Standard Water balance (adapted from Lambert and Hirner 2000). This thesis' focus is highlighted as red square.

frost protection, flushing of mains or sewers, gardening, water for street cleaning, building water, etc.... Which part of the authorized consumption is metered or billed relies on local practices of the WU. The **billed metered consumption** together with the **billed unmetered consumption** form the **revenue water**. **Non-revenue water** is calculated by subtracting the revenue water from the system input volume and describes the amount of water where the WU actually loses money. The **water losses** are obtained by subtracting the authorized consumption from the system input volume and can be further distinguished into real and apparent losses. **Apparent losses** contain **unauthorized consumption** (e.g. water theft or illegal use) and **metering inaccuracies**. How accurate water meters are, depends on careful meter management of the WU by, e.g., choosing the right meters, calibration of meters or the capability of the WU for detecting and changing inaccurate meters. The volume that is physically lost in the pressurized system until the customer meters form the **real losses**. From the customer meter to the end-use (e.g. water tap, toilet), the water is already billed and, therefore, losses are the customer's problem since the area of responsibility of the WU ends at the meter (at least in Austria). The real losses can be further distinguished in **leaks and overflows at tanks**, **leaks in the transmission or distribution pipes** and **leaks at the service connections** until the customer's water meter. This thesis focuses on methods to detect and locate leakages to minimize real losses on the latter two ones—on distribution mains and service connections.

1.2.1. Leak or leakage?

Different publications use different terminologies while describing similar methodologies for finding leaks. This is why the herein-used terminology is defined here at the beginning of the thesis to build a common vocabulary for preventing misunderstandings.

In Water Distribution Networks (WDNs), water is lost through holes, cracks or breaks in pipes or water escapes the pipe system through faulty joints and fittings between pipes. In many publications this is either called leak or leakage. The term **leak** is the physical hole in a pipe throughout which water is lost. While a **leakage** is the water escaping the system through leaks. Throughout this thesis, the terms leak and leakage are used synonymously since this thesis describes how to find leaks by the physical effects that the lost water causes (e.g. dropping pressure or increasing inflow). Furthermore, no distinction is made between holes, cracks, faulty joints, etc., since it is not relevant how the water is lost[1]for the mathematical methods described in this thesis, only the location and the amount are of interest. Consequently, only the terms leak and leakage are used.

Regularly, there exists some confusion about leak/leakage detection, location, localization, awareness and pinpointing. Again, this terms are used in different ways in different publications. In this thesis, leak **detection** refers the task of realizing that a leak occurred in the system without finding its location. It describes methods for detecting leaks in inflow and/or pressure data by, for example, time series analysis techniques as described in Chapter 2.

Leak **awareness** is the act of becoming conscious that there is one or more leaks in the system. This can be triggered either through analyzing the water balance, Minimum Night Flow (MNF) analysis or leak detection methods.

After a leak is detected, the location of the leak has to be found. Leak **localization** refers to the activity of finding the approximate location of a leak in a certain area. Finding the exact location of a leak is called leak **pinpointing**. Leak localization makes leak pinpointing easier since not the entire system where a leak is detected has to be searched, only the part of the network that has been identified by the localization technique.

Additionally, it has to be mentioned that, in general, there is a difference between WDS and WDN. The term **WDN** describes solely the pipe network itself. A WDN together with pumps, tanks and reservoirs builds the **WDS** in its entirety.

1.2.2. Detectable leaks and their run-times

The real losses through leaks described in Section 1.2 can be further subdivided in three categories (see Figure 1.2): (i) reported leaks, (ii) detectable unreported leaks and (iii) undetectable background leaks (see Thornton et al. 2008, for a more detailed description).

(i) **Reported leaks**: High flow rates are characteristic for reported breaks and leakages due to a large and fast disruptive event (Thornton et al. 2008). In general, they are reported after short time by customers or WU employees, as they cause high pressure drops, supply interruptions or a large amount of water suddenly being visible at the surface.

(ii) **Detectable unreported leaks**: These leaks can only be found if utilities actively search for leaks, since they are hidden under the surface and, hence, result in long run times. In general, they have moderate and smaller pressure drops and leakage outflows than reported leaks. In addition, they do not lead to supply interruptions, otherwise they will be reported. They are detected through their physical effects, either flow and/or pressure measurements or through the noise they are causing with acoustic leak localization and pinpointing equipment.

(iii) **Undetectable background leakages**: Background losses are very small leakages smaller than ≈ 0.07 (L/s) (Thornton et al. 2008) that occur at pipe fittings or joints. They are too small to be detected, because they do not have measurable physical effects. For that reason, they run for a long time. They grow over time by incrementally deteriorating the pipe and, thus, become detectable at some future point. The loss through background leakages can only be minimized by pressure management or replacing the affected infrastructure.

[1] It has to be noted that the shape and type of the leak actually influences the emitter exponent described in equation (2.63) introduced in Section 2.2—see for example Fuchs-Hanusch, Steffelbauer, Günther, and Muschalla 2015.

The losses through reported leaks and bursts—although spectacular when a huge amount of water is shooting out a big diameter pipe at high pressures—keep within limits. Due to their damage-causing nature (Thornton et al. 2008), they are detected, located and repaired quickly[2]. In contrast to common sense, most water in WDSs is lost due to unreported leaks and background leakages, because of their long run-times. For example, a leak with an leakage outflow of a quarter liter per second leads to water losses of more than 20 cubic meters a day.

Therefore, this thesis focuses on the detection and localization of unreported leakages, however, the developed techniques are also applicable to detect and locate bigger pipe bursts more quickly. Background leakages remain not detectable with the methods described in this thesis since they have no or too small physical effects to be measurable, according to their definition.

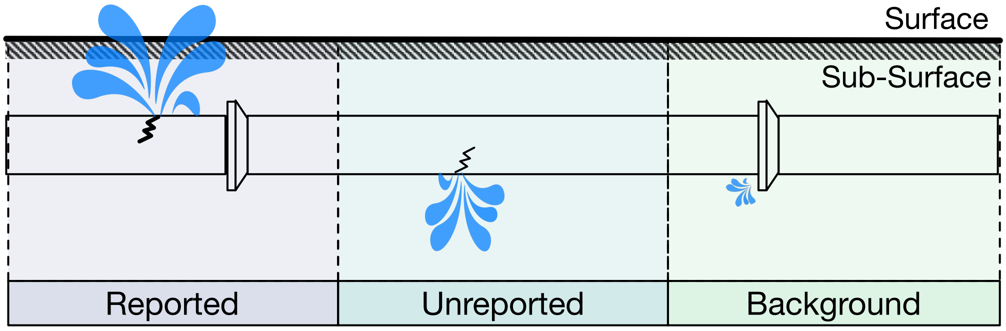


Figure 1.2.: Components of real losses (adapted from Thornton et al. 2008).

The leakage run-time is the overall amount of time that water is lost through a leak, starting when the leak occurs and ending when the damage in the system is fixed. The runtime of particular leak types (reported, unreported, background) is different and depends on the size of the leak. Smaller leaks are harder to detect and have usually longer leakage run-times. Big detectable leaks and pipe bursts tend to be visible on the surface and can lead to supply disruptions, thus, are generally reported faster. Due to the supply disruptions, WUs additionally tend to repair this leaks faster than smaller leaks.

The run-time of a leak is composed of the (i) awareness time, the (ii) localization time and the (iii) repair time (see for example Thornton et al. 2008).

(i) **Awareness time**: This is the time it takes the WU to become aware of the leak in their WDS. This strongly depends on the size of the leak. Smaller leaks are harder to become aware of. Becoming aware of unreported leaks is only possible, if the WU conducts active leakage control methods.

(ii) **Localization time**: Once the WU is aware of the leak, they have to localize it in their WDS. The localization time is the time it takes to find the leak. Besides the size of the leak, this time also depends on the tools and skills of the WU's personnel in finding leaks. Additionally, the time for leak-pinpointing is contained within this time.

(iii) **Repair time**: The time it takes for the WU's personnel to repair an already located leak. This depends on a lot of factors which are not elaborated here in detail since this thesis emphasizes on detecting and locating leaks. (For the interested reader chapter 17 of Thornton et al. 2008, is recommended).

[2] Usually, they are reported to the WU by customers with the words: "*No pressure, no water*" in contrast to the famous saying "No pressure, no diamonds" from before.

This thesis aims to develop methods to reduce the overall run-time of leaks by (i) automatically detecting leaks faster and (ii) automatically localizing these leaks[3]. The developed methods for detecting leaks reduce the awareness time, the methods for localizing the leak reduce the localization time.

The total water loss through a leak is the leakage outflow multiplied by the leak's run-time. Thus, reducing the run-time of leaks reduces the water loss, as can be seen graphically in Figure 1.3.

Regarding leakages, another important time is crucial—the time of MNF. In general, the MNF is the flow between 2:00 and 4:00 in the morning (according to Puust et al. 2010)[4]. This time is important for most leakage localization techniques, either focusing on the inflow in the system (e.g. step-testing) since customers are asleep and the noise in the inflow measurements are minimal, or for acoustic measurement devices the interfering noise of traffic is minimal during the MNF time.

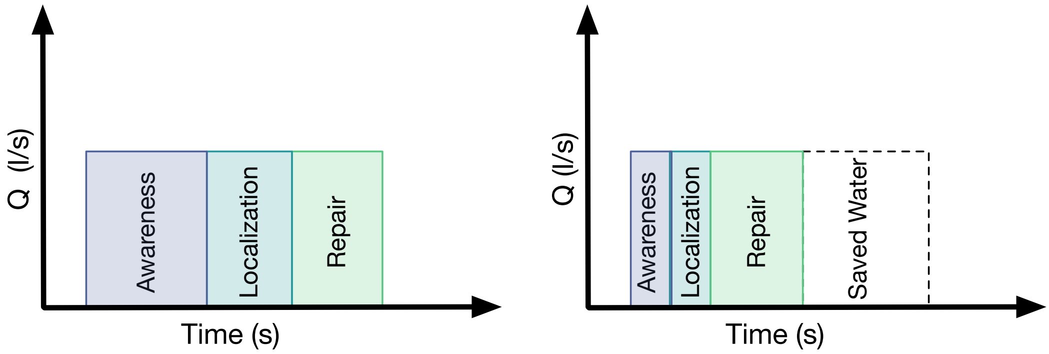


Figure 1.3.: Components of leak run-times: awareness-, localization- and repair-time (based on Thornton et al. 2008). Left: leak localization with conventional methods. Right: reduced awareness and localization times due to model-based approaches results in savings of valuable water.

1.3. State of the art methods for finding leaks

1.3.1. Physical effects of leaks

Leaks cause several physical effects. For example, they result in (i) increasing **inflow**, (ii) dropping **pressure** and (iii) acoustic **noise**[5].

(i) WDNs are pressurized[6] and all the water that escapes the system through the leak, has to additionally flow into the system. This results in an increasing **inflow** in the part of the WDS that is affected by the leak as a consequence of the flow conservation law [7] (see Chapter 2 Section 2.2).

[3] . . . in the sense of getting the approximate area where the leak occurs, not the exact location.

[4] This time can vary in different countries, or in rural and urban areas according to inhabitant's habits and should be determined by measurements prior to leak localization.

[5] Leaks are also increasing the humidity of the surrounding soil, influence the grounds temperature, etc . . . , but these effects are seldom used for locating leaks.

[6] Not in intermittent-supply-systems or during big failures.

[7] . . . and if the fluid is assumed to be incompressible

(ii) The increase in flow through the leak leads to higher velocities of water in parts of the system which are upstream to the leak. Higher velocities result in higher friction losses due to the non-linear friction laws in pressurized pipe systems (see again Section 2.2). This friction losses manifest themselves in **pressure** drops. Additionally, sudden breaks result in transient pressure waves (transients) traveling through the system until they dissipate their kinetic energy and are completely damped.

(iii) The water escaping the pressurized pipe system cause a whistling **noise** transferred on the pipe wall respectively the medium itself. The distance over which this noise is measurable depends on the pipe material.

All of these physical effects can be used to localize and pinpoint leaks.

1.3.2. Leakage control methods

Generally, water utilities deal with leakages through two contrary leakage control methods (according to Puust et al. 2010): (i) passive and (ii) active leakage control:

(i) In the **passive** approach, utilities wait until customers report leaks due to supply problems or leaks becoming visible at the surface. In most of the cases, as soon as leaks become visible, serious damage has already occurred at the surrounding infrastructure. Yet, not all leakages appear at the surface. This leads to unreported and consequently high total water losses.

(ii) In contrast, **active** leakage control is characterized by regularly examining the distribution system for leakages with the purpose to drastically reduce the time between leak occurrence, awareness and repair.

Consequently, active leakage control results in much lower water losses than passive leakage control (M. Farley and Trow 2003). Accordingly, its application is strongly recommended by, for example, the IWA and the American Society of Civil Engineers (ASCE).

1.3.3. Leak localization methods

At this point, different state-of-the-art leak localization methods are compared based on the more extensive review that can be found in Puust et al. 2010.

Most active leakage control methods have in common that they have to be applied on the whole WDN on a regular basis. One of the most widespread active methods is performing regular district audits with manual **listening sticks** by utilizing the noise effect of leaks. Although these devices seem simple, their handling requires a lot of skilled, trained and highly concentrated staff. More advanced electronically amplified listening sticks exist which have the advantage of lowering the demanded skills of their operators. This technique is very labor intensive, because WDNs of big cities easily contain thousands kilometers of water distribution pipes. Furthermore, such audits can only be performed during night time, when surrounding noises are minimal.

Another technique making use of leak noise is **acoustic logging**. Acoustic sensors are temporarily or permanently attached (e.g. magnetically) every few hundred meters to pipe fittings. During the nighttime, they continuously measure and store sound levels. Subsequently, the data is collected and analyzed on a regular basis. If a leak occurs, sound levels rise. Drawbacks of this method are the high number of sensors that have to be deployed and material dependency as noise is transmitted poorly through plastic pipes. Added to this, many acoustic techniques are insensitive to large leaks as they do not generate vibrations in the expected high frequency domains (Colombo, Lee, et al. 2009).

Many WUs use **step testing**, a renowned, material independent and effective but also labor intensive localization method which is not applicable to all networks. Step testing is also performed during MNF similar to the previous technique. During the night, valves are closed systematically, subdividing a monitored zone in subareas. The inflow into the zone is observed. If the inflow decreases while a certain sub-area is isolated, a leak might be situated in this part of the system. The drawbacks of this methods are that parts of the WDN are not pressurized, resulting in possible contamination (e.g. back-siphonage, groundwater intrusion).

Ground penetrating radar is a novel method which allows a non-intrusive leak inspection. Leaks are detected with radar by finding voids excavated by circulating water running out of the pipe or by finding relocated pipes. This method is not material dependent, but penetration depths of only two meters in the ground make it not suitable for finding leaks in Northern European countries with cold climate where water pipes are laid in depths of 1.8 meters and more. Furthermore, it is also labor intensive since all pipes in the network have to be checked individually. Additionally, anomalies like under-ground metal objects may lead to false conclusions.

WU's **Supervisory Control and Data Acquisition (SCADA)** systems can also be used for leakage awareness and localization. The inflow in a subsystem of the WDS is observed permanently and analyzed on a regular basis. If the flow increases over a longer time frame—especially during MNF hours—a leak may have occurred in the observed area.

All the here mentioned techniques share[8] that they yield only an approximate area where a leak occurred and not the exact position. Since excavating pipes and resurfacing streets after leak repairs are expensive, mistakes in locating leaks potentially cause avoidable interruptions. That is why leakage localization does only the groundwork for pinpointing techniques by narrowing down the search area. The exact location of a leak is only found by leak pinpointing techniques.

1.3.4. Leak pinpointing methods

Again, this is a brief overview of possible leak pinpointing techniques based on the more extensive review of Puust et al. 2010. Leak pinpointing techniques are the most accurate techniques to obtain leak locations in a pipe system, but come with high financial costs and are very labor intensive. In general, leak pinpointing methods are distinguished between invasive and non-invasive techniques.

First, the most common non-invasive technique to pinpoint leaks since the late 1970s is the application of **leak noise correlators**. Two microphones are brought in contact with the pipe (or a valve) surrounding the assumed leak spot. The noise signal from the leak is then correlated between the two microphones and from the difference in the travel time of the sound waves of the noise, the leak's positions is computed with an accuracy of up to one meter. Again, leaks in PVC or PE pipes are harder to locate than leaks in metallic pipes due to the noise damping properties of plastics.

Second, the exact position of a leak can be found by the invasive intrusion of **tracer gas**. This gas must have certain properties, e.g., lighter-than-air, non-toxic, non-soluble in water. At a leak, the gas escapes the pipe system and can be measured with special probes on the surface above the leak. Additionally—contrary to leak noise correlators—multiple leaks can be found at a single pipe section or branched pipes.

[8] Besides ground microphones and ground penetrating radar which can also be used for pinpointing.

Third, so called **pig-mounted sensing** devices can be inserted under pressure into the pipe system. These devices flow with the water stream, continuously measuring and transmitting physical quantities as well as their current position. The devices can be equipped with ultrasonic sensors, hydrophones, magnetic flux sensors, etc. ..., measuring quantities which enable to draw conclusions about the pipe's condition.

In general, leak pinpointing methods are labor intensive and/or expensive and should only be conducted after previous application of leak localization—after the search area is already narrowed down.

1.4. Why model-based leak localization

In general, all available techniques for active leakage control share that they are (i) expensive, (ii) time consuming, (iii) tedious resulting in long detection and localization times, (iv) only applicable in the middle of the night, (v) material dependent and working poorly especially in systems consisting of plastic pipes or arbitrary combinations of all mentioned drawbacks (Puust et al. 2010). Whereas the main drawback for WUs is certainly that these techniques are labor intensive and hence expensive.

Alternatively, one can take advantage of the hydraulic responses of pipe systems to leaks. A damaged pipe in a network results in leakage outflow where water escapes the pipe system changing the flow characteristics—increasing flow and decreasing pressure—in the network (Poulakis et al. 2003). Not surprisingly, it is possible to detect leaks and determine their positions in a WDS by these hydraulic phenomena which they are causing (Colombo, Lee, et al. 2009). Prerequisites for achieving this goal are that (i) hydraulic measurements are available in the network and (ii) an accurate hydraulic model exists of the system under observation.

The intensity of hydraulic system changes caused by a leak at specific measurement positions depends on two entities—the location and the outflow magnitude (Poulakis et al. 2003). If a leak causes a clear and unique footprint in flow and pressure measurements, it is theoretically possible to locate it (Colombo, Lee, et al. 2009). Permanent monitoring of flow and pressure in a system enables detecting the occurrences of leaks (Poulakis et al. 2003). Once a leak is detected, the hydraulic model can be updated through fitting the new measurement data under the leak's influence to draw conclusions about the leak's location. This is the basic principle of model-based leak localization.

Beneficial for a model-based leak localization approach is certainly the possibility to work in a complete automated way, alarming WU staff when a leak occurs in the system and already giving them the approximate location of the leak, enabling to directly start with leak pinpointing. This has the potential to drastically shorten the leak awareness and the leak localization time compared to state-of-the-art methods, reducing the leakage run-time and, hence, minimizing water losses.

Especially nowadays—as sensor and real-time data communication technologies are increasingly more affordable—the latest developments in sensor technology will encourage WUs to install more sensors in their WDSs in near future. This will be additionally in favor of model-based approaches. In more detail, it is envisaged that implementing an increased number of pressure sensors would be the water companies' preferred choice (Romano et al. 2013, according to). Reasons for that are that pressure sensors have lower installation costs and less maintenance effort is needed compared to flow sensors. Thus, technologies utilizing pressure sensors such as model-based leak localization with hydraulic sensors will play an important role in near real-time operational optimization of WDS in future (Romano et al. 2013).

The biggest competitor to leak localization with hydraulic sensors is without doubt acoustic logging. While the costs and data communication effort is similar to pressure sensors, the measurement grid of these sensors has to be more dense since sensors have to be placed every few hundred meters. This leads to high sensors numbers, especially, if one wants to detect all possible leaks in a system. Additionally, more and more WUs—particularly in rural areas—favor plastic materials for their new pipes. These materials are the Achilles heel for acoustic methods since the propagation of sonic waves is strongly damped leading to even higher sensor numbers. That is why acoustic loggers will lose out in favor of pressure and flow sensors in plastic pipe systems. Whereas no additional information can be retrieved from acoustic sensors—since they only contain indirect system information which is not relevant for WDS operation besides finding leaks—hydraulic sensors have further advantages and fields of application, e.g., optimization of network operations, hydraulic network calibration, detect closed valves, retrieving important information for future expansion and rehabilitation plans.

All this mentioned benefits predict model-based leak localization a bright future even though they are at the moment a subject of research and not yet ready for the market.

1.5. Objectives and research questions

This is a short and condense summary of the motivation for the research questions. A broader motivation of the questions can be found in the corresponding chapters.

Model-based approaches already exist in literature to detect and locate leaks, but, first, they are seldom applied on real-world studies. Second, if they are tested in the real-world, they are studied most of the time on huge pipe bursts of several liters per second. These bursts are easier to detect than small leaks with outflows lower than one liter per second. Third, most of the time the approaches utilize flow sensors only to localize leaks and it is stated as a great success if the method can identify the right DMA where a leak has happened. Fourth, if pressure sensors are involved for leak localization, then an uneconomically high number of sensors is used.

To bring model-based leak localization to everyday WU practice, these techniques have to be able to find smaller leaks in the real-world. Additionally, to actually reduce water losses, these techniques have to even detect and locate these small leaks faster, and, definitely cheaper than already used common techniques. And that all, of course, with less overhead for WU's staff than today's already successful applied techniques. Only this will increase their acceptance and decrease the barriers in the minds of the operators which always exist when applying new technologies.

This thesis will try to develop a methodology with such aims by dealing with three interconnected parts important for model-based water loss reduction: (1) Leak Detection, (2) Optimal Sensor Placement and the (3) Leak Localization problem itself.

(1) Leak Detection Seasonality effects and noise surmounting tiny pressure deviations resulting from small leaks in the system make leak detection a challenging problem. Additionally, pressure data is considered less reliable and often played the poor cousin to flow measurement data (B. Farley et al. 2010). Of course, statistical methods have to be used to retrieve appropriate information from small signals in the noisy environment of WDS, a problem similar to search for a needle in the haystack. All of that leads to the first research question:

Q.1.1: Is it possible to early detect small leaks (<1 L/s) in the real world with pressure sensors in an automatic way?

A question that has not been answered in the scientific literature till now.

(2) Optimal Sensor Placement The leak detection and localization performance depends on the positions where sensors are located in a WDS. In literature, there exist a few algorithms of how to place sensors in an optimal way. Every single method claims to find the optimal positions. Nevertheless, applying different algorithms on the same network leads to diverging sensor positions. The algorithms have never been benchmarked neither in simulations nor in the real-world. Hence, the second research questions is obvious

Q.2.1: Which optimal sensor placement algorithm finds the best pressure sensor positions for model-based leak localization?

Furthermore, uncertainties, e.g. uncertain water demand, may affect pressure measurements. Since customer demands affect the system in the same way as leaks, a leak might be overseen because of fluctuations in demand. That is why sensor locations with a high demand noise might be less ideal than other positions.

Q.2.2: How to incorporate different sources of uncertainties in sensor placement algorithms to obtain more robust optimal measurement locations?

Finding the right positions is one thing, but another interesting question arises especially for WUs which want to apply these techniques in their systems. (This question is also closely related to the costs of applying model-based leakage localization)

Q.2.3: How many sensors are needed for model-based leak localization?

(3) Leak Localization Model-based leak localization formulated as inverse problems solved with heuristic optimization algorithms has already been studied in scientific literature. Since heuristic algorithms are less efficient than deterministic algorithms to solve unimodal optimization problems, but fail in finding optimal solutions in multi-modal optimization problems, it is astonishing that the form of the optimization problem itself has never been examined in any scientific study to my knowledge.

Q.3.1: What is the actual type (form,shape) of the optimization problem?

Are heuristic optimization algorithms even necessary for finding leaks? Also related to an exhaustive study of the optimization problem is the question

Q.3.2: How to increase the performance of model-based leak localization?

Small leaks lead to small pressure signals which are already hard to detect. Are the differences in the pressures big enough to distinguish between different leak positions for small leaks? Again it has to be noted that former studies have applied localization on big leaks mostly simulations and seldom in reality. It is the aspiration of this thesis to actually address real-world problems which leads to the final research question

Q.3.3: Is model-based leak localization applicable for finding small leaks in the real-world?

Summarizing all the research questions in a single sentence, the objective of this thesis is best described through:

Finding smaller leaks with less and cheaper sensors automatically within a real-world DMA.

1.6. Overview of the thesis

A short overview of the thesis is given in this section consisting of short descriptions of the single chapters and their content.

After the **Introduction** of the thesis in this chapter, **Background** information—mainly consisting of the methods that are used throughout this thesis—is given. The Background chapter is divided in four main parts: (i) Optimization, (ii) Hydraulic simulations, (iii) Stochastics and (iv) the used Case Studies. (i) Optimization is a big part of this thesis used in almost every subsequent chapter. Terminologies used in mathematical optimization will be clarified (e.g. the difference between local and global optima). Since this thesis makes mainly use of meta-heuristic algorithms, like e.g Genetic Algorithm (GA) or Differential Evolution (DE), the difference between stochastic and deterministic algorithms will be highlighted. For that reason, different deterministic and stochastic optimization algorithms will be introduced and their field of application will be explained by applying these algorithms on examples from literature. (ii) The section on hydraulic simulations will give an overview of different hydraulic simulation approaches followed by a motivation, why a specific approach in this thesis is chosen for the purpose of model-based leak localization. (iii) The comprehensive although short section on stochastics will introduce different algorithms for different purposes. For example, Monte Carlo Simulations (MCS) algorithms for computing uncertainty effects, Markov-Chain-Monte-Carlo (MCMC) for probabilistic sampling, outlier detection with Tukey's test, Goodness-of-Fit (GoF) statistics for deciding between different modeling functions. Since model-based leak localization will be applied on real-world data, a big part will address time series analysis techniques, followed by an introduction on different event detection algorithms. (iv) Finally, the chapter will conclude with an overview of the three case study networks used throughout the thesis. Two networks are just used in simulations, but one network additionally serves as a real-world field study. Hence, also the measurements taken in this network are described and the network's calibration will be explained.

Then the main three chapters of this thesis will be introduced: Leak Detection, Optimal Sensor Placement and Model-based Leak Localization. Each chapter is more or less independent of each other, although, they are all interconnected. However, each of these three chapters share the same skeleton. It will start with a comprehensive literature review on the specific topic presenting the state-of-the-art, followed by a methodology section presenting all the methods which have not already been described in the Background chapter. Then the chapter's results are presented and it ends with a conclusion.

Chapter 3 tinkers the topic of **Leak Detection**. The methodology of this chapter is mainly a back reference since most of the methods on time series analysis and detection algorithms have been already introduced in the Background chapter. The results section starts with methods for extracting seasonalities from water related data. Additionally, a method for identifying days with abnormal water use will be introduced which can be also used for leak detection. Flow and pressure data will be analyzed to retrieve better models step-by-step by removing seasonalities and random fluctuations. The resulting models are then used to forecast unknown near-future states of the system. The purpose of this is that the better the forecast is—the better unusual events can be detected. This is the principle of leak detection. Subsequently, different event detection algorithms will be applied on the prior developed time series models on flow as well as pressure measurement data. Methods will be introduced that are able (i) to detect leaks very fast (also in pressure data) and subsequently raise a leak alarm, (ii) to estimate the exact time when the leak has appeared in the system afterwards and (iii) to retrieve the complete probability distributions and likelihoods of the leak events for all parameters. All presented methods are designed to work in a complete automatic way.

Chapter 4 deals with the **Optimal Sensor Placement** problem, since the success of model-based leak localization as well as leak detection depends on the measurement positions in a WDS. Different optimal sensor placement approaches are introduced which are benchmarked afterwards. A novel approach to place sensors that is also able to incorporate uncertainties will be introduced. Additionally, a method to enable cost-benefit analysis in a specific network is presented.

Chapter 5 is a chapter on **Model-Based Leak Localization**. An optimization capable of finding leaks will be introduced. Then the leak localization problem is studied in more detail, by examining the optimization problem itself through fitness landscape analysis. Different influences will be studied concerning the optimization problem, like the consequences of reducing the number of sensors, using different metrics in the optimization problem or changing the problem itself by rearranging the search space. All of this has influences on the convergence of the optimization which will be investigated in detail. Subsequently, the introduced localization algorithms are applied on real leaks introduced in a system during a field study. The influence of the sensor positions will be analyzed in the real-world as well as the performance of leak localization on small leaks. Furthermore, a method will be introduced that is capable of finding leaks by only using information from pressure sensors. Finally, the leak localization right after leak detection will be undertaken for testing if the developed methods are capable of actually saving water in the real-world by reducing leak detection and localization times.

The thesis will conclude by summarizing all important methods and findings and discussing their applicability in the real-world in a short **Conclusion** chapter. Moreover, possibly future research directions will be pointed out.

1.7. List of scientific publications and presentations related to this thesis

The author has published and contributed to research related to this thesis in multiple journals and conference proceedings. Here is a list of published work that influenced this thesis or was derived during the development of it:

Related Journal Papers

- D. B. Steffelbauer and D. Fuchs-Hanusch (2016a). «Efficient Sensor Placement for Leak Localization Considering Uncertainties». In: *Water Resources Management* 30.14, pp. 5517–5533. DOI: 10.1007/s11269-016-1504-6
- D. Fuchs-Hanusch, D. B. Steffelbauer, M. Günther, and D. Muschalla (2015). «Systematic material and crack type specific pipe burst outflow simulations by means of EPANET2». In: *Urban Water Journal*, pp. 1–11. DOI: 10.1080/1573062X.2014.994006
- D. Fuchs-Hanusch, D. B. Steffelbauer, and M. Günther (2015). «Drucksensoren zur Leckagelokalisierung - Optimale Platzierung der Sensoren durch Berücksichtigung von Verbraucherunsicherheiten». In: *Aqua & Gas* 11, pp. 26–31

Related Conference Publications (First Author)

- D. B. Steffelbauer, M. Günther, and D. Fuchs-Hanusch (2017). «Leakage Localization with Differential Evolution: A Closer Look on Distance Metrics». In: *Procedia Engineering* 186, pp. 444–451. DOI: `10.1016/j.proeng.2017.03.251`
- D. B. Steffelbauer and D. Fuchs-Hanusch (2016b). «Fitness landscapes and distance metrics for model-based leakage localization». In: 3rd International Conference on Control and Fault-Tolerant Systems, SysTol'16. Barcelona, Spain
- D. B. Steffelbauer and D. Fuchs-Hanusch (2015). «OOPNET: An object-oriented EPANET in Python». In: *Procedia Engineering*. Computing and Control for the Water Industry (CCWI2015) Sharing the best practice in water management 119, pp. 710–718. DOI: `10.1016/j.proeng.2015.08.924`
- D. B. Steffelbauer, M. Neumayer, M. Günther, and D. Fuchs-Hanusch (2014). «Sensor Placement and Leakage Localization Considering Demand Uncertainties». In: *Procedia Engineering*. 16th Water Distribution System Analysis Conference, WDSA2014 Urban Water Hydroinformatics and Strategic Planning 89, pp. 1160–1167. DOI: `10.1016/j.proeng.2014.11.242`
- D. B. Steffelbauer, M. Günther, M. Neumayer, and D. Fuchs-Hanusch (2014a). «Leakage Localization In Virtual District Metered Areas With Differential Evolution». In: *International Conference on Hydroinformatics*
- D. B. Steffelbauer, M. Günther, M. Neumayer, and D. Fuchs-Hanusch (2014b). «Sensor Placement and Leakage Isolation with Differential Evolution». In: *World Environmental and Water Resources Congress 2014*. American Society of Civil Engineers, pp. 408–416

Related Conference Publications (Co-Author)

- M. Cattani, C. A. Boano, D. B. Steffelbauer, S. Kaltenbacher, M. Günther, K. Römer, D. Fuchs-Hanusch, and M. Horn (2017). «Adige: An Efficient Smart Water Network Based on Long-range Wireless Technology». In: *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*. CySWATER '17. New York, NY, USA: ACM, pp. 3–6. DOI: `10.1145/3055366.3055367`
- D. Fuchs-Hanusch and D. B. Steffelbauer (2017). «Real-world Comparison of Sensor Placement Algorithms for Leakage Localization». In: *Procedia Engineering*. XVIII International Conference on Water Distribution Systems, WDSA2016 186, pp. 499–505. DOI: `10.1016/j.proeng.2017.03.262`
- S. Kaltenbacher, D. B. Steffelbauer, M. Cattani, D. Fuchs-Hanusch, K. Römer, and M. Horn (2017). «A Dynamic Model for Smart Water Distribution Networks». In: pp. 1–8
- M. Günther, D. B. Steffelbauer, and D. Fuchs-Hanusch (2016). «Fault detection data creation using an experimental water distribution system». In: *Proceedings of the 2016 3rd Conference on Control and Fault-Tolerant Systems*. Systol. Barcelona, Spain: IEEE, pp. 331–336
- M. Günther, D. Camhy, D. B. Steffelbauer, M. Neumayer, and D. Fuchs-Hanusch (2015). «Showcasing a Smart Water Network Based on an Experimental Water Distribution System». In: *Computing and Control for the Water Industry (CCWI2015) Sharing the best practice in water management* 119, pp. 450–457. DOI: `10.1016/j.proeng.2015.08.857`
- M. Günther, D. B. Steffelbauer, M. Neumayer, and D. Fuchs-Hanusch (2014). «Experimental Setup to Examine Leakage Outflow in a Scaled Water Distribution Network». In: *Procedia*

Engineering. 16th Water Distribution System Analysis Conference, WDSA2014 Urban Water Hydroinformatics and Strategic Planning 89, pp. 311–317. DOI: `10.1016/j.proeng.2014.11.193`

- M. Neumayer, D. B. Steffelbauer, M. Günther, and D. Fuchs-Hanusch (2014). «Computational efficient small signal model for fast hydraulic simulations». In: 11th International Conference on Hydroinformatics. New York City, USA

Other Publications (not directly related to this thesis) Additionally, the author contributed in scientific publications that were not directly related with this thesis, but—for the sake of completeness—are listed here:

- J. Leimgruber, D. B. Steffelbauer, G. Krebs, and D. Muschalla (2017). «Back to the roots: A storm event-based assessment of LID performance to restore the natural water balance». In: ICUD 2017. Prague
- R. Maier, T. Hofer, D. B. Steffelbauer, G. Gruber, and D. Muschalla (2015). «Integrated real-time control of a central storage tunnel for minimization of sewer emissions in Graz, Austria». In: EJSW 2015
- A. Pichler, D. B. Steffelbauer, and A. Nazarov (2014). «Examples for Genetic Algorithm based optimal RFID tag antenna design». In: *2014 IEEE RFID Technology and Applications Conference (RFID-TA)*. 2014 IEEE RFID Technology and Applications Conference (RFID-TA), pp. 223–227. DOI: `10.1109/RFID-TA.2014.6934232`
- Camhy D., Gruber G., Steffelbauer D. B., Hofer T. F., and Muschalla D. (2014). «OpenSDM - An Open Sensor Data Management Tool». In: *11th International Conference on Hydroinformatics*. HIC
- D. Camhy, V. Gamerith, D. B. Steffelbauer, D. Muschalla, and G. Gruber (2012). «Scientific Data Management with Open Source Tools – An Urban Drainage Example». In: *9th International Conference on Urban Drainage Modelling (UDM)*. ed. by J. Plavšić. Belgrade, Serbia: Faculty of Civil Engineering, University of Belgrade
- G. Gruber, V. Gamerith, J. Olsson, D. Camhy, D. B. Steffelbauer, M. Hochedlinger, P. Dihé, S. Schlobinski, and L. Gidhagen (2012). «SUDPLAN: Developing a Decision Support System to Cope with Climate Change – Urban Drainage Pilot Linz». In: *IWA World Water Congress*. Busan, Corea

List of Presentations at International Scientific Conferences The author presented scientific work related to this thesis at various international scientific conferences. Here is a list of the presentations:

Conference	**Title of presentation** and long name of conference
SysTol 2016	**Fitness Landscapes and Distance Metrics for Model-Based Leakage Localization** 3rd Conference on Control and Fault-Tolerant Systems, 7-9 Sept. 2016, Barcelona, Spain.
WDSA 2016	**Leakage Localization with Differential Evolution: A Closer Look on Distance Metrics** 18th International Conference on Water Distribution System Analysis, 24-28 July 2016, Cartagena, Colombia.
CCWI 2015	**OOPNET- An object-oriented EPANET in Python** 13th-International Conference - Computing and Control for the Water Industry, 2-4 Sept. 2015, Leicester, UK.
HIC 2014	**Leakage Localization in DMAs with Differential Evolution** and **OpenSDM - An Open Sensor Data Management Tool** 11th International Conference on Hydroinformatics, 17-21 August 2014, New York, USA.
WDSA 2014	**Sensor Placement and Leakage Localization considering Demand Uncertainties** 16th International Conference on Water Distribution System Analysis, 14-17 July 2014, Bari, Italy.

Chapter 2

Background

“All you really need to know for the moment is that the universe is a lot more complicated than you might think, even if you start from a position of thinking it’s pretty damn complicated in the first place."

— Douglas Adams, *Mostly Harmless*

2.1. Optimization

2.1.1. Mathematical definition of the problem

Mathematical optimization[9] has various real-world applications in economics, engineering, decision making, to name just a few fields. Mathematically, an optimization problem can be defined by finding the minimum of an objective function (also called fitness function) in following way

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \quad \ni \quad \begin{cases} c_i(\boldsymbol{x}) = 0, & i \in \mathcal{E} \\ c_i(\boldsymbol{x}) \geq 0, & i \in \mathcal{I} \end{cases} \tag{2.1}$$

where $\boldsymbol{x}$ is a n dimensional parameter vector of real numbers ($\boldsymbol{x} \in \mathbb{R}^n$), called the trial or optimization variables, f is the (scalar) objective function that one wants to minimize, c_i are (non-)linear constraint functions that the parameters $\boldsymbol{x}$ have to satisfy. $\ni$ is a mathematical symbol with meaning "such that" or "subject to". The constraints c_i can be equality constraints, where $\mathcal{E}$ is the set of indices belonging to this equality constraints, or inequality constraints represented by the set $\mathcal{I}$ of all inequality constraints for the specific optimization problem. Points in the parameter space $\boldsymbol{x} \in \mathbb{R}^n$ fulfilling this constraints are called feasible points. Note that maximization problems in equation (2.1) can be formulated by multiplying the objective function with -1, hence (2.1) describes optimization in general. Since the parameter consists of real numbers, the problem formulated in equation (2.1) is called a continuous nonlinear optimization problem. The roughness calibration in Section 2.4.3.2 belongs to this class of problems.

[9] Note that the mathematical definitions throughout this section follow the naming conventions from (Nocedal and Wright 2006) for optimization problems.

In some optimization problems, the elements of vector $\boldsymbol{x}$ are not real numbers, but rather have integrality constraints and thus consist of integer numbers $\boldsymbol{x} \in \mathbb{Z}^n$. These problems are called integer programming problems and the field of solving this problems is called discrete optimization. Combinatorial problems like the optimal sensor placement problem in Chapter 4, for example, are often formulated as discrete optimization problems. Continuous optimization problems are easier to solve since algorithms can retrieve information of the functions behavior close to a point through derivation, e.g., slope and curvature (Nocedal and Wright 2006). For discrete problems, in contrast, the objective function can vary significantly between one feasible point and a point close or even adjacent to it. If some parameters in the trial variable vector $\boldsymbol{x}$ are chosen from real numbers $x_i \in \mathbb{R}^n$ and some parameters have integrality constraints $x_j \in \mathbb{Z}^n$, this problem is named mixed-integer programming problem. The problem of model-based leakage localization in Chapter 5 is a mixed integer programming problem, because integrality constraints for the location of the leak exist. However, this problem is reformulated to a continuous problem by temporarily ignoring the integrality constraints to make the problem solvable for DE algorithm which only works in continuous parameter spaces.

2.1.1.1. Local and global optima and convex functions

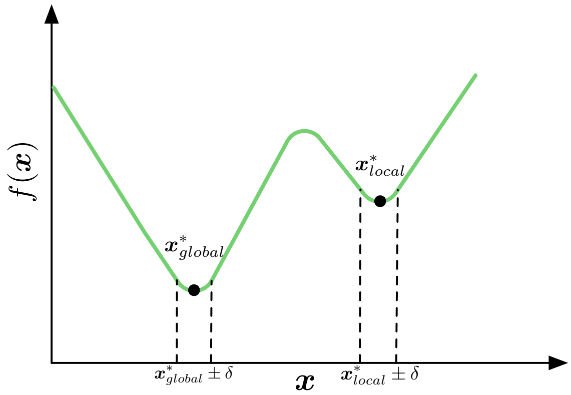


Figure 2.1.: Local and global optima of a function

$\boldsymbol{x}^*$ is called the solution of the problem defined in equation (2.1). The solution of the optimization problem stated in equation (2.1) is a global minimizer $\boldsymbol{x}^*$ of f or a **global optimal** solution if it is the point where the objective function has its minimal value over the whole parameter space. Mathematically (Nocedal and Wright 2006, see for example), a point $\boldsymbol{x}^*$ is a global minimizer if

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \mathbb{R}^n \quad . \tag{2.2}$$

In contrast, a point is a local minimizer or a **local optimum** if there exists only a neighborhood $\mathcal{N}$ of $\boldsymbol{x}^*$ such that

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \mathcal{N} \quad . \tag{2.3}$$

Usually, one lacks the global perspective of a function since—especially if a simulator is involved—one knows only the values of a function at certain points. Although, an interesting type of functions exist where global assertions on global optima can be made. This functions are called convex functions

($f : X \to \mathbb{R}$) on arbitrary vector spaces (X). f is a convex function if

$$\forall x_1, x_2 \in X,\ \forall t \in [0,1] : \qquad f(tx_1 + (1-t)x_2) \leq t f(x_1) + (1-t) f(x_2) \tag{2.4}$$

and is strictly convex if

$$\forall x_1 \neq x_2 \in X,\ \forall t \in (0,1) : \qquad f(tx_1 + (1-t)x_2) < t f(x_1) + (1-t) f(x_2)\ . \tag{2.5}$$

Every local minimum of a convex function is also a global minimum and a strictly convex function will have at most one global minimum. The property of a function being convex is strictly related to the properties of its second derivative, since a continuous, twice differentiable function (on a convex set) is convex if and only if its Hessian matrix $H(\boldsymbol{x})$ (see equation (2.7)) is positive (semi-)definite ($\boldsymbol{x}^{\mathsf{T}} H(\boldsymbol{x}) \boldsymbol{x} \geq 0$) (see for example Strang 1986). For optimization, convex functions have a lot of convenient properties which are efficiently used by diverse algorithms. The hydraulic solver in EPANET (Rossman 2000) minimizes such a function, for example.

2.1.2. Optimization methods

All manner of different optimization methods exist—some make use of algebra, others mimic nature. Each of them have in common that they are iterative methods, beginning with an initial guess of $\boldsymbol{x}$ and iteratively improving it until a global optimum $\boldsymbol{x}^*$ is reached or a termination criterion is met (Nocedal and Wright 2006). Yet, the methods differ in the strategy that is used to improve the trial vector by moving through the parameter space from one iteration step to the next. While some optimization methods make use of objective function $f(\boldsymbol{x})$ values, the constraints c_i of the optimization problem and perhaps its first and second derivative at current iteration—other methods use information of the optimization's history itself by incorporating knowledge from previous iteration steps in the present step. Others again, simultaneously solve the optimization problem not just with one trial vector $\boldsymbol{x}$, but with many possible solutions which exchange information of their current objective function values in the parameters space.

The No-Free-Lunch (NFL) theorem states that there cannot be an ideal algorithm solving all kinds of optimization problems (Wolpert and Macready 1997). Some algorithms perform better for some type of optimization problems, while other algorithms work better for other problems. Over the whole range of all mathematically possible optimization problems, all algorithms perform equally good. For example, classical methods (e.g. Newton's method) are efficient for linear, quadratic, strongly convex, unimodal objective functions, whereas evolutionary algorithms are better suited for discontinuous, non-differentiable, multi-modal and noisy objective functions (Bäck et al. 2000).

Nevertheless, one can define desired properties of good algorithms. According to Nocedal and Wright 2006, good optimization algorithms should be (i) robust, (ii) efficient and (iii) accurate. (i) Robust means that algorithms should perform well on a wide range of problems and also at the same problem with different starting points. (ii) Efficient means that the algorithms should not require exorbitant computation time, power or storage. (iii) Accurate means that good algorithms should be able to find a solution with any desired precision and not being sensitive to numerical or rounding errors. These goals may conflict since, for example, a more robust algorithm might be less efficient.

Additionally, algorithms can be classified according to which order of derivative they are using. Zero order methods use only the function $f(\boldsymbol{x})$ itself for finding the new optimization step. First order methods make use of the function and additionally the gradient of the function $\nabla f(\boldsymbol{x})$

$$\nabla f(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{pmatrix} \cdot f(x_1, \cdots, x_n) = \begin{pmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\boldsymbol{x})}{\partial x_n} \end{pmatrix} \quad . \tag{2.6}$$

Second order methods use the function, its gradient plus the second derivative, the Hessian matrix $H(\mathbf{x})$

$$H(\mathbf{x}) = \nabla \left(\nabla f(\boldsymbol{x})\right)^{\mathsf{T}} \overset{(2.6)}{=} \nabla \left(\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \cdots, \frac{\partial f(\boldsymbol{x})}{\partial x_n}\right) = \begin{pmatrix} \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1^2} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_1} \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_2} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_n} & \cdots & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n^2} \end{pmatrix} \tag{2.7}$$

Numerical calculations of the gradient in equation (2.6) are of order $\mathcal{O}(n) \cdot \mathcal{C}(f)$ (Naumann 2012), calculations of the Hessian in equation (2.7) are of order $\mathcal{O}(n^2) \cdot \mathcal{C}(f)$ (Naumann 2012). $\mathcal{O}(\cdot)$ is the Bachmann-Landau notation for classifying a computer algorithm's run time on the input size n and $\mathcal{C}(f)$ is the computational cost of an objective function evaluation. This corresponds to an EPANET simulation in most of the problems throughout this thesis.

Furthermore, optimization algorithms can be distinguished between deterministic (see Section 2.1.2.1) and stochastic (see Section 2.1.2.2) methods.

2.1.2.1. Deterministic algorithms

Deterministic methods are greedy methods for which $f(\boldsymbol{x})$ has to improve at each iteration step. Hence, deterministic methods end up in the same solution if they start from the same point x_0 in the parameter space. For that reason, they can get stuck in local optima.

There exist deterministic methods of different order. Zero order methods are, for example, the Simplex method or the Pattern search method; Gradient methods and Quasi Newton methods are of first order. Second order methods is the Nonlinear Conjugate Gradient (NCG) algorithm , for example.

All these methods have in common that they start from a vector $\boldsymbol{x}_k$ at iteration step k and move in the search direction $\boldsymbol{p}_k$ to retrieve a better point $\boldsymbol{x}_{k+1}$ with a better objective function value $f(\boldsymbol{x}_{k+1}) < f(\boldsymbol{x}_k)$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{p}_k \qquad \Rightarrow \qquad f(\boldsymbol{x}_{k+1}) < f(\boldsymbol{x}_k) \quad . \tag{2.8}$$

Yet, all these methods differ in how $\boldsymbol{p}_k$ is found.

2.1.2.1.1. Newton-Raphson Algorithm EPANET uses the Newton-Raphson Algorithm (NRA) for solving the hydraulic network equation problem in Section 2.2.2.1 equation (2.65). Originally, this method finds roots in functions. Therefore, if the global optimum of the function is zero and the function is convex, this method can find the global optimum. The idea behind this method is to approximate the function by its tangent and intercepting this tangent with $x = 0$. Subsequently, this gives a better estimate for the root and the iteration is started again until this root is found. Mathematically, this can be formulated in following way

$$y = \nabla f(\boldsymbol{x}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + f(\boldsymbol{x}_k) = 0 \qquad . \tag{2.9}$$

Rearranging leads to following iteration formula

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \nabla f(\boldsymbol{x}_k)^{-1} f(\boldsymbol{x}_k) \qquad . \tag{2.10}$$

The search direction is hence $\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k)^{-1} f(\boldsymbol{x}_k)$. Furthermore, this method is also the basis for the algorithm in Section 2.1.2.1.3.

2.1.2.1.2. Gradient Descent Algorithm The Gradient Descent Algorithm (GDA) is also called steepest descent. GDA updates the iteration step by going in the opposite direction of the gradient, since in this direction the function decreases the fastest

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k) \qquad . \tag{2.11}$$

Thus, the search direction is $\boldsymbol{p}_k = -\alpha \nabla f(\boldsymbol{x}_k)$. The α parameter can be updated in each iteration by line search methods in the current search direction or by using the inverse of the Hessian matrix $H(\boldsymbol{x})$ defined in equation (2.7) (see Section 2.1.2.1.3) leading to second order methods. Furthermore, there exist first order methods approximating $H(\boldsymbol{x})$ only by the gradient (Nocedal and Wright 2006).

2.1.2.1.3. Nonlinear Conjugate Gradient Algorithm The NCG algorithm (also known as nonlinear newton method) finds the ideal search vector $\boldsymbol{p}_k$ through a second order Taylor series approximation

$$f(\boldsymbol{x}_k + \boldsymbol{p}_k) \approx \underbrace{f(\boldsymbol{x_k}) + \nabla_x f(\boldsymbol{x_k})^\mathsf{T} \cdot \boldsymbol{p}_k + \frac{1}{2} \boldsymbol{p}_k^\mathsf{T} H(\boldsymbol{x_k}) \boldsymbol{p}_k}_{\Phi(\boldsymbol{p}_k)} + \cdots \qquad . \tag{2.12}$$

Higher order terms are neglected in the expansion. Minimizing the right hand side of the equation with respect to $\boldsymbol{p}_k$ results in an ideal search vector

$$\min_{\boldsymbol{p}_k \in \mathbb{R}^n} \Phi(\boldsymbol{p}_k) \quad \rightarrow \quad \nabla_p \Phi(\boldsymbol{p}_k) \overset{!}{=} \boldsymbol{0} \tag{2.13}$$

with the ∇_p operator defined in following way

$$\nabla_p = \begin{pmatrix} \frac{\partial}{\partial p_1} \\ \vdots \\ \frac{\partial}{\partial p_n} \end{pmatrix} \qquad . \tag{2.14}$$

Subsequently, throughout this derivation the indices p and x highlight the difference between the gradient in the parameter space defined in equation (2.6) and the operator in equation (2.14). Furthermore, following three mathematical identities—equation (2.15) to (2.17)—are needed

$$\nabla_p \left(\boldsymbol{u}^\mathsf{T} \boldsymbol{v}\right) = \left(\nabla_p \boldsymbol{u}^\mathsf{T}\right) \boldsymbol{v} + \left(\nabla_p \boldsymbol{v}^\mathsf{T}\right) \boldsymbol{u} \tag{2.15}$$

$$\nabla_p \boldsymbol{p}_k^\mathsf{T} = \begin{pmatrix} \frac{\partial}{\partial p_1} \\ \vdots \\ \frac{\partial}{\partial p_n} \end{pmatrix} \cdot (p_1, \cdots, p_n) = \begin{pmatrix} \frac{\partial p_1}{\partial p_1} & \cdots & \frac{\partial p_n}{\partial p_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial p_1}{\partial p_n} & \cdots & \frac{\partial p_n}{\partial p_n} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} = \mathbb{1} \tag{2.16}$$

$$\nabla_p \left(\boldsymbol{c}^\mathsf{T} \boldsymbol{p}_k\right) \overset{(2.15)}{=} \underbrace{(\nabla_p \boldsymbol{c}^\mathsf{T})}_{\mathbf{0}} \boldsymbol{p}_k + \underbrace{(\nabla_p \boldsymbol{p}_k^\mathsf{T})}_{(2.16)} \boldsymbol{c} = \mathbb{1} \cdot \boldsymbol{c} = \boldsymbol{c} \quad . \tag{2.17}$$

The first identity (equation (2.15)) is the derivative of the dot product of the vector values function $\boldsymbol{u}$ and $\boldsymbol{v}$, where the superscript $^\mathsf{T}$ denotes the transpose of the vector. The second identity (equation (2.16)) denotes the derivative of the transpose of the search vector $\boldsymbol{p}_k^\mathsf{T}$ resulting in an $(n \times n)$ identity matrix $\mathbb{1}$. Finally, the third equation (2.17) shows that the derivative of the transpose of constant vector $\boldsymbol{c}$ with respect to p (note that this vector can still be dependent on, e.g., $\boldsymbol{x}$) times the search vector results in this constant vector $\boldsymbol{c}$.

For equation (2.13), this leads to

$$\nabla_p \Phi(\boldsymbol{p}_k) = \underbrace{\nabla_p \left(\nabla_x f(\boldsymbol{x_k})^\mathsf{T} \cdot \boldsymbol{p}_k\right)}_{(A)} + \frac{1}{2} \underbrace{\nabla_p \left(\boldsymbol{p}_k^\mathsf{T} H(\boldsymbol{x_k}) \boldsymbol{p}_k\right)}_{(B)} \quad . \tag{2.18}$$

For clarity, the equation is divided into two parts (A) and (B). Starting with (A) and making use of identity (2.17) leads to

$$(A): \quad \nabla_p \left(\nabla_x f(\boldsymbol{x_k})^\mathsf{T} \cdot \boldsymbol{p}_k\right) \overset{(2.17)}{=} \nabla_x f(\boldsymbol{x_k}) \tag{2.19}$$

since the gradient $f(\boldsymbol{x}_k)$ at point $\boldsymbol{x}_k$ is not dependent on the search vector $\boldsymbol{p}_k$.

For part (B) starting with identity (2.15) and subsequently solving one resulting term after another leads to

$$\begin{aligned} (B): \quad \nabla_p \left(\boldsymbol{p}_k^\mathsf{T} H(\boldsymbol{x_k}) \boldsymbol{p}_k\right) &\overset{(2.15)}{=} \underbrace{(\nabla_p \boldsymbol{p}_k^\mathsf{T})}_{(2.16)} H(\boldsymbol{x_k}) \boldsymbol{p}_k + \underbrace{\left(\nabla_p \boldsymbol{p}_k^\mathsf{T} H(\boldsymbol{x_k})^\mathsf{T}\right)}_{(2.15)} \boldsymbol{p}_k = \\ &= \mathbb{1} \cdot H(\boldsymbol{x_k}) \boldsymbol{p}_k + \underbrace{(\nabla_p \boldsymbol{p}_k^\mathsf{T})}_{(2.16)} H(\boldsymbol{x_k})^\mathsf{T} \boldsymbol{p}_k + \underbrace{\nabla_p H(\boldsymbol{x_k})}_{\mathbf{0}} \boldsymbol{p}_k \boldsymbol{p}_k = \\ &= \left(H(\boldsymbol{x_k}) + H(\boldsymbol{x_k})^\mathsf{T}\right) \boldsymbol{p}_k = 2 \cdot H(\boldsymbol{x_k}) \boldsymbol{p_k} \quad . \end{aligned} \tag{2.20}$$

In the second row the Hessian matrix $H(\boldsymbol{x}_k)$ is not dependent on the search vector $\boldsymbol{p}_k$ and hence the derivative becomes zero. In the last row the symmetry of the Hessian matrix in equation (2.7) was used since

$$\frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j} = \frac{\partial^2 f(\boldsymbol{x})}{\partial x_j \partial x_i} \quad \rightarrow \quad H(\boldsymbol{x}_k) = H(\boldsymbol{x}_k)^\mathsf{T} \quad . \tag{2.21}$$

Thus, the result of equation (2.13) is

$$\nabla_p \Phi(\boldsymbol{p}_k) = \nabla_x f(\boldsymbol{x_k}) + H(\boldsymbol{x_k}) \boldsymbol{p}_k \overset{!}{=} 0 \quad . \tag{2.22}$$

Rearranging this equation

$$H(\boldsymbol{x_k}) \boldsymbol{p}_k = -\nabla_x f(\boldsymbol{x_k}) \tag{2.23}$$

leads to a formula for the ideal step direction

$$\boldsymbol{p}_k = -H^{-1}(\boldsymbol{x_k})\nabla_x f(\boldsymbol{x_k}) \quad . \tag{2.24}$$

Consequently, the update formula for a single Newton iteration step is equal to

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + H^{-1}(\boldsymbol{x_k})\nabla_x f(\boldsymbol{x_k}) \quad . \tag{2.25}$$

It has to be noted that the matrix inversion of the Hessian matrix in equation (2.25) comes at computational cost of $O(n^3)$. For that reason, NCG is computationally expensive for optimization problems in high dimensional parameter spaces and thus can be surpassed by meta-heuristic methods.

2.1.2.1.4. Levenberg-Marquardt The Levenberg-Marquardt method acts more like a gradient-descent method when the parameters are far from their optimal value, and acts more like the Gauss-Newton method when the parameters are close to their optimal value.

2.1.2.2. Stochastic algorithms

Stochastic algorithms, also called meta-heuristic algorithms, include a random component which allows escaping local optima. Reasonably, their convergence is due to their randomness usually slower, hence the big drawback is their higher computational burden compared to deterministic methods. Stochastic optimization algorithms are usually first order methods. Examples are Evolution Strategies (Rechenberg 1973), Genetic Algorithms (Goldberg 1989), Particle Swarm Optimization (Poli et al. 2007) or Simulated Annealing (Kirkpatrick et al. 1983).

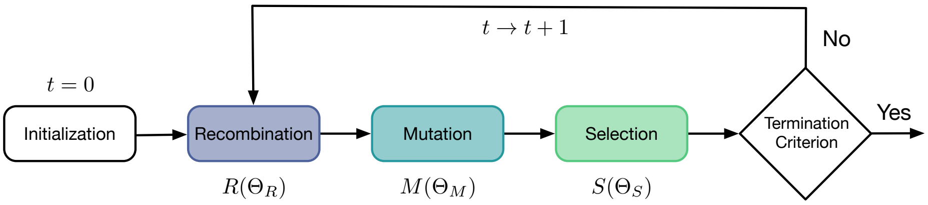


Figure 2.2.: Scheme of GA

2.1.2.2.1. Genetic Algorithms GAs are widely used to obtain optimal solutions to countless problems in the water related fields (Nicklow et al. 2009). GAs—first proposed by (Holland 1975)—mimic the principles of evolution to solve optimization problems. In contrary to the former mentioned algorithms, GAs are population based. Hence, not just a single solution evolves over the iterations, but rather they utilize the collective learning process of a population consisting of many single solutions (Bäck et al. 2000). Each solution—called individual—consists of parameters—called genes—which represents a single search point in the parameter space. Descendants of individuals are produced by random either (i) through reproduction by exchanging genes with other individuals or (ii) by mutation, introducing randomly small changes in genes mimicking germ line mutation effects. Subsequently, the fitness of each individual is determined and fitter individuals are more likely to survive and reproduce, thus, giving their good genes to potentially more descendants and hence increasing the fitness of the whole population over the iterations.

Mathematically, the population is defined as a time dependent $(n \times \mu)$-matrix $P(t)$

$$P(t) = (\boldsymbol{x}_1(t), \ldots, \boldsymbol{x}_\mu(t)) \in \mathbb{R}^{n \times \mu} \tag{2.26}$$

where a single column is a n-dimensional parameter vector $\boldsymbol{x_i} \in \mathbb{R}^n$, a possible solution of the optimization problem as described in Section 2.1.1. μ is the size of the population respectively the number of individuals in it and t denotes the current time or iteration step since the population and its individuals change over time. Recombination, mutation and selection can be described as operators (R, M, S) effecting the population and are applied consecutively on the population (see Figure 2.2). Additionally, an **initialization** procedure generates a population of individuals (Bäck et al. 2000). Usually, the single individuals are produced randomly over the whole parameter space. After the initialization, the fitness of the individuals can be determined and the fittest individuals can be selected before the evolution starts.

Recombination The first operator in the evolution is the recombination operator, also called crossover operator. Recombination is responsible for large changes in the solution vectors. The operator produces κ new solutions by combining genes of different individuals of the population of size μ

$$R(\Theta_R): \quad \mathbb{R}^{n \times \mu} \quad \rightarrow \quad \mathbb{R}^{n \times \kappa} \qquad , \tag{2.27}$$

In addition to the population, the operator also depends on an additional set of parameters Θ_R, controlling the reproduction of the individuals such as the probability p_r that the genes of two individuals are recombined. GAs have in common that they favor recombination over mutation (Bäck et al. 2000), hence, p_r is chosen to be high (e.g. $p_r = 0.8$).

Single-Point-Crossover (SPC) is a specific example of a recombination operator (see Figure 2.3). Two individuals are taken and a crossover point is selected by random between $c \in \mathbb{N} \cap [1, \ldots, n]$. All genes beyond that crossover point are swapped between the two individuals to produce two descendants. Let i, j be random integer numbers drawn from $i, j \in \mathbb{N} \cap [1, \ldots, \mu]$ choosing two individuals $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ from the population $P(t)$, then the SPC operator works on the population $P(t)$ as follows

$$R(P(t)|\Theta_R) = R \begin{pmatrix} \cdots & x_{1,i} & \cdots & x_{1,j} & \cdots \\ & \vdots & & \vdots & \\ & x_{c,i} & & x_{c,j} & \\ & x_{c+1,i} & & x_{c+1,j} & \\ & \vdots & & \vdots & \\ \cdots & x_{n,i} & \cdots & x_{n,j} & \cdots \end{pmatrix} = \begin{pmatrix} \cdots & x_{1,i} & \cdots & x_{1,j} & \cdots \\ & \vdots & & \vdots & \\ & x_{c,i} & & x_{c,j} & \\ & x_{c+1,j} & & x_{c+1,i} & \\ & \vdots & & \vdots & \\ \cdots & x_{n,j} & \cdots & x_{n,i} & \cdots \end{pmatrix} = P'(t) \tag{2.28}$$

to produce and altered population $P'(t)$ containing two new individuals $\boldsymbol{x}'_i$ and $\boldsymbol{x}'_j$.

Mutation The second operator in the GA is the mutation operator. This operator makes small changes to the genes of an individual, hence, broadening the genetic variability of a population

$$M(\Theta_M): \quad \mathbb{R}^{n \times \kappa} \quad \rightarrow \quad \mathbb{R}^{n \times \lambda} \quad . \tag{2.29}$$

The operator also depends on an additional set of parameters Θ_M, controlling the mutation of individuals. GAs have in common that the mutation operator is applied with a low probability p_m so that

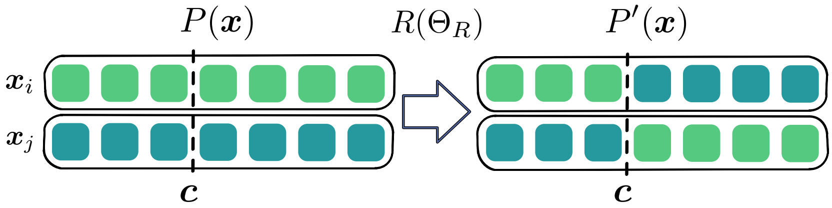


Figure 2.3.: Scheme of SPC recombination operator

mutation works more as a "background operator" (Bäck et al. 2000) (usually $p_m < 0.2$).

Uniform-Integer Mutation (UIM) is an example for a specific mutation operator, where a gene x_{ij} with $j \in \mathbb{N} \cap [1, \ldots, n]$ of an individual is chosen at random and, subsequently, this gene is replaced by drawing randomly an integer $x'_{ij} \in \mathbb{N} \in [a_j, \ldots, b_j]$ within the parameter space boundaries of parameter j (see Figure 2.4). The effect of UIM on a population $P'(t)$ can mathematically be described as

$$M(P'_i(t)|\Theta_M) = M\left((x_{i1} \ldots x_{ij} \ldots x_{in})^\intercal\right) = (x_{i1} \ldots x'_{ij} \ldots x_{in})^\intercal = P''_i(t) \qquad . \tag{2.30}$$

The mutation operator is applied to each individual x_i of the population with probability p_m.

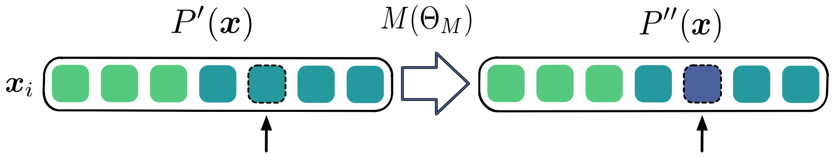


Figure 2.4.: Scheme of UIM operator

Selection The last operator in the GA is the selection operator. This operator chooses μ individuals from the through mutation and recombination altered population of λ individuals based on their fitness

$$S(f, \Theta_S): \quad \mathbb{R}^{n\times\lambda} \quad \to \quad \mathbb{R}^{n\times\mu} \qquad . \tag{2.31}$$

Once again, this operator can depend on additional parameters Θ_S, controlling the selection process. Usually, for GAs the selection is probabilistic so that unfit individuals have a chance to further participate in the evolution by giving their genes to descendants (Bäck et al. 2000). Before the selection operator is applied on the population, the fitness of each individual has to be evaluated through the objective function $f(\mathbf{x})$

$$f(\boldsymbol{x}): \quad \mathbb{R}^n \quad \to \quad \mathbb{R} \qquad , \tag{2.32}$$

leading to λ fitness values. The evaluation of the fitness corresponds normally to an EPANET simulation throughout this thesis.

For example, tournament selection serves as a specific selection operator. This operator applies for the survival of the fittest μ tournaments between k at random chosen individuals, where the winner of each tournament will survive. The probability of an individual to win depends on its fitness as well as on the fitness of his opponents in the tournament. Mathematically, the fitness of the k individuals is sorted in ascending order $f(x_1) \leq f(x_2) \leq \ldots \leq f(x_k)$ and the probability $\mathcal{P}(i)$ of the i-th fittest individual to win is

$$\mathcal{P}(i) = p_T\,(1 - p_T)^{i-1} \qquad . \tag{2.33}$$

$p_T \in \mathbb{R} \cap (0,1]$ represents the probability of the fittest individual to win the tournament. For $p_T = 1$ the best individual wins any tournament.

After selection, the convergence of the algorithm is examined by checking one or more **termination criteria**. This criteria can be fulfilled, for example, (i) if a certain fitness value is reached, (ii) if the fitness of the population did not decrease over a certain number of iterations or (iii) if a maximum number of iterations $N_{\max(I)}$ is performed.

If the termination criterion is not fulfilled, the circle of recombination, mutation and selection is again repeated with the current population. This circle is named a generation in GA terminology. In contrary, if the termination criterion is fulfilled after I iterations, the individual of the current population with the smallest fitness value is chosen as the best solution of the problem

$$\boldsymbol{x}^* \quad \ni \quad \min_{f(\boldsymbol{x})} P(t+I) \quad . \tag{2.34}$$

Due to the stochastic components of GAs, it can not be guaranteed that $\boldsymbol{x}^*$ is a global optimal nor a local optimum solution. Coupling the GA with a subsequent deterministic method by using the GA's result as the starting point of the deterministic algorithm, can at least guarantee local optimality. However, just meta-heuristic algorithms have the chance to find global optima for multi-modal, non-convex and noisy fitness functions, where deterministic algorithms alone fail.

It has to be noted that the default case for population sizes throughout one iteration for GAs is $\kappa = \lambda = \mu$. Furthermore, the computational cost of the GA described in this section is approximately

$$\mu(1+(p_r+p_m)N_I)\cdot \mathcal{C}(f) \qquad , \tag{2.35}$$

where μ is the population size, p_r and p_m are the recombination (crossover) and mutation probability, N_I is the number of iterations that are performed and $\mathcal{C}(f)$ is the computational cost of a fitness function evaluation. Note that the dimensionality n of the problem does not occur in the computational costs in contrary to the NCG algorithm in Section 2.1.2.1.3.

2.1.2.2.2. Differential Evolution The DE algorithm was introduced by Ken Price and Rainer Storn two decades ago (Storn and Price 1995, 1997). Basically, DE works the same way as GAs. The algorithm is again population based. Individuals—who are now called candidate solutions or agents—build a population of solutions. The candidates move around the search space and are altered and combined by simple algebraic formulas. The crucial idea behind DE is that this algorithm generates new candidate solutions by adding a weighted difference vector between two population agents to a third agent (Storn and Price 1995). This process can again be seen as a sequence of mutation, recombination and selection operators. Mutation leads again to new solutions in the parameter space while

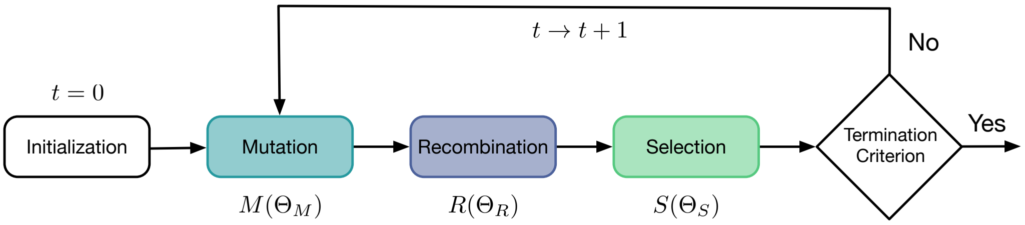


Figure 2.5.: Scheme of DE algorithm

recombination combines information contained in two agents, one from the original population and one as the result from mutation, to obtain new candidate solutions. Confusingly, mutation in the DE algorithm is also similar to recombination in GAs (according to Storn and Price 1995). Furthermore, this operators are used in different order compared to GAs since mutation and recombination are switched (see Figure 2.5). If a candidate solution leads to better fitness values, it is selected to replace the agent with whom its fitness was compared and becomes part of the population in the following iteration. Thus, this selection procedure is in general greedier than in GAs.

Before the algorithm starts, a first population $P(t=0)$ of μ solutions has to be generated randomly within the boundaries of the parameter space. This is called the **initialization** phase.

Mutation The mutation operator generates new parameter vectors—called donors—by adding a weighted difference vector with weights F_i between (at least) two population members $\boldsymbol{x}_{r1}$ and $\boldsymbol{x}_{r2}$ randomly drawn from the population to a third member, which (i) can also be randomly drawn $\boldsymbol{x}_{r3}$ from the population, or (ii) is the same vector as used for recombination $\boldsymbol{x}_j$ or (iii) is the best solution $\boldsymbol{x}_{best}$ of the current population. Which of these three cases is applied depends on the mutation strategy that is used. Table 2.1 shows different mutation strategies described in Storn 1996. Figure 2.6 shows the mutation on the example of the DE/best/1 strategy. Two random vectors are taken from the population, their difference vector is build and this vector is subsequently multiplied by the mutation factor F_1, which controls the amplification of the differential variation (Storn and Price 1995), and then added to the best solution $\boldsymbol{x}_{best}$ so far to generate the donor vector v. Once the donor vector v_j is produced, it is passed to the recombination operator.

Table 2.1.: Mutation strategies for DE algorithm (according to Storn 1996)

Strategy	**Donor Formula ($\boldsymbol{v}_j = \ldots$)**
DE/rand/1	$\boldsymbol{x}_{r1} + F_1 (\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3})$
DE/best/1	$\boldsymbol{x}_{best} + F_1 (\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3})$
DE/rand to best/1	$\boldsymbol{x}_{r1} + F_1 (\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}) + F_2 (\boldsymbol{x}_{best} - \boldsymbol{x}_{r1})$
DE/current to best/1	$\boldsymbol{x}_j + F_1 (\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}) + F_2 (\boldsymbol{x}_{best} - \boldsymbol{x}_j)$
DE/rand/2	$\boldsymbol{x}_{r1} + F_1 (\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3} + \boldsymbol{x}_{r4} - \boldsymbol{x}_{r5})$
DE/best/2	$\boldsymbol{x}_{best} + F_1 (\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3} + \boldsymbol{x}_{r4} - \boldsymbol{x}_{r5})$

Recombination The recombination operator increases the diversity of the parameter vectors by combining the donor vector $\boldsymbol{v}_j$ by forming a trial vector $\boldsymbol{u}_j$ through following procedure (Storn and Price 1997)

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } \mathrm{rand}_{i,j} \leq CR \quad \vee \quad i = j_{rand} \\ x_{i,j} & \text{else} \end{cases} , \tag{2.36}$$

where $\mathrm{rand}_{i,j}$ is a uniform random number drawn within the interval $\mathrm{rand}_{i,j} \in [0,1]$ and CR is the crossover probability parameter chosen from $CR \in [0,1]$ determined by the user. $\boldsymbol{x}_j$ is called the target vector and j_{rand} is a randomly drawn integer from $j_{rand} \in \mathbb{N} \cap [1,n]$, assuring that at least one parameter differs between the trial vector $\boldsymbol{u}_j$ and the target vector $\boldsymbol{x}_j$. Thus, if $\mathrm{rand}_{i,j}$ is smaller than CR for the i-th element of the vectors, the i-th element is transferred from the donor to the trial vector, otherwise the i-th element is taken from the target vector. Figure 2.7 shows a graphical representation of the recombination procedure (based on Storn and Price 1997, Figure 2).

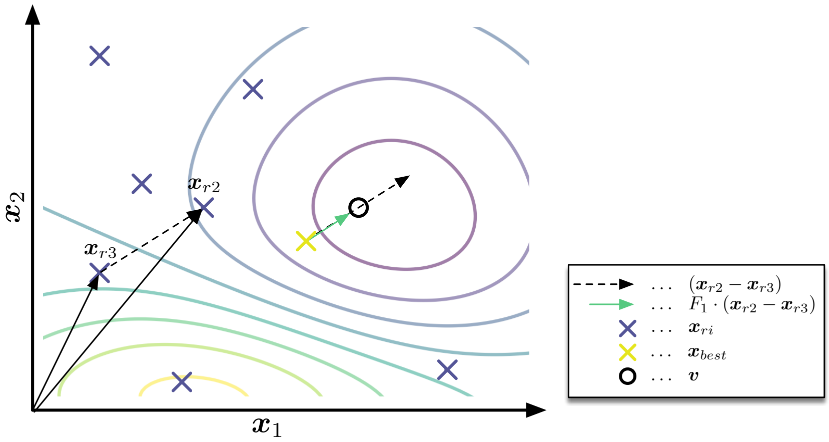

$\boldsymbol{x}_2$
$\boldsymbol{x}_1$
$\boldsymbol{x}_{r2}$
$\boldsymbol{x}_{r3}$
... $(\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3})$
... $F_1 \cdot (\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3})$
... $\boldsymbol{x}_{ri}$
... $\boldsymbol{x}_{best}$
... $\boldsymbol{v}$

Figure 2.6.: Graphical representation of DE/best/1 mutation operator in DE algorithm (based on Storn and Price 1995)

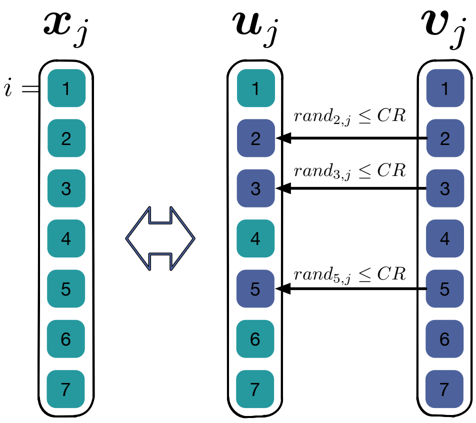


Figure 2.7.: Graphical representation of recombination operator in DE algorithm (based on Storn and Price 1997)

Selection Before selection, the fitness of the trial vectors has to be evaluated. Subsequently, the selection operator compares the fitnesses of $\boldsymbol{x}_j$ and $\boldsymbol{u}_j$ in iteration k and the fitter individuals will proceed to the next iteration. This can be mathematically written as

$$\boldsymbol{x}_j^{k+1} = \begin{cases} \boldsymbol{u}_j^k & \text{if} \quad f(\boldsymbol{u}_j^k) < f(\boldsymbol{x}_j^k) \\ \boldsymbol{x}_j^k & \text{else} \end{cases} \quad . \tag{2.37}$$

The iteration again stops, if a termination criterion is met. This criterion can be the same as for GAs. Furthermore, an additional criterion might be if the crowding radius c_r is below a certain value. The crowding radius of a population $P(t) \in \mathbb{R}^{n \times \mu}$ of μ candidate solutions with parameter dimension n is defined as follows

$$c_r = \max\left(\sum_{i=1}^{n} (x_{ij} - \bar{x}_i)^2\right) \quad \text{with} \quad \bar{x}_i = \frac{1}{\mu}\sum_{j=1}^{\mu} x_{ij} \quad . \tag{2.38}$$

Other possibilities are reseeding (initializing) the entire population by only keeping the best solution so far $\boldsymbol{x}^*$. This can be done if the variability in the population described by c_r is low to increase once again the diversity of the population.

Finally, the computational cost of the algorithm is

$$\mu(N_I + 1) \cdot \mathcal{C}(f) \quad , \tag{2.39}$$

where μ is the population size, N_I is the number of iterations and $\mathcal{C}(f)$ is the computational cost of a fitness function evaluation. Again, the cost is not dependent on the parameter space dimension similar to GAs.

Moreover, it has to be mentioned that DE works only in continuous parameter spaces. For parameter spaces with integrality constraints this problem can be circumvented by mapping real values on integer values just for the fitness evaluation through, e.g., floor or ceiling functions

$$\text{floor}(x_i) = \lfloor x_i \rfloor \quad \text{or} \quad \text{ceiling}(x_i) = \lceil x_i \rceil \quad \text{with} \quad x_i \in \mathbb{R} \quad , \tag{2.40}$$

where $\lceil x_i \rceil$ ($\lfloor x_i \rfloor$) is the least (greatest) integer that is greater (less) than or equal to x_i. This workaround will reappear in the chapter on model-based leak localization (Chapter 5).

2.1.3. Optimization examples

To test the former introduced mathematical optimization algorithms—deterministic as well as stochastic ones—two optimization test functions are taken from literature: (i) the Rosenbrock function in Section 2.1.3.1 and (ii) the Rastrigin function in Section 2.1.3.2. These functions have different properties (e.g. unimodal, multi-modal) and their fitness landscapes will serve as excellent examples for objective functions faced later on in this thesis. The objective functions will be from real-world optimization problems in WDSs (e.g. roughness calibration, model-based leakage localization, optimal sensor placement). Furthermore, the strengths and drawbacks of different optimization algorithms will be shown, providing the reader of this thesis with a deeper understanding which algorithm is useful for what problem class. Additionally, fitness landscapes as well as convergence considerations will be examined that will also reappear throughout this thesis. The examples serve also as test examples showing that the implemented optimization algorithms function properly.

2.1.3.1. Rosenbrock function

The Rosenbrock function is a non-convex function with one minimum (unimodal) and is widely used as a performance test problem for optimization algorithms (Rosenbrock 1960). Its general form for n dimensions is

$$f(\boldsymbol{x}) = \sum_{i=1}^{n-1} \left((a - x_i)^2 + b \left(x_{i+1} - x_i^2 \right)^2 \right) \qquad . \tag{2.41}$$

The two-dimensional form $n = 2$ results in

$$f(\boldsymbol{x}) = (a - x_1)^2 + b(x_2 - x_1^2)^2 \qquad . \tag{2.42}$$

The optimum $\boldsymbol{x}^*$ of the Rosenbrock function

$$\boldsymbol{x}^* = \begin{pmatrix} a \\ a^2 \end{pmatrix} \quad \text{with} \quad f(\boldsymbol{x}^*) = 0 \tag{2.43}$$

lies in a flat, narrow and curved channel, which gives the function its nickname Rosenbrock's banana. For applying GDA optimization algorithm, calculation of the gradient is needed. The gradient according to equation 2.6 for the two-dimensional Rosenbrock function results in

$$\nabla f(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \frac{\partial f(\boldsymbol{x})}{\partial x_2} \end{pmatrix} = \begin{pmatrix} -4b \left(x_2 - x_1^2 \right) x_1 - 2 \left(a - x_1 \right) \\ 2b \left(x_2 - x_1^2 \right) \end{pmatrix} \qquad . \tag{2.44}$$

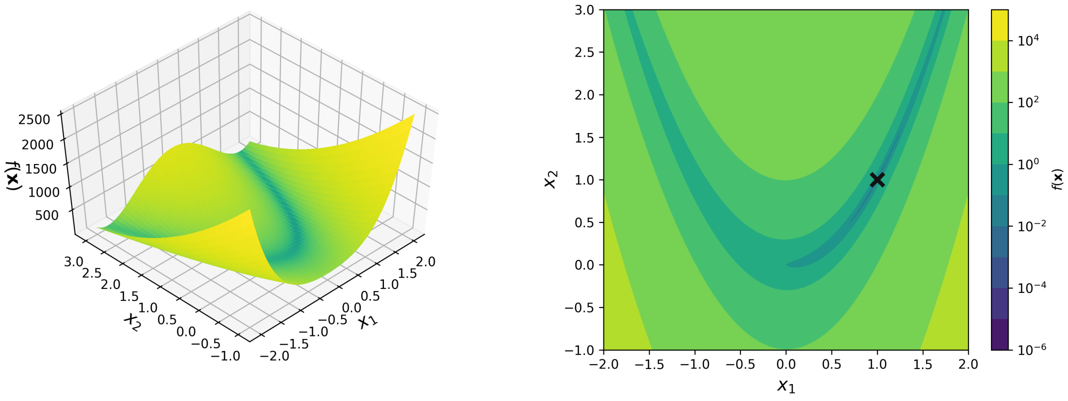
Additionally, for the NCG method, the Hessian matrix according to 2.7 has to be calculated. The Hessian $H(\boldsymbol{x})$ results in

$$\nabla \left(\nabla f(\boldsymbol{x}) \right)^{\mathsf{T}} = \begin{pmatrix} \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1^2} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_1} \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_2} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} 12bx_1^2 - 4bx_2 + 2 & -4bx_1 \\ -4bx_1 & 2b \end{pmatrix} \tag{2.45}$$

.

Setting $a = 1$ and $b = 100$ within the parameter space boundaries $x_1 \in [-2, 2]$ and $x_2 \in [-1, 3]$ results in a specific example of the Rosenbrock function that is examined in this thesis. This function has a global minimum at $\boldsymbol{x}^* = (1, 1)^{\mathsf{T}}$. The gradient at this point is $\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$ and the determinant of the Hessian is positive $|H(\boldsymbol{x}^*)| > 0$, proofing that this is in fact a minimum. The fitness landscape of the function over the parameter space is depicted in Figure 2.8—on the left hand side as a 3-d plot (see Figure 2.8a) and on the right hand side as a contour plot (see Figure 2.8b), in which the global minimum is depicted as a black cross. Note the $\log_{10}$ coloring scheme to enhance the visibility of the function's narrow banana-shaped valley.

2.1.3.1.1. GDA VS NCG The convergence properties of GDA (see Section 2.1.2.1.2) and NCG (see Section 2.1.2.1.3) are compared on the two-dimensional Rosenbrock function (see Figure 2.8) defined in equation (2.42). A special formulation of the GDA method is used—the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formulation (see for example Fletcher 1987; Nocedal and Wright 2006). In



(a) 3-d plot

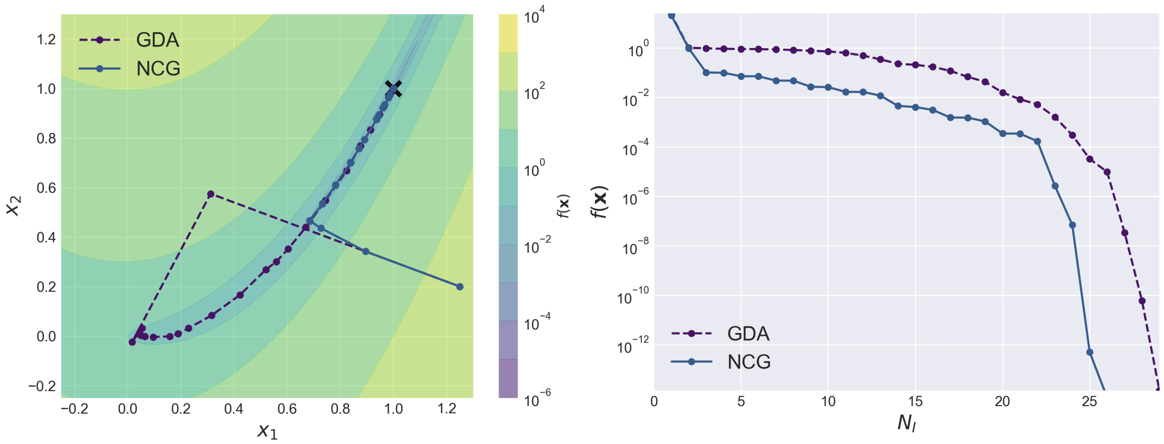
(b) contour plot in $\log_{10}$ scale

Figure 2.8.: Fitness landscape of Rosenbrock function

this algorithm, the Hessian matrix is approximated through evaluations of the gradient to find the α parameter needed for the optimal search vector in equation (2.11). NCG is used as defined in equation (2.25). The implementations of the GDA and NCG algorithm are taken from Python's SciPy package (Jones et al. 2001) throughout this thesis.

First, to compare the convergence of GDA and NCG, both algorithms are started from the same starting point $\boldsymbol{x}_0 = (1.25, 0.2)^\intercal$. Figure 2.9 shows the results of the optimization run. Figure 2.9a presents the trajectory of the solution over the iterations in the parameter space, while Figure 2.9b shows the fitness values of the algorithms as a function of the iterations N_I in $\log_{10}$ scale. Figure 2.9a clearly shows that, despite starting from the same point, the algorithms perform differently. GDA overshoots the valley already in the first iteration step and takes time to recover from that while NCG is more reluctant in entering the valley. This is a result of utilizing information of the real Hessian matrix in NCG compared to using approximations in the BFGS formulation of GDA. After overshooting, GDA starts in the wrong direction and is zig-zagging while entering the valley. Finally, once the trajectory has entered the canyon, GDA converges fast. NCG does not have this problems and converges straight to the optimal solution marked as black cross in Figure 2.9a. This can also be seen in a faster convergence of the fitness values of NCG compared to GDA in Figure 2.9b. Hence, if the Hessian of a problem is known, NCG is preferred over GDA for unimodal differentiable functions. Additionally, the greediness of the two methods can be seen in Figure 2.9b, since the fitness values monotonically decrease at each iteration step ($f(\boldsymbol{x}|N_{I+1}) < f(\boldsymbol{x}|N_I)$).

2.1.3.1.2. Different Starting Points Second, the effect of different starting points on the convergence behavior of NCG is examined again on the two-dimensional Rosenbrock function. The optimization algorithm starts at three different starting points distributed over the parameter space, namely $\boldsymbol{x}_{01} = (-1.8, 2.0)^\intercal$, $\boldsymbol{x}_{02} = (-0.5, 1.5)^\intercal$ and $\boldsymbol{x}_{03} = (1.0, 2.8)^\intercal$. Logically, this results in different trajectories in the parameter space (see Figure 2.10a). Furthermore, the starting points also result in different convergence speeds. Whereas the NCG algorithm takes 25 iterations to find the global optimum starting at $\boldsymbol{x}_{03}$, the algorithm needs approximately ten times more iterations starting from $\boldsymbol{x}_{01}$ (see Figure 2.10b). Hence, gradient-based algorithms are sensitive to their starting points. Nevertheless, since the Rosenbrock function is unimodal, all optimization runs at all starting points result finally in the global

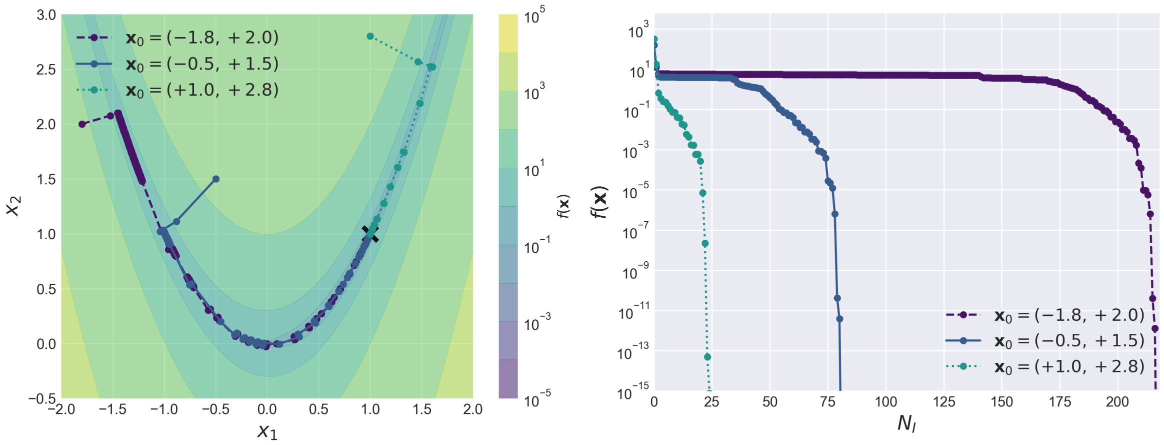


(a) trajectory

(b) convergence

Figure 2.9.: Comparison of convergence of GDA and NCG algorithm on the Rosenbrock function

optimum.

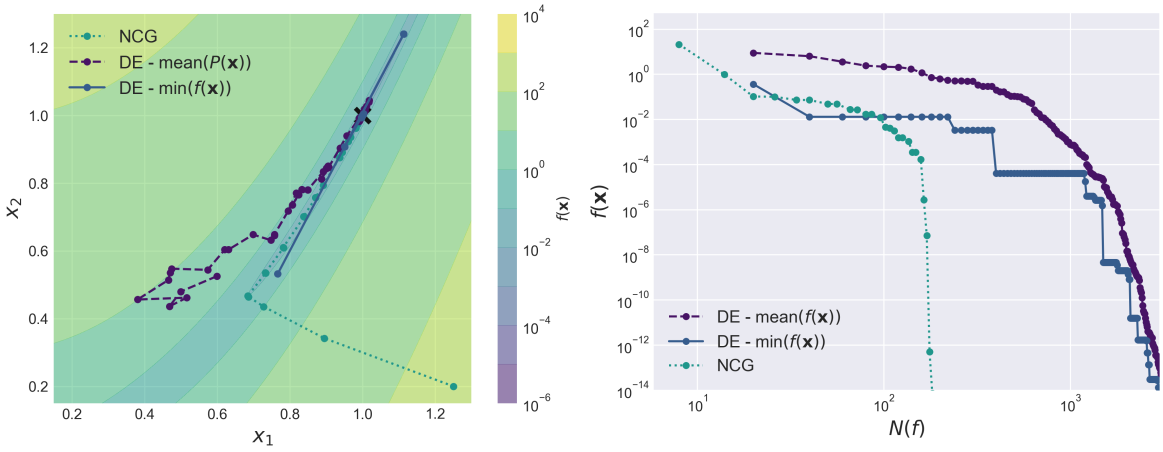


(a) trajectory

(b) convergence

Figure 2.10.: Comparison of convergence of NCG with different starting points on the Rosenbrock function

2.1.3.1.3. DE VS NCG Finally, the performance of the DE (Section 2.1.2.2.2) and the NCG algorithm are compared on the Rosenbrock function (see Figure 2.11). Since DE is a population based algorithm, the comparison differs compared to Section 2.1.3.1.1 and 2.1.3.1.2. The starting point for NCG is the same as in section 2.1.3.1.1. The DE has no single starting point since the population $P(\boldsymbol{x})$ is initialized randomly over the whole search space. The boundaries of the search space are limited to $x_i \in [0.15, 1.3]$ in this example fitting the segment of the Rosenbrock function depicted in Figure 2.11a. DE with DE/rand/1 mutation strategy is used with a population size of $\mu = 20$, a crossover probability of $c_r = 0.5$ and a mutation factor of $F = 0.8$. The parameters are chosen according to the rule of thumb described in (Storn 1996). The trajectory of the mean of all solutions (mean($P(\boldsymbol{x})$)) of the population $P(\boldsymbol{x})$ in one generation and the best solution $\boldsymbol{x}^*$ with the lowest fitness value $(\min(f(\boldsymbol{x})))$ of the DE algorithm is shown in Figure 2.11a. This is compared with the trajectory of the NCG algorithm. It can be seen that DE needs crucially longer to find the optimal solution. At first the variability in the solution is high shown in a high difference of the mean of the population and the best solution. Over the generations this discrepancy gets lower indicating lower variability as the fittest candidate solutions survive the selection. Since DE evaluates the fitness function μ times in each generation, the number of fitness function evaluations $N(f)$ is taken into account instead of the number of iterations N_I in the previous examples (see Figure 2.11b). The computation of the Hessian as well as the gradient is also taken into account in $N(f)$ for the NCG algorithm. The fitness value convergence is depicted in a $\log_{10} - \log_{10}$-scale in Figure 2.11b. The NCG algorithm converges nearly 100 times faster than DE. Consequently, deterministic algorithms like NCG should be favored to stochastic algorithms like DE for unimodal problems due to their higher convergence speed. Stochastic algorithms are less efficient than deterministic algorithms on unimodal problems.



(a) trajectory

(b) convergence

Figure 2.11.: Comparison of convergence of DE and NCG on the Rosenbrock function

2.1.3.2. Rastrigin function

The second test function is the famous Rastrigin function which was first formulated by Rastrigin 1974. It is a non-convex, multi-modal function with many local optima and is widespread used as a test problem for optimization algorithms. It is defined in n dimensions as follows

$$f(\boldsymbol{x}) = a \cdot n + \sum_{i=1}^{n} \left(x_i^2 - a \cdot \cos\left(2\pi\, x_i\right)\right) \tag{2.46}$$

and has besides its many local minima a global minimum $\boldsymbol{x}^*$ at

$$\boldsymbol{x}^* = \boldsymbol{0} \quad \text{with} \quad f(\boldsymbol{x}^*) = 0 \qquad . \tag{2.47}$$

For applying GDA and NCG, again, the calculation of the gradient and the Hessian is necessary. The gradient results in

$$\frac{\partial f(\boldsymbol{x})}{\partial x_i} = 2x_i + 2\pi a \sin(2\pi x_i) \qquad . \tag{2.48}$$

The function has an infinite number of local optima at

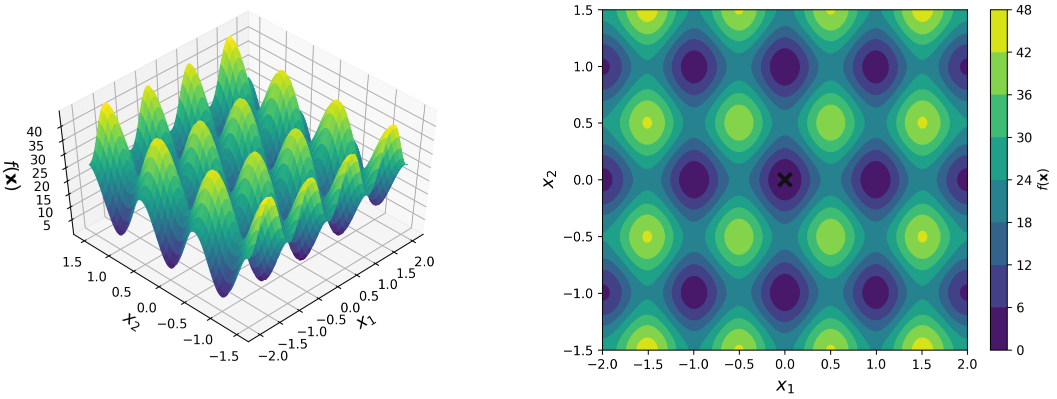
$$x_i = \frac{n_i}{2} \quad \text{with} \quad n_i \in \mathbb{Z} \quad \forall\, i \qquad . \tag{2.49}$$

The Hessian calculation leads to

$$\frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j} = \begin{cases} \text{if } i = j: & 2 + 4\pi^2 a \cos(2\pi x_i) \\ \text{if } i \neq j: & 0 \end{cases} \qquad . \tag{2.50}$$

The Hessian is a sparse, diagonal matrix. The determinant of the Hessian is positive if n_i is even leading to a local minimum, whereas if n_i is odd, a negative determinant and a local maximum results.

The two-dimensional Rastrigin function with $a = 10$ within the parameter space boundaries $x_i \in [-2, 2]$ is shown in Figure 2.12. Figure 2.12a shows the 3-d plot of the function's fitness landscape while Figure 2.12b shows its contour. The global minimum at $\boldsymbol{x}^* = (0,0)^\mathsf{T}$ is depicted as a black cross. The function's many local optima can be seen in Figure 2.12.

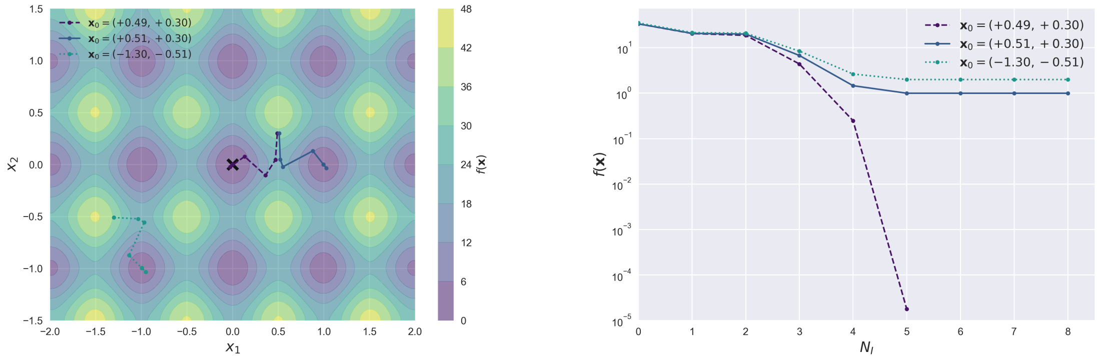


(a) 3-d plot

(b) contour plot

Figure 2.12.: Fitness landscape of Rastrigin function

2.1.3.2.1. Deterministic Algorithm with Different Starting Points First, the effect of different starting points is examined similar to Section 2.1.3.1.2. However, the NCG algorithm is now applied on a multi-modal function. Again, optimization is started at three different starting points ($\boldsymbol{x}_{01} = (0.49, 0.30)^\mathsf{T}$, $\boldsymbol{x}_{02} = (0.51, 0.30)^\mathsf{T}$ and $\boldsymbol{x}_{03} = (-1.30, -0.51)^\mathsf{T}$). The results can be seen in Figure 2.13. Figure 2.13a shows again the convergence in parameter space, while 2.13b shows the convergence of the fitness function $f(\boldsymbol{x})$ as a function of iterations. In contrast to Section 2.1.3.1, the algorithm does not find the global optimum for every start point. Only $\boldsymbol{x}_{01}$ converges to the global optimal solution, whereas $\boldsymbol{x}_{02}$ and $\boldsymbol{x}_{03}$ get stuck in the local optimal solution $\boldsymbol{x}^*_{local} = (1.0, 0.0)^\mathsf{T}$ with $f(\boldsymbol{x}^*_{local}) = 1.0$ respectively $\boldsymbol{x}^*_{local} = (-1.0, -1.0)^\mathsf{T}$ with $f(\boldsymbol{x}^*_{local}) = 2.0$—also visible in the convergence of the fitness function in Figure 2.13b. In fact, only starting points located in proximity of the global optimum in the subsection $x_i \in (-0.5, 0.5)$ of the parameter space converge to the global optimum. Other starting points lead to solutions attracted by local optima. This is a result of the greediness of deterministic search algorithms described in Section 2.1.2.1. Hence, deterministic algorithms are not robust on multi-modal problems.



(a) trajectory

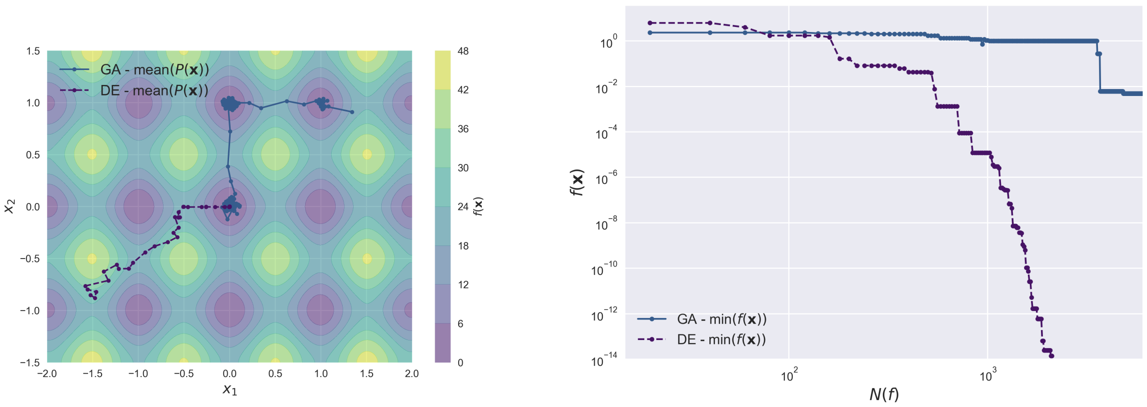
(b) convergence

Figure 2.13.: Comparison of convergence of NCG with different starting points on the Rastrigin function

2.1.3.2.2. GA vs DE The reason why multi-modal optimization problems have to be solved with stochastic algorithms is the lack of robustness of deterministic algorithms resulting from their greediness. This section compares the performance of two stochastic algorithms, e.g. GA and DE, on the Rastrigin function. GA and DE are implemented with the help of Python's DEAP toolbox (Fortin et al. 2012) in this thesis. The results can be found in Figure 2.14.

DE with DE/rand/1 mutation strategy is used with a population size of $\mu = 20$, a crossover probability of $c_r = 0.5$ and a mutation factor of $F = 0.8$. The parameters are chosen according to the rule of thumb described in (Storn 1996).

The settings for GA consist of a populations size of $\mu = 20$, a two-point crossover operator is applied with the crossover probability of $p_c = 0.8$ and a Gaussian mutation operator with $\sigma = 1.0$ is applied with a mutation probability of $p_m = 0.2$. DE is initialized in the lower left square of Figure 2.14a within the boundaries $x_1 = [-2.0, -1.0]$ and $x_2 = [-1.5, -0.5]$. GA is initialized in the upper right square of Figure 2.14a within the boundaries $x_1 = [1.0, 2.0]$ and $x_2 = [0.5, 1.5]$. Since the Rastrigin function is symmetric, this results in the same starting conditions for the two algorithms. The mean of the population $P(\boldsymbol{x})$ is depicted in the parameter space in Figure 2.14a while the convergence of the mean fitness function values $f(\boldsymbol{x})$ is shown in Figure 2.14b. DE is greedier than GA resulting in faster convergence. However, DE is not greedy enough to get stuck in a local optimum for this problem. Furthermore, GA gets stuck in local optima, but due to stochastic processes in the algorithm especially introduced through the mutation operator, pulling the population out of local optima. Finally, the algorithm finds the surroundings of the global optimum, but it takes a long time until the optimum is found with high precision. DE is more efficient than GA in converging to the global optimum. Note, that both algorithms are initialized in an area far from the global optimum, which is surrounded by maxima. Nevertheless, both algorithms managed to overcome this maxima to find the global optimal solution.



(a) trajectory

(b) convergence

Figure 2.14.: Comparison of convergence of GA and DE on the Rastrigin function

2.2. Hydraulic water distribution network analysis

2.2.1. Short history of hydraulic network analysis

Basic hydraulic principles were already known by ancient civilizations (e.g., Egyptians, Greeks, Romans, . . .) millennia ago (see Mala-Jetmarova et al. 2015). However, this knowledge has been based more on experience rather than on scientific principles, hence, these civilizations were not able to mathematically describe the complexity of WDS with satisfactory models (Crouch 1993). It took till the 17th century until the first hydraulic principles were formulated and, additionally, this mathematical description evolved slowly. According to Walski 2006, Castelli discovered the continuity principle[10], his student Torricelli showed the relationship between velocity and head, in 18th hundreds Newton and Bernoulli[11] developed the principles of fluid flow and at the end of the 19th century Reynolds investigated laminar and turbulent flow regimes (Reynolds 1883). Darcy-Weisbach and Hazen-Williams head loss equations date back to 1845 respectively 1906 (Mala-Jetmarova et al. 2015).

In the early 20th century, the increasing complexities of WDSs made precise estimations of flows and pressures necessary. The quest for methods that analyze entire networks gave birth to the field of WDN analysis (Ramalingam et al. 2002). The former described hydraulic principles built the foundation for the analysis of whole WDS. To calculate flows, head losses and pressures for an entire WDN, a large set of non-linear equations has to be solved. These equations are based on physical conservation laws, e.g., conservation of mass, flow and energy. Such calculations were impossible until recent advantages in computer modeling were achieved (Walski 2006). Ormsbee 2008 divides the evolution of modeling of pipe networks in three periods: (i) the pre-computer age (late 1800s to the 1960s), (ii) the dawn of the computer age (in the 1960s around 1957) and (iii) the advent of advanced computer methods (1960s to present).

(i) During the pre-computer age different methods were used to retrieve flows and pressure in pipe networks ranging from *graphical* methods, methods using *physical* analogies, e.g., and finally *mathe-*

[10] Originally discovered from no less a figure than Leonardo da Vinci already in the 15th century, but unfortunately forgotten by mankind (Mala-Jetmarova et al. 2015)

[11] Bernoulli's equation was actually developed by Euler around 1750.

matical methods (Ormsbee 2008). *Graphical* methods involved preparation of curves of discharge vs. head loss of pipes and complex arrangements of pipes on transparent thin celluloid and subsequently combining this graphs to shifting and juggling this curves (Ramalingam et al. 2002). Freeman's graphical method (Freeman 1892) dominated this period. *Physical* methods made use of the similarities between electrical networks and WDN. In 1934 at MIT, Camp and Hazen 1934 applied their electric network analyzer consisting of a bunch of resistors (corresponding to water pipes) and electrical "buses" with flexible-lead schemes (corresponding to network junctions) on WDSs. The MIT network analyzer has been flexible enough to represent any desired pipe network by arranging the resistors in a specific way (Ramalingam et al. 2002). The only serious problem has been that voltage drop in a resistor differs from the head loss in water pipes. Whereas the drop of voltage is directly proportional to the current flow, the head loss in water pipes is an exponential function of discharge. That is why years followed of developing resistor elements showing the desired characteristics of water pipes. Finally, McIlroy developed the direct-reading network analyzer using non-linear transistors providing a method to visualize and analyze WDSs in 1950 (McIlroy 1950). *Mathematical* methods were already applied since the advent of Hazen-Wiliams equation in 1906, which provided engineers with an easy way to decompose or aggregate composite systems of pipes into a hydraulically equivalent single pipe (Ramalingam et al. 2002). In 1936, Hardy Cross developed a mathematically method for solving pressure and flows in closed looped WDSs (Cross 1936). Interestingly, this method was initially developed for performing moment distribution analysis for statically determinate structures. Cross actually presented two methods: the node and the loop method—because the convergence of the node method was not satisfactory, the loop method gained more popularity and is what is now known as the "Hardy Cross Method" (Ramalingam et al. 2002).
(ii) The Hardy Cross Method, which was intentionally not developed for computer simulations, has been the best candidate for such computer applications (Ormsbee 2008) during the second period, the dawn of the computer age. Hoag and Weinberg 1957 were the first solving hydraulic network equations with the Hardy Cross method on a digital computer. Already in the same year, first companies started to sell pipe network analysis programs (Ramalingam et al. 2002). The first models were just able to solve steady-state hydraulic problems on large mainframe computers with punch-card input (Walski 2006).
(iii) Because of the bad convergence properties of the Hardy Cross method, more advanced computer methods—benefiting from advantages of modern computers ability to efficiently solve matrix problems (Walski 2006)—were developed in the subsequent decades (Mala-Jetmarova et al. 2015), e.g., the simultaneous node method by Martin and Peters 1970, the simultaneous loop method by Epp and Fowler 1970, the linear method by Wood and Charles 1972 and the gradient method by Todini and Pilati 1987. The last method had been adopted in the development of the most widespread hydraulic simulation package EPANET (Rossman 2000). Nowadays, EPANET is used extensively in conjunction with various optimization techniques to solve numerous optimization problems resulting in the field of WDS analysis (Mala-Jetmarova et al. 2015).

2.2.2. Different hydraulic models for different purposes

The purpose of a mathematical hydraulic model is to simulate the behavior of a WDS given different operating conditions (see Kapelan 2002, p. 73) in a pressurized pipe network. For that reason, a hydraulic model computes flows and head losses at all n_p links and all unknown pressures at all n_n nodes in a WDN. This is achieved by simultaneously solving a large set of non-linear equations that describe different physically principles, e.g., conservation of mass and energy.

Every mathematical model is an approximation of the real-world. The better a model should describe reality, the more mathematically complex it becomes since the more physical effects have to be added to the model. Hence, mathematical modeling of WDSs is always finding a compromise between simplicity of use and accuracy of prediction (B. S. Jung and Karney 2016). Additionally, the less simple a model becomes, the more computational expensive it gets to solve the more complex mathematical equations. Finally, the decision is up to the modeler if a higher computational price is worth a more complex model (B. S. Jung and Karney 2016), always keeping in mind that no matter how complex a mathematical model becomes, it is still only an approximation of the real world (Kapelan 2002).

Four types of different models exists for WDS analysis according to Kapelan 2002. The following four subsections will introduce these models, starting from the simplest one (steady flow models) and increasing their complexity until reaching the unsteady incompressible flow models (see Figure 2.15).

The main focus of this thesis is optimization in WDSs on the example of model-based leak localization. The time scales for leak localization are much longer than the time frame of, for example, control valve optimization. Thus, the focus will be on (quasi-)steady flow models, which will be presented in more detail compared to unsteady models in the following sections.

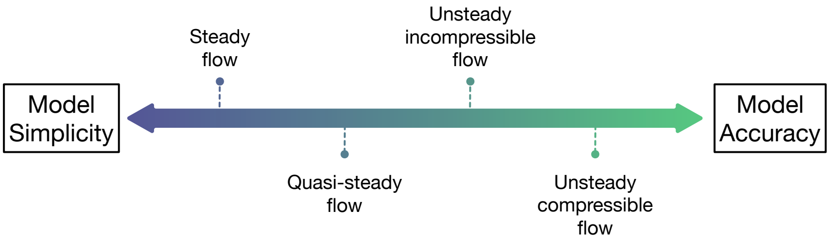


Figure 2.15.: Simplicity versus accuracy of different hydraulic models (based on B. S. Jung and Karney 2016, Figure 1 (a))

2.2.2.1. Steady flow models

The steady flow model, also called steady-state simulation model, is the biggest simplification of a WDS. Mathematically, it is the easiest to formulate and the least expensive one to solve in terms of computational costs (see Kapelan 2002). EPANET (Rossman 2000) uses this type of hydraulic model for performing steady-state simulations. In EPANET, a hydraulic model consists of links (pipes, pumps and valves) and nodes (junctions, tanks and reservoirs). Junctions are nodes which connect pipes and where water can leave network (customer, leaks). As mentioned already before, the aim of hydraulic model is to compute all flows and head losses at links and pressures at nodes in a network. The hydraulics of a WDN are completely defined by the pipe parameters (e.g. roughness, diameter, length), the demands at junctions, the hydraulic heads at fixed head nodes (tanks, reservoirs), and, of course, the topology of the WDN. The network's topology is mathematically defined by the incidence matrix $A \in \mathbb{R}^{n_p \times n_n}$ (see for example Abraham and Stoianov 2016)

$$A_{ij} = \begin{cases} -1 & , \quad \text{if pipe } i \text{ leaves node } j \\ +1 & , \quad \text{if pipe } i \text{ enters node } j \\ 0 & , \quad \text{otherwise} \end{cases} \qquad . \tag{2.51}$$

This incidence matrix relates the links to the nodes in the network.

Note that there exists a second matrix relating links to nodes in a network—the adjacency matrix A'_{ij} for a not weighted graph. The difference between the adjacency matrix and the incidence matrix is that the adjacency matrix is not directed. The relation between those matrices is:

$$A'_{ij} = |A_{ij}| \quad . \tag{2.52}$$

Physical principles behind the steady flow network calculations are two conservation laws, the conservation of energy respectively the conservation of mass. Due to the assumption that the fluid is incompressible, the conservation of mass is equal to the conservation of flow (Abraham and Stoianov 2016).

The conservation of energy states that difference in energy between two points in a network is equal to the friction loss along the path between them. Thus, the head loss Δh_{jk} across a pipe i connecting node j with node k is represented in following way (Abraham and Stoianov 2016)

$$\Delta h_{jk} = h_j - h_k = r_i |q_i|^{n-1} q_i \quad , \tag{2.53}$$

where h_j is the head at node j and h_k at node k, q_i is the flow through the pipe, r_i is its resistance coefficient and n is a number depending on the head loss equation that is used. The resistance coefficient r_i is defined as (Elhay and Simpson 2011)

$$r_i = \frac{\alpha L_i}{C_i^n D_i^m} \quad , \tag{2.54}$$

where L_i, C_i and D_i are the length, roughness and diameter of pipe i. The coefficients α and m are again depending on the head loss model.

In the Hazen-Williams (HW) head loss model, the coefficient are defined as $\alpha = 10.67$, $n = 1.852$ and $m = 4.871$, which gives the HW resistance coefficient

$$r_i^{HW} = \frac{10.67 \cdot L_i}{C_i^{1.852} D_i^{4.871}} \quad . \tag{2.55}$$

For the Darcy-Weisbach (DW) head loss model, in addition, the flow regime has to be distinguished (Simpson and Elhay 2011). The flow regime is given by the Reynolds number R which is defined as follows

$$R = \frac{V_i D_i}{\nu} \quad . \tag{2.56}$$

ν is the kinematic viscosity of water, V_i is the average velocity in pipe i and D_i is again its diameter. $R < 2000$ represents laminar flows, whereas all flows with $R > 4000$ are turbulent. In between ($2000 \geq R \geq 4000$) lies the regime of transitional flow. It has to be noted that the flow in pressurized WDSs is always turbulent, except in the perimeter of the system, where due to low demands, laminar and or even stagnant flows can occur (according to Walski et al. 2003, p. 28)

The Darcy-Weisbach head loss formulas for the different regimes are (according to Simpson and Elhay 2011, equation 1 and 2)

$$\Delta h_{jk} = \begin{cases} r_i^{DW} q_i & \text{for laminar flow} \\ r_i^{DW} |q_i|^{n-1} q_i & \text{for turbulent flow} \end{cases} \quad . \tag{2.57}$$

Note that $n = 1$ (resulting in $|q_i|^{1-1} = |q_i|^0 = 1$) for laminar and $n = 2$ for turbulent flow in this equation. The DW resistance factor r_i^{DW} is defined in the following way (Simpson and Elhay 2011)

$$r_i^{DW} = \begin{cases} \frac{128\nu}{\pi g}\frac{L_i}{D_i^4}, & \text{for laminar flow} \\ \frac{8}{\pi^2 g}\frac{L_i f_i}{D_i^5} & \text{for turbulent flow} \end{cases} \quad . \tag{2.58}$$

ν is again the kinematic viscosity of water, g is the gravitational acceleration constant, f_i is the Darcy-Weisbach friction factor, which is a function of the pipe's relative roughness ε_i/D_i, where ε_i is the roughness of the pipe, and the Reynolds number R. It has to be noted that for laminar flows r_i is independent of the pipe's flow q_i, while for turbulent flows r_i depends on q_i.

Through bringing the term $r_i|q_i|^{n-1}$ into matrix form by building a diagonal matrix $G(\boldsymbol{q}) \in \mathbb{R}^{n_p \times n_p}$ (Abraham and Stoianov 2016)

$$G_{ii}(q_i) = r_i|q_i|^{n-1}, \qquad i = 1,\ldots,n_p \qquad , \tag{2.59}$$

together with the incidence matrix A from equation (2.51), the energy equation (2.53) can be formulated for the whole network by

$$G(\boldsymbol{q})\boldsymbol{q} = A\boldsymbol{h} + A_F\boldsymbol{e}_F \qquad , \tag{2.60}$$

where $\boldsymbol{q} \in \mathbb{R}^{n_p}$ is vector of unknown flows, $\boldsymbol{h} \in \mathbb{R}^{n_n}$ represents the unknown heads. Additionally in equation (2.60), fixed head nodes are defined through the fixed head node incidence matrix A_F and the vector of fixed head elevations $\boldsymbol{e}_F$, with n_f being the number of the fixed head nodes (reservoirs plus tanks) in the system.

A second type of losses exist in WDS modeling—the minor head losses or local losses–caused by turbulences at bends and fittings (Rossman 2000). The minor loss h_M is defined by the product

$$h_M = k_i\left(\frac{v_i^2}{2g}\right) \qquad , \tag{2.61}$$

where k_i is the pipe's minor loss coefficient, v_i is the velocity in pipe i and g is again the gravitational acceleration constant.

The conservation of flow states that the inflow at every node equals the outflow, in other words, the net flow in a node is zero (Walski et al. 2003). If there is a customer demand at this node, the sum of all flows of all pipes that enter or leave node j must equal its demand d_j

$$\sum_{\forall i \mapsto j} q_i = d_j \qquad . \tag{2.62}$$

$\forall i \mapsto j$ defines all links i that are connected to node j.

EPANET performs demand-driven simulations, where the demand is assumed to be known. It has to be noted that there also exist pressure-driven simulation approaches, where nodal demand is a function of the pressure p_i at the node ($d_i(p_i)$) (see for example Giustolisi and Laucelli 2011; Giustolisi, Savic, et al. 2008; Pathirana 2011, among others). The pressure at node i is given by the head h_i minus the node's elevation e_i ($p_i = h_i - e_i$). Additionally, there exist a possibility to add pressure dependent emitters in EPANET, normally used for simulating the pressure dependency of leaks. The emitter equation in EPANET is defined as follows

$$Q_i = c_e \cdot p_i^{e_e} \qquad , \tag{2.63}$$

where Q_i is the emitter (leakage) outflow at node i, c_e is the emitter coefficient, e_e is the emitter exponent and p_i is the pressure at node i.

For the whole network, the flow conservation equation can be written in matrix form (Sanz Estapé 2016)

$$A^{\intercal}\boldsymbol{q} = \boldsymbol{d} \quad . \tag{2.64}$$

$\boldsymbol{d} \in \mathbb{R}^{n_n}$ is the vector of all nodal demands in the system and $A^{\intercal}$ is the transpose of the incidence matrix A ($A^{\intercal}_{ij} = A_{ji}$).

Equations (2.60) and (2.64) can be put together leading to following matrix equation (see Todini and Pilati 1987)

$$f(\boldsymbol{q},\boldsymbol{h}) = \begin{pmatrix} G(\boldsymbol{q}) & A \\ A^{\intercal} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{h} \end{pmatrix} + \begin{pmatrix} A_F \boldsymbol{e}_F \\ -\boldsymbol{d} \end{pmatrix} = 0 \quad . \tag{2.65}$$

This is the set of nonlinear equations that define the steady-state flow conditions (Abraham and Stoianov 2016). The non-linearity of the equation is given by the implicit dependency of matrix $G(\boldsymbol{q})$ on $\boldsymbol{q}$. Since all head losses are monotonically increasing functions for all pipes, it can be proved that a unique solution of the system exists (see Todini and Pilati 1987). The nonlinear set of equations $f(\boldsymbol{q},\boldsymbol{h}) = 0$ can be iteratively solved using Newton's method (see for example Strang 1986)

$$\nabla f(\boldsymbol{x}^k)(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) = -f(\boldsymbol{x}^k) \quad , \tag{2.66}$$

which results for equation (2.65) in (see for example Abraham and Stoianov 2016)

$$\begin{pmatrix} NG(\boldsymbol{q}^k) & A \\ A^{\intercal} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{q}^{k+1} - \boldsymbol{q}^k \\ \boldsymbol{h}^{k+1} - \boldsymbol{h}^k \end{pmatrix} = - \begin{pmatrix} G(\boldsymbol{q}) & A \\ A^{\intercal} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{q}^k \\ \boldsymbol{h}^k \end{pmatrix} + \begin{pmatrix} -A_F \boldsymbol{e}_F \\ \boldsymbol{d} \end{pmatrix} \quad . \tag{2.67}$$

A solution for flows and heads is found by repeatedly solving this equation system.

2.2.2.2. Quasi-steady flow models

Quasi-steady flow models are the most extensively used models by WUs and engineering firms for planning the basic components of system operation, design and expansion (B. S. Jung and Karney 2016). In EPANET they are called Extended Period Simulations (EPSs). These models simulate changing states of a WDS over time, although, each time step in these models is a steady-state simulation as described in section 2.2.2.1. The variation of demand and heads at reservoirs is given in EPANET in the form of time patterns (demand patterns for demands, respectively, head patterns for reservoir heads). Subsequently, the water levels in tanks have to be updated at each time step according to the water that enters or leaves the tank. The change of the water level over time $(dh_j(t)/dt)$ at tank j defined through following equation (Kapelan 2002)

$$A_j \frac{dh_j(t)}{dt} = \sum_{\forall i \mapsto j} q_i(t) \quad , \tag{2.68}$$

where A_j is the cross section area of tank j, $h_j(t)$ is the head as a function of time t, $q_i(t)$ is the flow in pipe i connected to tank j ($\forall i \mapsto j$). This is an extension of the flow conservation law in equation (2.62) for tanks; the left side of equation (2.68) is the change in the water volume in the tank and equals all the flows that enter or leave the tank at the right side of the equation.

The ordinary differential equation (2.68) is solved using the forward Euler method (see Kapelan 2002, p. 77)

$$A_j \frac{(h_j(t+\Delta t) - h_j(t))}{((t+\Delta t) - t)} = \sum_{\forall i \mapsto j} q_i(t) \qquad , \tag{2.69}$$

where Δt is the increment in time to the next calculated time step $t + \Delta t$. This results in following update formula for the tank levels for the next time step

$$h_j(t+\Delta t) = h_j(t) + \frac{\Delta t}{A_j} \sum_{\forall i \mapsto j} q_i(t) \qquad . \tag{2.70}$$

The tank level at the initial time step ($t = 0$) is assumed to be known.

2.2.2.3. Unsteady incompressible flow models

Unsteady incompressible flow models, the most rarely used in practice (B. S. Jung and Karney 2016), are also known as slow transient models or rigid water column models. This class of models assume that the fluid is incompressible, hence, neglecting convective acceleration (B. S. Jung and Karney 2016). Nevertheless, inertia effects of water are included in the model. This class of models are important, when despite the incompressibility of the fluid, mass oscillations exist in the network over time (Sanz Estapé 2016). For example, this happens for slow valve closures (according to Fox 1977). To include inertia effects, the energy conservation principle in equation (2.53) has to be extended

$$\Delta h_{jk} = h_j - h_k = r_i |q_i|^{n-1} q_i + \frac{4L_i}{g \pi D_i^2} \frac{dq_i}{dt} \qquad , \tag{2.71}$$

resulting in an Ordinary Differential Equation (ODE), where the last term accounts for the acceleration (dq_i/dt) of the water column in pipe i. L_i and D_i are the pipe's length and diameter and g is again the gravitational acceleration constant. Since in this model fluids are assumed to be incompressible, the flow continuity equation (equation (2.62)) stays the same as for the static model. A formulation of the network equations resulting in unsteady incompressible flow models for whole WDSs can be found in Kaltenbacher et al. 2017.

2.2.2.4. Unsteady compressible flow model

Rigid models do not realistically represent rapid flow changes. As the operational time of devices decreases, pressure in these models increases without bounds (B. S. Jung and Karney 2016). Since compressibility—the physical effect limiting infinite increasing pressures—is not modeled in slow transient models, more complicated models are needed. These models are called water hammer or (full) transient models.

The physical principles behind the water hammer model is the same as for the other already discussed approaches. Therein, again two conservation laws have to be fulfilled—the conservation of mass as wells as the conservation of energy. The conservation of energy is given in form of the dynamic equation, which is also called the equation of motion, and is mathematically formulated as follows (see Wylie and Streeter 1978, p.19 eq. 2.8)

$$\frac{\partial h(x,t)}{\partial x} + \frac{1}{\lambda_i} \frac{\partial q(x,t)}{\partial t} + J = 0 \tag{2.72}$$

with the constant λ_i for pipe i

$$\lambda_i = \frac{\pi g}{4} D_i^2 \qquad . \tag{2.73}$$

The hydraulic head $h(x,t)$ as well as the flow $q(x,t)$ in pipe i are now functions of space x and time t. The friction slope J can be split into a steady (J_s) and and an unsteady (J_u) term

$$J = J_s + J_u \qquad . \tag{2.74}$$

The steady friction slope J_s is again represented through Hazen-Williams or Darcy-Weisbach formula as already mentioned in equation (2.53). The unsteady friction term J_u accounts for the change in the velocity profile and is defined as follows

$$J_u = g\,\lambda_i^2\; q(x,t)\frac{\partial q(x,t)}{\partial x} \qquad . \tag{2.75}$$

The second conservation law, the conservation of mass, is formulated in following way (see Wylie and Streeter 1978, p.23 eq. 2.28)

$$\frac{\partial h(x,t)}{\partial t} + \frac{c^2}{\lambda_i}\frac{\partial q(x,t)}{\partial x} = 0 \qquad . \tag{2.76}$$

This is the continuity equation in the transient model formulation, where parameter c represents the wave speed.

The equations representing the continuity of mass and energy (equations (2.72) and (2.76)), are a pair of quasi-linear hyperbolic differential equations, which have to be formulated for each pipe in a WDS. Unfortunately, this pair of equations cannot be solved analytically (Fox 1977). Hence, the price of the most accurate hydraulic model comes in form of the most complicated mathematical formulation as well as the highest computational burden.

2.2.3. What hydraulic model for model-based leak localization

Despite their complexity, the use of transient models for model-based leak detection and localization (see Colombo, Lee, et al. 2009, for a short review on different methods) has gained massive popularity in the last decades (Puust et al. 2010). The reason for that is that a large amount of data can be recorded in short time, resulting in always over-determination of the inverse leak localization problem (Pudar and Liggett 1992) and hence promising high leak localization accuracies. Additionally, the signals are less sensitive to accurate estimation of pipe friction factors compared to steady-state models (Puust et al. 2010). Nevertheless, the advantages does not come without drawbacks. First, other sources of uncertainties besides pipe friction factor estimates heavily influence the goodness of the results (Puust et al. 2010). For example, pressure waves get reflected by unknown components and bends in a pipe system. In real-world underground systems, these informations can't be known with the demanded precision by the models. That is why transient methods mainly focus on leak localization in single pipeline systems or very simple systems consisting of few pipes only (Puust et al. 2010). Second, expensive pressure sensors are necessary which are capable of obtaining accurate measurements in very high time resolutions to resolve pressure waves accurately. The reason for that is the high propagation velocities of perturbations in water resulting from high wave speeds of water in pipes. Additionally, due to the high time resolution, the transmitted data packages are also huge. Third, the computational complexity and hence the computational cost of solving the transient model for a real-world WDS is very high compared to steady state models. That is another reason why they are not applicable in real-world WDN. Potentially, with growing computer power and more affordable data communication as well as cheaper high-performance sensors, leak localization with transient models may render possible in future.

Since nowadays the time resolution of common used pressure sensors is not high enough to perform transient-based localization and the demanded computer power is not available, a step back in model complexity to more simple models for leak localizations is made. In this thesis **steady** and **quasi-steady** flow models will be used as described in Sections 2.2.2.1 and 2.2.2.2. This has been already proposed nearly three decades ago by Pudar and Liggett 1992, but was found as not suitable for leak localization by the authors stating that "*in the case of leak detection by static methods, an inverse program is unlikely to provide the definitive results that would supplant more conventional methods*". However, later on in this thesis, we will see that this simple approach can indeed be satisfactory for finding leaks.

2.2.4. Graph theory and water distribution networks

[12] The concept of a network is describing an object by its elements and connections between these elements (Brandes and Erlebach 2005). The mathematical discipline of describing networks is called graph theory. In view of this, a graph $G = (V,E)$ is defined as an abstract object formed by a set of vertices $V(G)$ (or nodes) and a set of edges $E(G)$ (or links) which connect pairs of vertices. Two vertices are called neighbors (or adjacent) if they are connected by an edge. Directed and undirected graphs exist. In directed graphs, two vertices are only connected in one direction. The graphs in this thesis are always supposed to be undirected. It is possible to associate numerical values to edges as weights ω ($E(G) \rightarrow \omega$). Depending on the context, these weights can describe different properties (e.g. cost, distance, travel time, capacity, interaction strengths).

A WDN can be described as such a graph. Vertices are the WDN's nodes (junctions, tanks and reservoirs), edges are the WDN's links (pipes and valves). Graph theory has already many fields of application in WDS analysis as, for example, to improve the efficiency of hydraulic simulations (Elhay and Simpson 2011), WDN optimization (Zheng et al. 2013), WDS calibration (Sophocleous et al. 2016), resilience analysis (Herrera et al. 2016) or WDN sectorization (Tzatchkov et al. 2006). In this thesis, the graph theoretic description of a WDN will be used for efficiently computing shortest paths between nodes in the system with the shortest path algorithm of Dijkstra 1959. The length of the pipes l will be used as edge weights ω for these calculations.

2.3. Probability theory, statistics and time series analysis in a nutshell

This section presents a short and concise overview of the statistical techniques used in this thesis.

2.3.1. Monte Carlo simulation

Since all hydraulic modeling approaches which were introduced in Section 2.2 describe deterministic forward models, hydraulic simulations rely heavily on model input parameters. Unfortunately, these

[12] Mathematical definitions in this section are taken from Brandes and Erlebach 2005.

input parameters are usually fraught with uncertainties and these uncertainties in input parameters will again produce uncertain model outputs. To estimate propagations of these uncertainties in the model, different mathematical approaches exist, e.g. the First-order-second moment method[13], MCS or Latin hypercube sampling (D. S. Kang et al. 2009), among others. In this thesis, MCS are chosen to calculate the output uncertainties since this technique converges to the exact estimates as long as a large number of parameter sets is simulated. Furthermore, the implementation of this technique is easy and straight-forward and it enables parallel computations similar to GAs described in Section 2.1.2.2.

MCS generate a large number of input parameter set realizations by randomly drawing samples from probability distributions associated with the model input uncertainties. Subsequently, a hydraulic model is used to evaluate all these realizations producing output parameter sets, which then represent in their entirety the model output uncertainties. A large number of input parameter sets has to be evaluated to get good estimates of the output uncertainties. Latin hypercube sampling can provide remedy by more sophisticated sampling of the model input spaces.

MCS will be used later on in this thesis in the Chapter 4 to incorporate uncertainties in optimal sensor placement algorithms.

2.3.2. Markov-Chain-Monte-Carlo methods

MCMC is a class of sampling methods specifically useful for Bayesian applications, because they can be used to sample arbitrary probability density functions and to compute marginal densities, posterior distributions and evidences (Ruanaidh and Fitzgerald 1996) (see for example Section 2.3.6.3). The mathematical definitions and the description of the Metropolis Hastings algorithm follow Ruanaidh and Fitzgerald 1996 in this section.

A Markov chain is the first generalization of an independent process where each step only depends on the previous step

$$P(x_{N+1}|x_1x_2\dots x_N) = P(x_{N+1}|x_N) \quad . \tag{2.77}$$

The random variables are defined in a certain range in a continuous or discrete space—known as the state space. A Markov chain begins at a certain point in this state space and moves through it with a defined transition probability T between those states

$$P_{N+1}(x) = \sum_{x'} P_N(x')T_N(x',x) \quad . \tag{2.78}$$

If T_N is not dependent on N, the Markov chain is homogeneous. Two properties must be fulfilled for MCMC: (i) the existence of an invariant distribution $\pi(x)$ (which can be formulated more restrictive for a Markov chain to be reversible) and (ii) the Markov chain has to be ergodic.

(i) The reversibility criterion is mathematically formulated as

$$\pi(x)T_N(x,x') = \pi(x')T_N(x',x) \quad . \tag{2.79}$$

Hence, the probability to travel in one direction is the same as the probability for the reverse direction.

[13] First-order-second-moment is a first order approximation method that propagates uncertainties in a system by matrix multiplications of model input uncertainties with corresponding sensitivity matrices—which are linearized mappings of network equations.

(ii) Ergodicity means that regardless of the initial distribution the Markov chain has to converge to the invariant distribution $\pi(x)$ for $N \to \infty$

$$\lim_{N\to\infty} P_N(x) = \pi(x) \quad . \tag{2.80}$$

Furthermore, a certain number of initial states in the so-called burn-in phase should be discarded.

Metropolis-Hastings The most famous MCMC is without doubt the Metropolis-Hastings formulation. It is also the standard form of the simulated annealing algorithm—a stochastic optimization algorithm (see Section 2.1.2.2). For this algorithm, a probability density function $P(x)$ is assumed in functional form $y = P(x)$—no normalization of the density function is needed. A next realization of the probability function is produced by adding random perturbations to the current realization

$$y_i = x_i + \xi \quad , \tag{2.81}$$

where ξ is randomly drawn from the proposal density $s(\xi)$. For the next step in the Markov chain, the proposed step is either accepted $x_{i+1} = y_i$, or rejected. If it is rejected, the next step is equal to the old step $x_{i+1} = x_i$. The acceptance probability is defined through

$$A(x_i, y_i) = \min\left(1, Q(x_i, y_i)\right) \quad , \tag{2.82}$$

with

$$Q(x_i, y_i) = \frac{p(y_i)T(y_i|x_i)}{p(x_i)T(x_i|y_i)} \quad . \tag{2.83}$$

$T(x|y)$ is usually the proposal density $s(\xi)$. If this density is symmetric, then $T(y_i|x_i) = T(x_i|y_i)$.

If u is a uniform random variable between 0 and 1, then the Metropolis-Hastings algorithm can be written as

$$x_{i+1} = \begin{cases} y_i & \text{if} \quad p(y_i)T(y_i|x_i) > u \cdot p(x_i)T(x_i|y_i) \\ x_i & \text{else} \end{cases} \quad . \tag{2.84}$$

The Markov chain usually moves through the state space like a random-walker, hence, each sample drawn from the chain is highly correlated with previous samples. Additionally, the random walk behavior has the drawback that the time to move a certain distance is proportional to the square of this distance. For that reason, using random walks is very inefficient in exploring the parameter space—starting points far from the probability mass might never converge in reasonable time. That is why more sophisticated methods have been developed.

Hamiltonian-Monte-Carlo (HMC) For example, HMC (also called hybrid Monte Carlo) avoids the random walk behavior by taking a series of steps that is informed by the first derivatives information. This results in much higher convergence speeds especially for high-dimensional target distributions (Hoffman and Gelman 2014). However, this class of algorithms is sensitive to the step size and the number of steps of the algorithm.

No-U-Turn Sampler (NUTS) No-U-Turn Sampler (NUTS)—an extension to HMC developed by Hoffman and Gelman 2014—promise remedy. This algorithm is at least as efficient as HMC without the necessity of any tuning or user intervention, thus, it functions in a complete automatic way. That is why NUTS is used in this thesis when performing MCMC.

2.3.3. Tukey's fence test

This method represents a simple statistical test to recognize outliers in datasets. Tukey's test identifies outliers with help of the interquartile range Δq

$$\Delta q = q_3 - q_1 \quad . \tag{2.85}$$

q_i is the i-th quartile of the dataset. If a measurement is outside the range

$$[q_1 - k\Delta q, q_3 + k\Delta q] \quad , \tag{2.86}$$

then this data point is defined as an outlier. $k = 1.5$ defines the boundaries for (normal) outliers and $k = 3.0$ defines the boundaries for far outliers. Most people are already familiar with Tukey's fence test without even knowing—the whiskers around the ends of box-plots represent Tukey's outlier boundaries.

Tukey's outlier test will be used later on for determining unusual customer demand during MNF hours and serves as a simple approach for detecting leak in real-world WDS data.

2.3.4. Goodness of fit statistics

GoF statistics will be used to determine the best model order of ARMA models (described in Section 2.3.5.5) as well as for identifying the ideal cost-benefit functions for the optimal sensor placement algorithm described in Section 4.2.2.4. GoF statistics include Akaike's Information Criterion (AIC) and Bayes Information Criterion (BIC). Both criteria are based on χ^2 statistics

$$\chi^2 = \sum_{i=1}^{n} (y_i - f(x_i))^2 \quad , \tag{2.87}$$

where y_i is the data point on position x_i, $f(x_i)$ is the value calculated by the fitting function and n is the total number of data points. Additionally, the degrees of freedom ν, respectively the number of function parameters $(n-\nu)$ must be considered. For example, the reduced χ^2_r statistics derived from equation (2.87) aiming to take the number of parameters into consideration in following way

$$\chi^2_r = \frac{\chi^2}{\nu} \quad . \tag{2.88}$$

The AIC takes the function parameters into account by punishing fitting functions with many parameters. It is defined through following equation

$$\mathrm{AIC} = n \ln \left(\chi^2/n\right) + 2(n-\nu) \quad . \tag{2.89}$$

BIC penalizes the number of parameters of the fitting function even more than AIC

$$\mathrm{BIC} = n \ln\left(\chi^2/n\right) + (n-\nu)\ln(n) \quad , \tag{2.90}$$

and, hence, prefers even smaller parametrized models. The smaller the AIC or the BIC value are, the better a certain model will describe the data.

2.3.5. Time series analysis techniques

The following short section addresses some Time Series Analysis (TSA) techniques. The mathematical definitions within this section are based on standard time series analysis methodologies and can be found, for example, in (Shumway and Stoffer 2006).

2.3.5.1. Mathematical notation and definitions

In the field of TSA, a single data point of a time series is assumed to be the realization of an underlying stochastic process. Adjacent observations are not independent and identically distributed, they are correlated with previous observations (Shumway and Stoffer 2006). This effect distinguishes the field of TSA from conventional statistical analysis. Before these techniques are explained in more detail, a standardized notation is defined to provide the reader with clarity throughout this thesis.

A series of data points $\{x_i\}$ ordered by time $\{x_1, x_2, \ldots x_N\}$ is called a time series x_t of length N. Note that throughout this thesis there will be no distinction in the notation between the stochastic process underlying a time series and its realization. Both will be defined through x_t. The mean $\bar{x}_t$ over the whole time series x_t is defined through

$$\bar{x}_t = \frac{1}{N} \sum_{i=1}^{N} x_i \quad . \tag{2.91}$$

Consequently, the sample standard deviation σ_t of the time series is written as

$$\sigma_t = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x}_t)^2} \quad . \tag{2.92}$$

Similarly to equation (2.91), the median $\tilde{x}_t$ is defined in following way

$$\tilde{x}_t = \mathrm{median}\left(\{x_i\}\right) \quad . \tag{2.93}$$

The median is more robust to outliers than the mean. A measurement similar to the sample standard deviation, but also more robust to outliers, is the median absolute deviation (mad_t)

$$\mathrm{mad}_t = \mathrm{median}\left(\{x_i - \tilde{x}_t\}\right) \quad . \tag{2.94}$$

Frequently, time series posses one or more repeating components with fixed period lengths P. Hence, cyclic operators are defined. For example, the cyclic mean with period length of P is described in following way

$$\bar{x}_{t|P} = \frac{1}{N_P} \sum_{i=1}^{N_P} x_{P(i-1)+j} \qquad \forall j = 1, 2 \ldots, P \quad . \tag{2.95}$$

where $N_P = \lfloor \frac{N}{P} \rfloor$ is the number of full periods in the time series. Definitions of the cyclic sample standard deviation $\sigma_{t|P}$, the cyclic median $\tilde{x}_{t|P}$ and the cyclic median absolute deviation $\mathrm{mad}_{t|P}$ follow the same procedure.

Moving averages are used to suppress random fluctuations in measurement data when illustrating them in figures. The moving average MA_w operator with a window size w of a time series x_t is defined in following way

$$\mathrm{MA}_w(x_t) = \frac{1}{w} \sum_{i=0}^{w-1} x_{t-i} \quad . \tag{2.96}$$

2.3.5.2. Autocorrelation functions

The AutoCorrelation Function (ACF)—the correlation of a time series with a delayed copy of itself—is defined in following way

$$\rho(x_s,x_t) = \frac{\gamma(x_s,x_t)}{\sqrt{\gamma(x_s,x_s)\gamma(x_t,x_t)}} \quad , \tag{2.97}$$

where x_s and x_t represent observations at time s and t, and $\gamma(x_s,x_t)$ is the auto-covariance function defined as follows

$$\gamma(x_s,x_t) = \mathrm{cov}(x_s,x_t) = E\left[(x_s - \bar{x}_s)(x_t - \bar{x}_t)\right] \quad . \tag{2.98}$$

E defines the expectation value or the average and $\gamma(x_t,x_t)$ is the variance of the time series ($\gamma(x_t,x_t) = \sigma_t^2$), if the variance does not change with time.

If x_t is a weakly stationary time series[14], then equation (2.97) can be written as (according to Shumway and Stoffer 2006)

$$\rho(h) = \frac{\gamma(x_{t+h},x_t)}{\sqrt{\gamma(x_{t+h},x_{t+h})\gamma(x_t,x_t)}} \quad . \tag{2.99}$$

The Partial AutoCorrelation Function (PACF) is defined as the autocorrelation with removed linear dependencies.

2.3.5.3. Deseasonalization

In Section 3.3.3.1, seasonal effects in water related measurement data will be introduced (e.g. daily and weekly seasonalities). This section deals with separating these seasonal effects from data to smooth time series. In general, seasonal components can be extracted from the measurement data by subtracting the cyclic mean (equation (2.95)) with P being the season's period length. Two different seasonalities exist in water related measurement data; (i) daily seasonality with a period length of $P = 24H$ and (ii) weekly seasonal effects with $P = 168H = 7D$. The daily seasonality is defined in following way

$$\hat{s}_t^1 = \bar{x}_{t|24H} \quad . \tag{2.100}$$

[14] A weakly stationary time series is a time series (i) with finite variance, (ii) without time dependency of its mean value and (iii) an auto-covariance function that only depends on the difference $|s-t|$ between time steps s and t (see for example Shumway and Stoffer 2006).

Additionally, this seasonal effect $\grave{s}_t^1(x_j)$ is normalized by subtracting the mean value over the whole season to get rid of additive effects

$$s_t^1 = \grave{s}_t^1 - \frac{1}{P}\sum_{i=1}^{P} \grave{s}_i^1 \quad . \tag{2.101}$$

The seasonal component s_t^1 over a whole time series is given by repeating the seasonal component

$$s_t^1 = s_{t\%P}^1 \quad , \tag{2.102}$$

where % represents the modulo operator. Subtracting s_t^1 from x_t results in time series $\grave{x}_t$

$$\grave{x}_t = x_t - s_t^1 \tag{2.103}$$

that is free of daily seasonal effects. The same procedure is applied to extract the weekly seasonality s_t^2 from $\grave{x}_t$ with period $P = 168H = 7D$

$$s_t^2 = \bar{\grave{x}}_{t|7D} - \frac{1}{P}\sum_{i=1}^{P} \bar{\grave{x}}_{i|7D} \quad . \tag{2.104}$$

Subsequently, the time series consisting of summing up the two seasonal effects plus the mean value of all measurements will serve as a first simple time series model[15]

$$\hat{x}_t = \overline{x}_t + s_t^1 + s_t^2 \quad . \tag{2.105}$$

2.3.5.4. Time series models, residuals and white noise

The aim of time series modeling is to reveal and extract all parts of a time series that obey some mathematical law until only a stochastic completely random component remains which can't be described by any deterministic model anymore. The residual r_t is defined as the observations x_t minus the time series model $\hat{x}_t$

$$r_t = x_t - \hat{x}_t \quad . \tag{2.106}$$

Ideally, if a time series $\hat{x}_t$ describes all underlying parts that can be modeled, the residuals r_t possess white noise properties.

White noise w_t is defined as an independent and identically distributed, uncorrelated random variable with zero mean $\overline{w}_t = 0$ and finite variance $\sigma_{t,\omega}^2 < \infty$. A special form of white noise is Gaussian white noise $w_t = \mathcal{N}(0, \sigma_{t,\omega}^2)$, but it has to be mentioned that white noise can have more general forms than a Gaussian distribution (e.g. heavy tail distributions).

r_t should be checked if they have white noise properties subsequently after time series modeling. This can be done by analyzing the ACF and PACF of r_t to check if r_t is uncorrelated, among others. Only if $r_t \in w_t$, the time series model describes the measurement data sufficiently.

[15] Note that time series models will always be formulated as $\hat{x}_t$ throughout this thesis. It emerges from the context which specific model is used.

2.3.5.5. Autoregressive-moving-average models

An AutoRegressive (AR) model of order p is based on the idea that current values x_t of a time series can be explained by the past p observations $x_{t-1}, x_{t-2}, \dots, x_{t-p}$ plus white noise w_t (Shumway and Stoffer 2006)

$$x_t = \sum_{i=1}^{p} \varphi_i x_{t-i} + w_t \quad , \tag{2.107}$$

where $\varphi_1 \dots \varphi_p$ are constants with $\varphi_p \neq 0$. The model in equation (2.107) is called an AR(p)—an autoregressive model of order p.

In contrary, the idea behind Moving Average (MA) models is that a linear combination of the last q white noise processes forms the current observation (Shumway and Stoffer 2006)

$$x_t = \sum_{i=1}^{q} \theta_i \, w_{t-i} + w_t \quad , \tag{2.108}$$

where $\theta_1 \dots \theta_q$ are constants with $\theta_q \neq 0$. The model in equation (2.108) is called a MA(q)—a moving average model of order q.

AutoRegressive-Moving Average (ARMA) models represent a class of models that are not sufficiently described either only by a MA processes nor solely by AR processes (Shumway and Stoffer 2006). Hence an ARMA(p,q) model is defined as a combination of both

$$x_t = c + w_t + \sum_{i=1}^{p} \varphi_i \, x_{t-i} + \sum_{i=1}^{q} \theta_i \, w_{t-i} \quad . \tag{2.109}$$

c is a constant, p is the order of the AR part, q is the order of the MA part, φ_i and θ_i are coefficients that have to be fitted, x_{t-i} is the time series value and ε_{t-i} is the error term i time steps ago.

Fitting φ_i and θ_i to the observations is an optimization problem which will be solved in this thesis by the BFGS algorithm—a famous deterministic optimization algorithm defined through a special formulation of GDA (see section 2.1.2.1.2).

2.3.5.6. State space models and Kalman filter

This section is again based on the book of (Shumway and Stoffer 2006) on time series analysis.

State Space Form The state space form of a model is defined as a system of two sets of linear equations. Suppose the model consists of a time series of vectors $\boldsymbol{\alpha}_t$ with m parameters—the state vectors—describing the current state of a system. For example, $\boldsymbol{\alpha}_t$ can describe meaningful but imperfectly measured physical variables, but not necessary. Typically, $\boldsymbol{\alpha}_t$ is not observed, thus, an observed variable x_t is introduced which corresponds to a data point x_t in TSA.

The first set of equations in the state space form is the **transition equation**—describing the evolution of the system

$$\boldsymbol{\alpha}_t \; = \; K\boldsymbol{\alpha}_{t-1} + \boldsymbol{v}_t \quad . \tag{2.110}$$

The second set of equations is the **measurement equation**—describing the relation between the system and the observation

$$x_t \;=\; \boldsymbol{b}^T \boldsymbol{\alpha}_t + u_t \quad . \tag{2.111}$$

K is a $(m \times m)$ matrix of constants, $\boldsymbol{b}$ is a $(m \times 1)$ dimensional vector, $\boldsymbol{v}_t$ is a vector of white noise processes of size $(m \times 1)$ and u_t is a scalar white noise process.

Advantages All ARMA models described in Section 2.3.5.5 can be translated into the state space form. The state space representation posses many advantages over the ARMA representation. Just to name a few examples of benefits: (i) the possibility for handling non-stationary processes, (ii) easy handling of missing observations and the possibility to impute them, (iii) computations with non-regular time series—that are time series which have irregular spaced time stamps, (iv) using multiple sensors of different types to estimate states (sensor fusion) and (v) highly efficient update and forecast formulas[16] using only information from the current observation without refitting the model to the whole dataset by utilizing Kalman filters.

ARMA in State Space Every ARMA(p,q) model—thus also any MA and AR model—can be represented in state space through following derivation. Let m be $m = \max(p, q+1)$, then by setting not used orders of coefficients to zero and applying $c = 0$ without loss of generality, one can rewrite equation (2.109) into

$$x_t \;=\; w_t + \theta_1 w_{t-1} + \ldots + \theta_{m-1} w_{t-m+1} + \varphi_1 x_{t-1} + \ldots + \varphi_m x_{t-m} \tag{2.112}$$

Then any order of ARMA model can be represented through defining the relationship between equation (2.112) and the sets of state space equations (2.110) and (2.111)

$$u_t = 0 \quad \text{and} \quad \boldsymbol{v}_t = \boldsymbol{d} \cdot \boldsymbol{w}_t \tag{2.113}$$

together with

$$\boldsymbol{b}_{(m \times 1)} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \;, \quad \boldsymbol{c}_{(m-1 \times 1)} = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_{m-1} \end{pmatrix} \;, \quad \boldsymbol{d}_{(m \times 1)} = \begin{pmatrix} 1 \\ \theta_1 \\ \vdots \\ \theta_{m-1} \end{pmatrix} \tag{2.114}$$

and the matrix

$$K_{(m \times m)} \;=\; \begin{bmatrix} \boldsymbol{c} & \mathbb{1} \\ \varphi_m & \boldsymbol{0} \end{bmatrix} \quad , \tag{2.115}$$

where the identity matrix is of dimension $\mathbb{1}_{(m-1 \times m-1)}$ and the zero vector is of dimension $\boldsymbol{0}_{(1 \times m-1)}$.

The state space representation of an ARMA(1,1) results, for example, in the transition equation

$$\begin{pmatrix} \alpha_t^1 \\ \alpha_t^2 \end{pmatrix} \;=\; \begin{bmatrix} \varphi_1 & 1 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \alpha_{t-1}^1 \\ \alpha_{t-1}^2 \end{pmatrix} \;+\; \begin{pmatrix} w_t \\ \theta_1 w_t \end{pmatrix} \tag{2.116}$$

[16] This is the most important aspect in this thesis why the state space formulation is used.

and the measurement equation

$$x_t = (1\,0) \cdot \begin{pmatrix} \alpha_t^1 \\ \alpha_t^2 \end{pmatrix} = \alpha_t^1 \tag{2.117}$$

The truth of these equations can be checked through substituting the second line of equation (2.116) in to equation (2.117). The first line results in

$$\alpha_t^1 = \varphi_1 \alpha_{t-1}^1 + \alpha_{t-1}^2 + w_t \quad . \tag{2.118}$$

The second line is equal to

$$\alpha_t^2 = \theta_i w_t \quad . \tag{2.119}$$

Substituting α_t^2 in equation (2.118) results in

$$\alpha_t^1 = w_t + \varphi_1 \alpha_{t-1}^1 + \theta_i w_{t-1} \quad . \tag{2.120}$$

Since $\alpha_t^1 = x_t$ (equation (2.117)) this is indeed the ARMA(1,1) model.

Kalman filter The Kalman Filter is an algorithm used to solve state space models in the linear case and was first derived by Kalman 1960. It can be used for (i) filtering, (ii) forecasting as well as (iii) smoothing of time series. The filter takes the state space formulation and defines prediction and updating equations from it. The Kalman filter evolves by predicting and updating the predictions of $\boldsymbol{\alpha}_t$. The best guess for an observation x_t at time t based on all former available information at time $t-1$ will be further written as $x_{t|t-1}$. Furthermore, the conditional variance $P_{t|t-1}$ of the prediction error $\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_{t|t-1}$ is defined as

$$P_{t|t-1} = E\left[(\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_{t|t-1})(\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_{t|t-1})^T\right] \quad . \tag{2.121}$$

The **prediction equations** of the Kalman filter have following form

$$\boldsymbol{\alpha}_{t|t-1} = K\boldsymbol{\alpha}_{t-1|t-1} \tag{2.122}$$

$$x_{t|t-1} = \boldsymbol{b}^T \boldsymbol{\alpha}_{t|t-1} \tag{2.123}$$

$$P_{t|t-1} = KP_{t-1|t-1}K^T + \boldsymbol{d}\sigma_{w_t}^2 \boldsymbol{d}^T \quad . \tag{2.124}$$

With the help of

$$\nu_t = x_t - x_{t|t-1} \quad , \tag{2.125}$$

the variance matrix can be defined as

$$\Sigma_t = E(\nu_t \nu_t^T) = E\left((x_t - x_{t|t-1})(x_t - x_{t|t-1})^T\right) = \boldsymbol{d}^T P_{t|t-1} \boldsymbol{d} \quad . \tag{2.126}$$

The **updating equations** are defined as follows

$$\boldsymbol{\alpha}_{t|t} = \boldsymbol{\alpha}_{t|t-1} + P_{t|t-1}\boldsymbol{b}\Sigma_t^{-1}\nu_t \tag{2.127}$$

$$P_{t|t} = P_{t|t-1} - P_{t|t-1}\boldsymbol{b}\Sigma_t^{-1}\boldsymbol{b}^T P_{t|t-1} \quad . \tag{2.128}$$

The updating equations can be interpreted in the following way: any new information enters the system of equations through the Kalman gain $P_{t|t-1}\boldsymbol{b}\Sigma_t^{-1}\nu_t$. Intuitively, the gain can be seen as a measure of how much to trust in the current measurement by taking all past observations into account.

2.3.5.7. Time series analysis forecast error metrics

Error metrics are used to judge the quality of forecasts of time series data. The metrics presented here can be found in Hyndman and Athanasopoulos 2014. These metrics are used on the residuals r_t of a time series forecast. Note that it is important that the forecast is applied not on the same data on which the time series mode has been fitted. Now follows a list of forecast metrics used to judge the goodness of a time series model. The first metric is the Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{N}\sum_{t=1}^{N}|r_t| \quad . \tag{2.129}$$

N is the number of residual observations r_t. Another often used metric is the Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N} r_t^2} \tag{2.130}$$

To compare the forecast performance between different data sets, the scale independent Mean Absolute Percentage Error (MAPE) can be used

$$\text{MAPE} = 100 \cdot \frac{1}{N}\sum_{t=1}^{N}\left|\frac{r_t}{x_t}\right| \quad . \tag{2.131}$$

For comparing the performance of a certain model with a reference model, the Forecast Skill Score (SS) can be used

$$\text{SS} = 1 - \frac{\text{MSE}_F}{\text{MSE}_R} \quad , \tag{2.132}$$

with the Mean Squared Error (MSE) defined through

$$\text{MSE} = \frac{1}{N}\sum_{t=1}^{N} r_t^2 \tag{2.133}$$

The subscript F stands for the forecast time series and R stands for the reference time series. SS is 1 if the forecast is perfect, between 0 and 1 if the forecast is better than the reference, 0 if the forecast and the reference are equally good and negative if the reference performs better.

2.3.6. Stochastic event detection

A stochastic event detection algorithm has to be able (i) to determine whether there is a statistical evidence for claiming that an event has occurred in the system and (ii) to automatically detect the most likely time when this event started. Since the measurement signal is overshadowed by uncertainties and stochastic fluctuations, probability theory is necessary to extract signal from noise. All methods presented in this section can be found, for example, in (Basseville and Nikiforov 1993).

2.3.6.1. CUSUM method

The CUmulative SUM control chart (CUSUM) method is a technique to identify changes in data. It is based on computations of the cumulative sum of positive g_t^+ and negative g_t^- differences (Basseville and Nikiforov 1993). An event is detected if these changes surpass a specific threshold τ. If this happens, an alarm is triggered and g_t^+ and g_t^- is set to zero. The algorithm can be applied on models containing a drift with an additional drift parameter d. Additionally, this drift parameter can be used to clear the memory of the algorithm on small changes going only in one direction in the past. Mathematically, the CUSUM method is formulated in following way

$$C_t = x_t - x_{t-1} \quad . \tag{2.134}$$

The sum of positive changes is defined as

$$g_t^+ = \max\left\{C_t + g_{t-1}^+ - d, 0\right\} \quad . \tag{2.135}$$

Similarly, the sum of negative changes is defined as

$$g_t^- = \max\left\{g_{t-1}^- - C_t - d, 0\right\} \quad . \tag{2.136}$$

The event detection time t_D—the time when an anomaly is dected—is the time when the positive and/or negative signal escapes the threshold defined by

$$t_D = \min\left(t | g_t^\pm \geq \tau\right) \quad . \tag{2.137}$$

The CUSUM algorithm depends heavily on the τ value. The higher τ—the stronger the signal changes have to be to become detected timely. In contrary, small τ will lead to a lot of false alarms. The ideal τ can be found by analyzing historical measurement data.

Later in this thesis, the CUSUM technique will be used for leakage detection. Equation 2.135 will be used for flow measurements due to the inflow increase caused by the leak—Equation 2.136 will be used for detecting pressure drops. But also both criteria can be used simultaneously if one wants to detect a signal escaping a certain band, which occurs during sensor failures, for example.

2.3.6.2. Likelihood-ratio test

This is a log-likelihood approach for detecting an event and can be found in Basseville and Nikiforov 1993 and Granjon 2013, for example. The event is supposed to change the parameters ξ_t of the signal's underlying stochastic process. For example, a realization of the stochastic process can be the data points of a time series x_t (Shumway and Stoffer 2006).

The task for event detection is (i) to identify if the parameters ξ_t of the probability distributions changes at all, and if that is true, (ii) to identify the point in time τ when they change. Subsequently, additional information can be retrieved of the magnitude of this change.

The null hypothesis $\mathcal{H}_0$ is that the parameters do not change, $\mathcal{H}_1$ that ξ_t changes at a certain point in time τ

$$\mathcal{H}_0 : \quad \xi_1 = \xi_2 = \ldots = \xi_{N-1} = \xi_N \tag{2.138}$$

$$\mathcal{H}_1 : \quad \xi_1 = \xi_2 = \ldots \xi_\tau \neq \xi_{\tau+1} = \ldots = \xi_{N-1} = \xi_N \quad . \tag{2.139}$$

The detection is based on the log-likelihood ratio. The likelihoods $\mathcal{L}$—a measure of the goodness of a hypothesis—is defined through (Granjon 2013)

$$\mathcal{L}(\mathcal{H}_0) \;=\; p(\boldsymbol{x}|\mathcal{H}_0) = \prod_{i=1}^{N} p(x_i|\xi) \tag{2.140}$$

$$\mathcal{L}(\mathcal{H}_1) \;=\; p(\boldsymbol{x}|\mathcal{H}_1) = \prod_{i=1}^{\tau} p(x_i|\xi_1) \cdot \prod_{j=\tau+1}^{N} p(x_i|\xi_2) \quad . \tag{2.141}$$

Subsequently, the log-likelihood ratio is (Basseville and Nikiforov 1993)

$$\mathcal{R}_\tau = \ln\left(\frac{\mathcal{L}(\mathcal{H}_1)}{\mathcal{L}(\mathcal{H}_0)}\right) = \sum_{i=1}^{\tau} \ln p(x_i|\xi_1) + \sum_{j=\tau+1}^{N} \ln p(x_i|\xi_2) - \sum_{i=1}^{N} \ln p(x_i|\xi) \quad . \tag{2.142}$$

This is a function of τ ($\mathcal{R}_\tau = f(\tau)$).

The generalized likelihood $\mathcal{G}$ is defined as (Basseville and Nikiforov 1993)

$$\mathcal{G} = \max_{i\leq\tau\leq N} \mathcal{R}_\tau \quad . \tag{2.143}$$

The time instant $\hat{\tau}$ when a change is detected, is then defined through the optimization

$$\hat{\tau} = \arg\max_{i\leq\tau\leq N} \mathcal{R}_\tau \quad . \tag{2.144}$$

$\mathcal{H}_0$ is rejected if $\mathcal{G}$ is larger than a threshold value λ^*

$$2\mathcal{G} = 2\mathcal{R}_{\hat{\tau}} > \lambda^* \quad . \tag{2.145}$$

λ^* can be chosen, for example, as AIC or BIC values (see Section 2.3.4).

For leakage detection, we will suppose that the measurement values are normal distributed[17] and that the mean of the distribution changes due to a leak—either the mean increases in the flow measurements or the mean decreases in the pressures

$$x_t \;=\; \begin{cases} \mathcal{N}(\mu_1, \sigma) & \text{for} \quad t \leq \tau \\ \mathcal{N}(\mu_2, \sigma) & \text{for} \quad t > \tau \quad . \end{cases} \tag{2.146}$$

The null hypothesis is that the mean does not change over time at all ($x_t \in \mathcal{N}(\mu, \sigma)\ \forall t$). The probability density function of the normal distribution $\mathcal{N}$ is defined over its mean μ and its standard deviation σ in following way

$$\mathcal{N}(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad . \tag{2.147}$$

[17] In fact, the approach is flexible enough that any distribution can be used.

Thus, the null hypothesis results in

$$\mathcal{L}(\mathcal{H}_0) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^N \prod_{i=1}^{N} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad . \tag{2.148}$$

Consequently, the alternative hypothesis is

$$\mathcal{L}(\mathcal{H}_1) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^N \prod_{i=1}^{\tau} e^{-\frac{(x-\mu_1)^2}{2\sigma^2}} \prod_{i=\tau+1}^{N} e^{-\frac{(x-\mu_2)^2}{2\sigma^2}} \tag{2.149}$$

The log-likelihood ratio results then in

$$\mathcal{R}_\tau = \left(\frac{\mathcal{L}(\mathcal{H}_1)}{\mathcal{L}(\mathcal{H}_0)}\right) = \frac{1}{2\sigma^2}\left(\sum_{i=1}^{\tau}(x_i-\mu_1)^2 + \sum_{j=\tau+1}^{N}(x_i-\mu_2)^2 - \sum_{i=1}^{N}(x_i-\mu)^2\right) \quad . \tag{2.150}$$

If this value is greater than λ^* then a leak occurred. The start time of the leak $\hat{\tau}$ can be found by applying equation (2.144).

2.3.6.3. Bayesian step detector

The problem defined in equation (2.146) can be formulated in terms of Bayes' theorem

$$P(A|B) \ = \ \frac{P(B|A)P(A)}{P(B)} \quad . \tag{2.151}$$

The short discussion of the derivation presented here in this section is based on Ruanaidh and Fitzgerald 1996.

The problem rewritten in the Bayes' theorem context results in

$$P(\mu_1\mu_2\sigma\tau|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\mu_1\mu_2\sigma\tau)P(\mu_1\mu_2\sigma\tau)}{P(\boldsymbol{x})} \quad , \tag{2.152}$$

where $P(\mu_1\mu_2\sigma\tau|\boldsymbol{x})$ is the posterior distribution, $P(\boldsymbol{x}|\mu_1\mu_2\sigma\tau)$ is the likelihood and $P(\mu_1\mu_2\sigma\tau)$ is called the prior distribution. $P(\boldsymbol{x})$ is the evidence—since it is analytically difficult to calculate and since it is a constant factor, its calculation can be skipped. Then the posterior distribution is proportional to the likelihood and the prior

$$P(\mu_1\mu_2\sigma\tau|\boldsymbol{x}) \propto P(\boldsymbol{x}|\mu_1\mu_2\sigma\tau)P(\mu_1\mu_2\sigma\tau) \quad . \tag{2.153}$$

Constant uninformative priors c_i are assumed for the parameters: $P(\mu_1) = c_1$, $P(\mu_2) = c_2$, $P(\tau) = c_3$ and $P(\sigma) = c_4 = \frac{1}{\sigma}$. $P(\tau|\boldsymbol{x})$ is sought and it can be calculated through marginalization

$$P(\tau|\boldsymbol{x}) \ \propto \ \int_0^\infty d\sigma \int_{-\infty}^{\infty} d\mu_1 \int_{-\infty}^{\infty} d\mu_1 \frac{P(\boldsymbol{x}|\mu_1\mu_2\sigma\tau)}{\sigma} \tag{2.154}$$

The likelihood function is given through

$$P(\boldsymbol{x}|\mu_1\mu_2\sigma\tau) = \prod_{i=1}^{\tau} P(x_i|\mu_1\sigma) \prod_{j=\tau+1}^{N} P(x_i|\mu_2\sigma) = \dots \tag{2.155}$$
$$\dots = \left(2\pi\sigma^2\right)^{-N/2} \exp\left[-\frac{1}{2\sigma^2}\left(\sum_{i=1}^{\tau}(x_i^2+\mu_1^2-2\mu_1 x_i) + \sum_{j=\tau+1}^{N}(x_i^2+\mu_2^2-2\mu_2 x_i)\right)\right] \quad .$$

Following two identities can be used to solve the integrals for marginalization through integration

$$\int_{-\infty}^{\infty} \exp\left(-ax^2-bx+c\right)dx = \sqrt{\frac{\pi}{a}}\exp\left(\frac{b^2}{4a}-c\right) \tag{2.156}$$

and

$$\int_{0}^{\infty} x^{\alpha-1}\exp(-Qx)dx = \frac{\Gamma(\alpha)}{Q^{\alpha}} \quad . \tag{2.157}$$

After integration the distribution of τ—depending on the measurements only—results in

$$P(\tau|\boldsymbol{x}) \propto \frac{1}{\sqrt{\tau(N-\tau)}}\left(\sum_{k=1}^{N} x_k^2 - \frac{(\sum_{i=1}^{\tau} x_i)^2}{\tau} - \frac{(\sum_{j=\tau+1}^{N} x_j)^2}{N-\tau}\right)^{-\frac{N-2}{2}} \quad . \tag{2.158}$$

Together with the log-likelihood defined through the logarithm of the likelihood function in equation (2.155), the single marginal distributions $P(\tau|\boldsymbol{x})$, $P(\mu_1|\boldsymbol{x})$, $P(\mu_2|\boldsymbol{x})$ and $P(\sigma|\boldsymbol{x})$ can be also retrieved numerically by MCMC (see Section 2.3.2). This will be examined in more detail for leakage detection in Section 3.3.5.5.

2.4. Case studies and field tests

In this thesis the developed methods and algorithms will be tested on three networks—a simple network taken from literature and two real-world networks from Austria. These three networks, the purpose why they have been chosen, for what task they will be used and field tests (if any) will be presented in this section.

2.4.1. Poulakis

The first network is an artificial network taken from literature. It was first introduced by Poulakis et al. 2003 who used it for testing a developed Bayesian probabilistic system identification framework for leakage detection and localization. According to Poulakis et al. 2003, this network represents a simplified municipal WDS that is typical for an industrial unit. It is small in terms of the numbers of nodes, big in terms of total pipe length and very densely meshed.

Hydraulic model Figure 2.16 shows the hydraulic model of the Poulakis network simulated with OOPNET (Steffelbauer and Fuchs-Hanusch 2015). Nodes are shown as circles with colors depending on their simulated pressure value in bar. Their names—respectively ids—start in this network with a *J*. The network's pipes are represented as lines between these nodes with colors corresponding to their actual flow value in L/s. Pipe names start in this network with P. The reservoir in the upper left corner is represented in a diamond shape with id J-01.

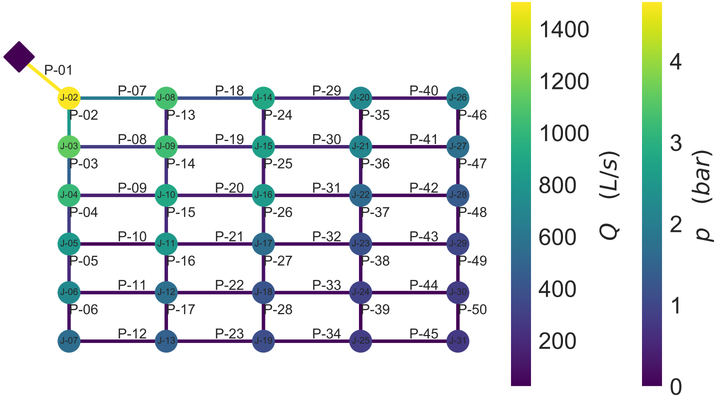


Figure 2.16.: Network of Poulakis (Poulakis et al. 2003) with simulated pressures p and flows Q

The network consists of 31 nodes—30 junctions plus one reservoir—connected through 50 pipes. This network is highly meshed possessing 20 loops in total. The pipes have different diameters ranging from 300 to 600 mm. The pipe length is 1000 m for vertical and 2000 m for horizontal pipes, resulting in total length of the network of 71 km. All pipes possess the same roughness coefficient of $C_i = 0.26$ mm. The junctions have zero elevation and each junction has a water demand of $q_D = 50$ L/s. The junctions are supplied by a tank (J-01) with an elevation of $e = 52$ m. The total water demand respectively the inflow in the system equals $Q = 1500$ L/s.

Purpose The model is a theoretical network taken from literature and serves as a concise example. It will be used for vividly illustrating leakage localization algorithms and fitness landscape analysis in simulations (see Section 5.3.2).

2.4.1.1. Transformation of pressure units

The hydraulic simulation software EPANET (Rossman 2000) returns heads and pressures in meter water column (mH$_2$0)[18]. Usually, pressure sensors provide measurements in (derived) SI units—Pascal (Pa) respectively N/m^2 or bar (1 mbar=1 hPa = 100 Pa).

The conversion factor c_F between pressure in meter (water column) and bar is calculated in the following way:

$$p \quad [\mathrm{m}] = \underbrace{\frac{10^5}{\rho_{H_2O}(T)\cdot g}}_{c_F} \, p \quad [\mathrm{bar}] \quad , \tag{2.159}$$

where ρ_{H_2O} is the density of water (note that this quantity is temperature dependent) and g is the gravitational acceleration on Earth ($g = 9.80665$ [m/s^2]).

[18] ... or pounds per square inch (psi) in US Customary units.

The temperature dependence of ρ_{H_2O} is approximated through following formula[19]

$$\rho_{H_2O}(T) = 999.972 - \left(7.0 \cdot 10^{-3}\right) \cdot (T - 4.0)^2 \quad . \tag{2.160}$$

T is given in degree Celsius. If not stated otherwise, T is assumed to be $T = 4$ [°C] resulting in a conversion factor of $\rho_{H_2O}(4) = 999.972$.

2.4.2. Linz-Pichling

The second network is a real-world WDS. It is located in a small rural part around the city of Linz—Austria's third biggest city. This network has been chosen as a case study network during a project because of its District-Metered Area (DMA) structure as it has a single inlet and a single outlet point.

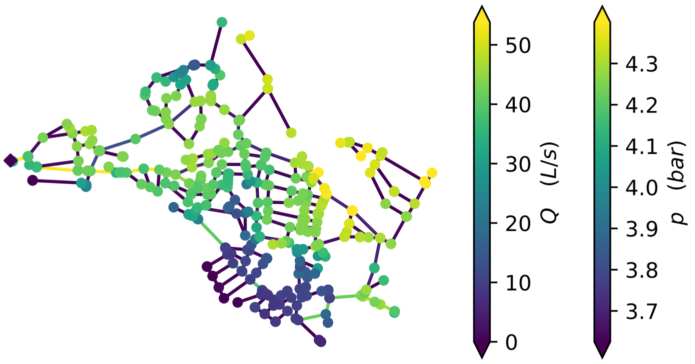


Figure 2.17.: Case study Linz network with simulated pressure p and flows Q

Hydraulic model The hydraulic network is gravitationally fed by a reservoir from the west side of the system (diamond shape). It consists in total of this reservoir, 392 junction nodes and 452 pipes, with a total length of approximately 37 km. The pipe diameters vary from 70 to 400 mm. The nodal base demands were allocated from billing information of the particular customers. The area is mainly a residential zone with a few small industries. A big industrial customer is in the north of the measurement zone. The roughness values of the pipes result from a calibration campaign undertaken by a company prior to the project. Calibration scenarios have been generated by opening hydrants during the night to produce high velocities in the network.

Purpose First, field tests for leakage localization were planned in this network, but later on during the project, they have been canceled. Nevertheless, sensor placement algorithms and leakage localization were widely tested on this network. For the purpose of leak localization and hence for the application of the introduced sensor placement approach, the nodal minimum night consumption has been of interest. Nodal consumption was derived for each node from the nodal demand taken from yearly billing information and a pattern generated from inflow measurements of the DMA.

[19] This formula is taken from Formula 2.3 in `http://www.iup.uni-heidelberg.de/institut/studium/lehre/AquaPhys/docMVEnv3_11/MVEnv3_2011_p1_2_Dichte.pdf`.

2.4.3. Graz-Ragnitz

The third studied system is a real-world network in an rural area located in the surroundings of the city of Graz (Austria). The network has been chosen due to its DMA structure and its proximity to our institute. Long-time real-world field studies have been performed in this study area during 2015 and 2016.

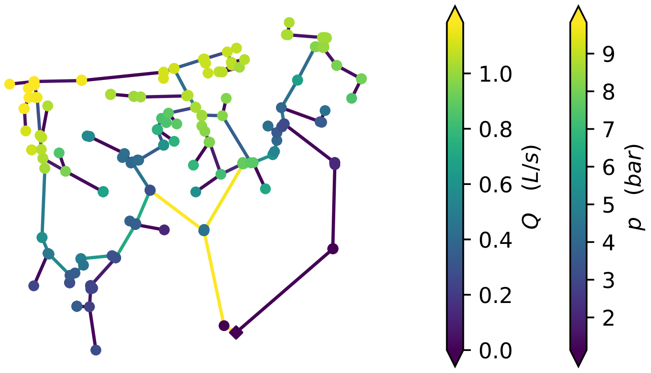


Figure 2.18.: Case study Ragnitz-Simple network with simulated pressures p and flows Q

Hydraulic model Three different hydraulic models exist for this WDS in different resolutions: (i) the **Ragnitz-Simple** model, (ii) the **Ragnitz-Complicated** model and the (iii) **Ragnitz-Detailed** model. All three models are supplied by a single tank in the southern part of the system (e.g. diamond in Figure 2.18) since they represent the same network. From this tank, water enters the system over two pipes. The inflow in the system has been measured on the left pipe—the right pipe, where no inflow was measured, has been closed during the whole field testing campaign. The demand data was retrieved from billing information from the years 2012 to 2015 and a nodal demand equal to the average consumption of the customers during this time-span has been allocated to the nodes in the system. In total, 35 hydrants are situated in the WDS where outflows are generated for calibration and leakage localization purposes as well as pressures were measured. The system consists mainly of PVC and PE pipes, but also some steel and cast iron parts can be found in the system. Noticeable is the high pressure of over 9 bar in some parts of the system resulting from huge elevation differences between the tank and the northern area of the network—North elevation is 404.5 m, South elevation is 505.2 m. No pressure reducing valves are installed in the measurement zone.

(i) **Ragnitz-Simple** is the most elementary model consisting of 216 pipes and 212 junctions. Diameters are ranging between 25.4 mm (1 inch) and up to 125 mm. The total pipe length of the simple system is 9.6 km. The graphical representation of the model with simulated pressure and flow values can be found in Figure 2.18.

(ii) **Ragnitz-Complicated** is a less skeletonized model of the network than Ragnitz-Simple. It consists of 650 nodes 658 pipes with a total pipe length of 10.2 km. The pipes have the same diameter range as the simple model. A graphical representation of the model with simulated pressure and flow values can be found in Figure 2.19.

(iii) **Ragnitz-Detailed** is the model containing the most details. It has been generated from a GIS export during the Aquademia project by a student worker. It consists of 1300 nodes and 1309 pipes and has a total pipe length of 16.2 km since all household connections are contained in this model. A hydraulic simulation of this model is depicted in Figure 2.20.

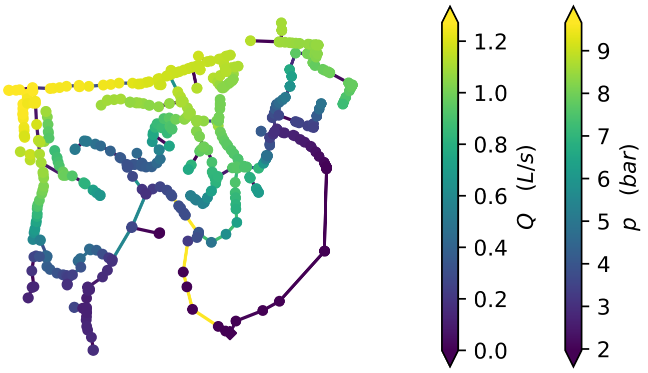


Figure 2.19.: Case study Ragnitz-Complicated network with simulated pressures p and flows Q

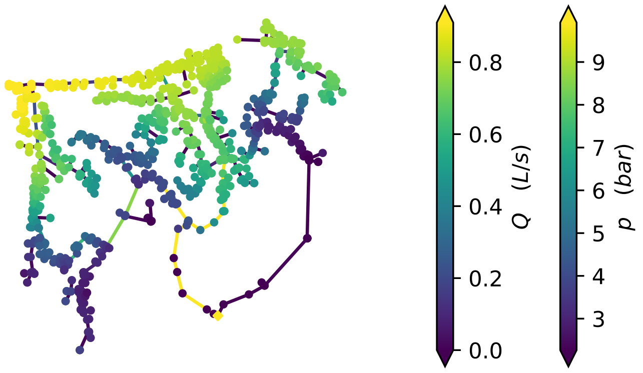


Figure 2.20.: Case study Ragnitz-Detailed network with simulated pressures p and flows Q

Purpose The purpose of these networks is to test algorithms for calibration, optimal sensor placement, leakage detection and localization in simulations and with real-world data.

2.4.3.1. Measurements

For real-world case studies, (i) flow, (ii) pressure and the (iii) tank level were measured in the system.

(i) Flow Flow was measured at the tank with two measurement devices at the DMA inlet of the system: (a) a Woltmann water meter and an (b) Ultrasonic Flow Meter (UFM).

(a) The data from the Woltmann water meter has been provided by the WU from their SCADA system. This measurement device has been used to check the measurements of the UFM.

(b) The UFM at the inflow point is a clamp-on device of type FLUXUS ADM 6725 by FLEXIM. Data was manually read out from the internal data storage of this device, usually once a week (depending on the time resolution). The accuracy according to the data sheet for flow measurements is 1 % to 3 % of the measurement value ±0.01 m/s.

(ii) Pressure At peak times, up to twelve pressure loggers where installed at selected hydrants in the system. The measurement positions and the hydrants are depicted in Figure 2.21. The hydrants were selected according to optimal sensor placement algorithms described in Chapter 4. All high precision pressure sensors were of the same type—SEWAD 30 with a measurement range of 0 - 30 bar and a resolution of up to 10 mbar (or ≈ 0.01 [mH_2O]) with an accuracy of ±0.2% of the current measurement value. Data was recorded and stored at the measurement device and manually collected approximately once every ten days (also depending on the time resolution).

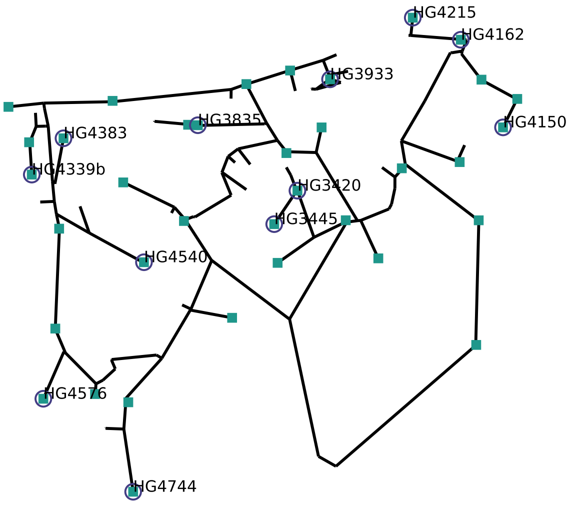


Figure 2.21.: Ragnitz-Simple network with hydrants (green squares) and pressure sensor locations (blue circles)

(iii) Tank level The tank level data was retrieved from the WU's SCADA system and serves as (a) a boundary condition for the hydraulic model in the automatic calibration process and (b) for model-based leak detection and localization the tank levels are subtracted from the pressure data to get rid of the water level influences on the measurements.

All the measurement data has been collected and stored in the institutes Time Series Database (TSDB)—an InfluxDB data base part of the Open-source Scientific Data Management System (OpenSDM) (Camhy D. et al. 2014; Camhy et al. 2012).

2.4.3.2. Hydraulic model calibration

Calibration scenarios The calibration experiments were conducted in the night from 11^{th} to the 12^{th} of April 2016. The MNF has been chosen in order that the influence of customer demand is minimal on the calibration measurements. For successful roughness parameter assessment, high velocities have to be produced in the system (Walski 2000) to guarantee that the effect of roughness on pressure becomes larger than the measurement uncertainties. For that reason, hydrants were opened to produce large water flows in the system. Pressure was measured at during the calibration night 12 locations. The measurement locations are depicted in Figure 2.21. The inflow has been measured at the tank. Additionally, the outflow and pressure at the opened hydrants has been measured with additional devices—Hydatlog 80mm with Storz B hose coupling—magnetic inductive flow meters capable of measuring flows between $Q_{min} = 1.5$ [L/s] and $Q_{max} = 60.3$ [L/s] with an accuracy of 0.5%. The most time consuming part of the night measurements has been relocating and connecting the hydrant measurement devices. Hence, a scheduled hydrant opening scheme was developed where hydrants were opened in parallel to produce supplementary calibration scenarios without the necessity of relocating the devices. This promised the most effective use of the short time frame during the MNF.

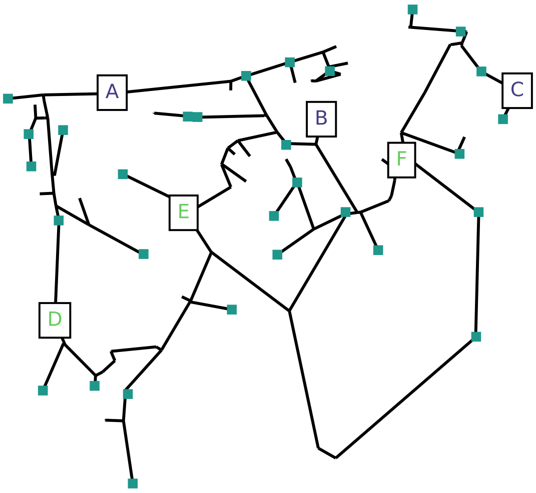


Figure 2.22.: Ragnitz-Simple network with hydrant opening positions for generating calibrations scenarios named from A to F

The locations of the hydrant openings can be found in Figure 2.22. The measurement campaign was originally planned with three measurement devices. The scenarios have been generated in two stages: In the first round, hydrants were opened at positions A, B and C. The generated outflows and measurement times are listed in Table 2.2. The measurements were taken between t_{start} and t_{end}. t_{start} is the time when the hydrants are already fully opened and the pressures in the system steadied down. t_{end} is the time-step before the hydrants were closed again. Through parallel opening, seven scenarios—named from S_1 to S_7—were generated within the first round. S_1, S_4 and S_6 were generated by opening a single hydrant. In scenarios S_2, S_3, S_5 and S_7 hydrants were opened in parallel. Subsequently, the measurement devices were moved to positions D, E and F. Then the second measurement round was performed by opening these hydrants. The corresponding outflows with start- and end-times are listed in Table 2.3.

Table 2.2.: Calibration scenarios S_i for the first round with measured outflows for hydrant openings at positions A, B and C (see Figure 2.22) from t_{start} to t_{end}.

	A	B*	C	t_{start}	t_{end}
Scenario	[L/s]	[L/s]	[L/s]	[H:M:S]	[H:M:S]
S_1	15.03	—	—	01:37:45	01:40:30
S_2	8.06	4.11	—	01:47:30	01:49:45
S_3	4.97	1.14	5.10	01:58:00	01:59:30
S_4	—	11.54	—	02:04:00	02:06:00
S_5	—	2.62	7.05	02:09:15	02:11:45
S_6	—	—	12.03	02:15:45	02:18:15
S_7	6.93	—	7.06	02:22:15	02:25:15

Table 2.3.: Calibration scenarios S_i for the second round with measured outflows for hydrant openings at positions D, E and F (see Figure 2.22) from t_{start} to t_{end}.

	D*	E	F	t_{start}	t_{end}
Scenario	[L/s]	[L/s]	[L/s]	[H:M:S]	[H:M:S]
S_8	7.60	—	—	03:23:45	03:26:30
S_9	1.81	6.06	—	03:31:30	03:33:30
S_{10}	1.27	5.08	5.05	03:36:45	03:39:30
S_{11}	—	15.97	—	03:44:00	03:45:45
S_{12}	—	6.00	6.00	03:51:45	03:54:45
S_{13}	2.20	—	6.02	03:58:30	04:01:00
S_{14}	—	—	9.07	04:04:30	04:07:45

Unfortunately, only two instead of three hydrant measurement devices were available in the calibration night. Therefore, outflows generated at locations B* and D* were not measured directly at the hydrants. That is why they are marked with a star in Tables 2.2 and 2.3. The corresponding outflows have been estimated by subtracting the MNF from the corresponding inflow measurements at the times of the hydrant openings. The MNF for the measurement night has been calculated by averaging the inflow respectively the according pressures between 1:22 and 1:31 in the morning. The night flow results in $Q_{MNF} = 1.217$ [L/s]. Differences in pressure and flow of the calibration scenarios and the MNF values are depicted in Tables 2.4 and 2.5. The system's inflow $Q(t)$ and the measured pressures $p(t)$ during the scenarios can be seen in Figure 2.23 and 2.24. Additionally, the times when the measurements are used for calibration and for leak localization are highlighted for each scenario in the time series.

Automatic calibration Model-based leak localization depends on measured flow and pressure differences when a leak occurs. The pressure differences are generated by higher friction losses resulting from the increased flow respectively velocities in the network. The roughness parameters of the pipes are responsible for these head losses. Consequently, these roughnesses have to be estimated as accurate as possible. For that reason, a computer program has been developed to calculate pipe roughness

Table 2.4.: This table shows the differences to the MNF and calibration scenario measurements for the inflow (ΔQ) and the first six pressure measurements (Δp). See Figure 2.21 for the measurement device locations.

	Inflow	HG3420	HG3445	HG3835	HG3933	HG4150	HG4162
	ΔQ [L/s]	Δp [bar]	Δp [bar]	Δp [bar]	Δp [bar]	Δp [bar]	Δp [bar]
S_1	14.66	-0.92	-0.92	-1.53	-1.95	-0.92	-0.92
S_2	12.12	-0.68	-0.68	-1.01	-1.14	-0.68	-0.68
S_3	11.16	-0.64	-0.64	-0.72	-0.77	-1.14	-0.88
S_4	11.49	-0.65	-0.65	-0.85	-0.84	-0.65	-0.65
S_5	9.63	-0.52	-0.52	-0.49	-0.49	-1.41	-0.95
S_6	11.75	-0.76	-0.76	-0.67	-0.66	-3.10	-1.91
S_7	13.69	-0.91	-0.91	-1.02	-1.12	-1.80	-1.35
S_8	7.69	-0.30	-0.30	-0.45	-0.54	-0.30	-0.30
S_9	7.83	-0.30	-0.30	-0.41	-0.41	-0.31	-0.31
S_{10}	11.36	-0.64	-0.64	-0.68	-0.69	-0.68	-0.68
S_{11}	15.69	-1.03	-1.03	-1.31	-1.30	-1.03	-1.03
S_{12}	11.62	-0.67	-0.67	-0.70	-0.70	-0.73	-0.73
S_{13}	8.17	-0.38	-0.38	-0.39	-0.40	-0.44	-0.44
S_{14}	9.34	-0.50	-0.50	-0.45	-0.45	-0.62	-0.63

Table 2.5.: This table shows the differences to the MNF and calibration scenario measurements for the inflow (ΔQ) and the last six pressure measurements (Δp). See Figure 2.21 for the measurement device locations.

	Inflow	HG4215	HG4339b	HG4383	HG4540	HG4576	HG4744
	ΔQ [L/s]	Δp [bar]	Δp [bar]	Δp [bar]	Δp [bar]	Δp [bar]	Δp [bar]
S_1	14.66	-0.92	-2.97	-2.90	-2.84	-2.61	-1.16
S_2	12.12	-0.68	-1.43	-1.40	-1.38	-1.29	-0.77
S_3	11.16	-0.88	-0.89	-0.88	-0.87	-0.83	-0.60
S_4	11.49	-0.65	-0.78	-0.78	-0.77	-0.75	-0.64
S_5	9.63	-0.95	-0.47	-0.47	-0.47	-0.47	-0.43
S_6	11.75	-1.91	-0.65	-0.65	-0.65	-0.64	-0.61
S_7	13.69	-1.34	-1.35	-1.33	-1.31	-1.24	-0.87
S_8	7.69	-0.30	-0.96	-1.00	-1.02	-1.10	-0.40
S_9	7.83	-0.31	-0.44	-0.45	-0.45	-0.48	-0.36
S_{10}	11.36	-0.68	-0.71	-0.71	-0.71	-0.75	-0.63
S_{11}	15.69	-1.03	-1.28	-1.28	-1.28	-1.31	-1.22
S_{12}	11.62	-0.73	-0.68	-0.68	-0.68	-0.71	-0.64
S_{13}	8.17	-0.44	-0.43	-0.43	-0.44	-0.48	-0.34
S_{14}	9.34	-0.62	-0.45	-0.45	-0.45	-0.49	-0.41

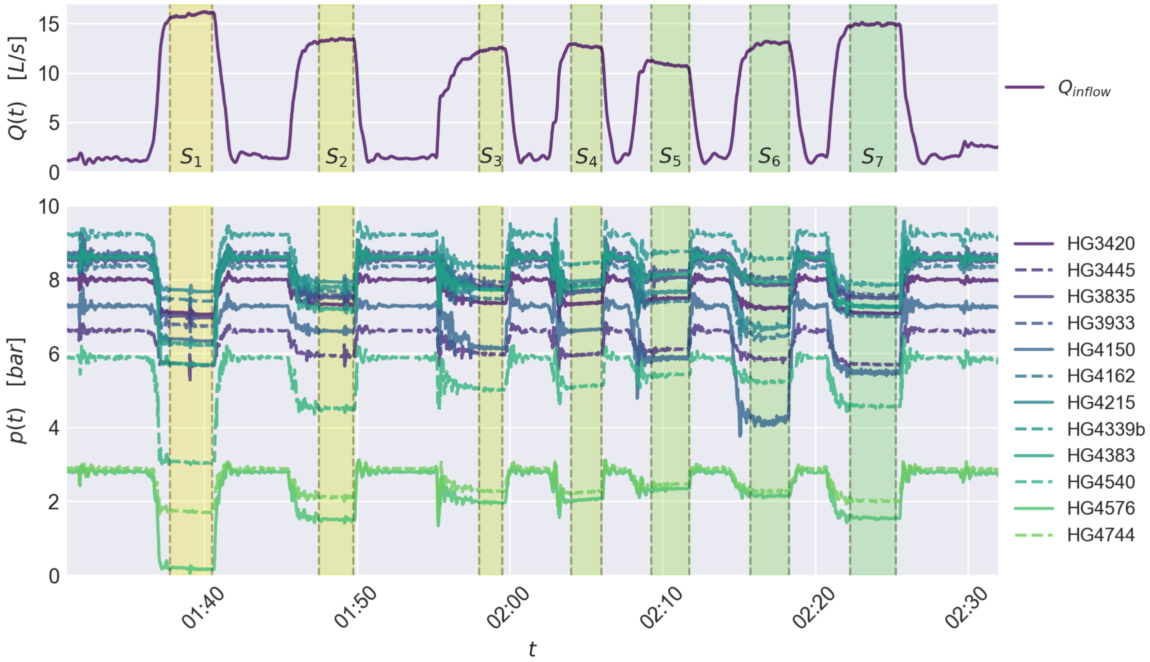


Figure 2.23.: Flow $Q(t)$ (top) and pressure $p(t)$ (bottom) measurements during the first round of the calibration. The scenarios S_1 to S_7 are marked and the time-spans where the measurements are used for automatic calibration are highlighted

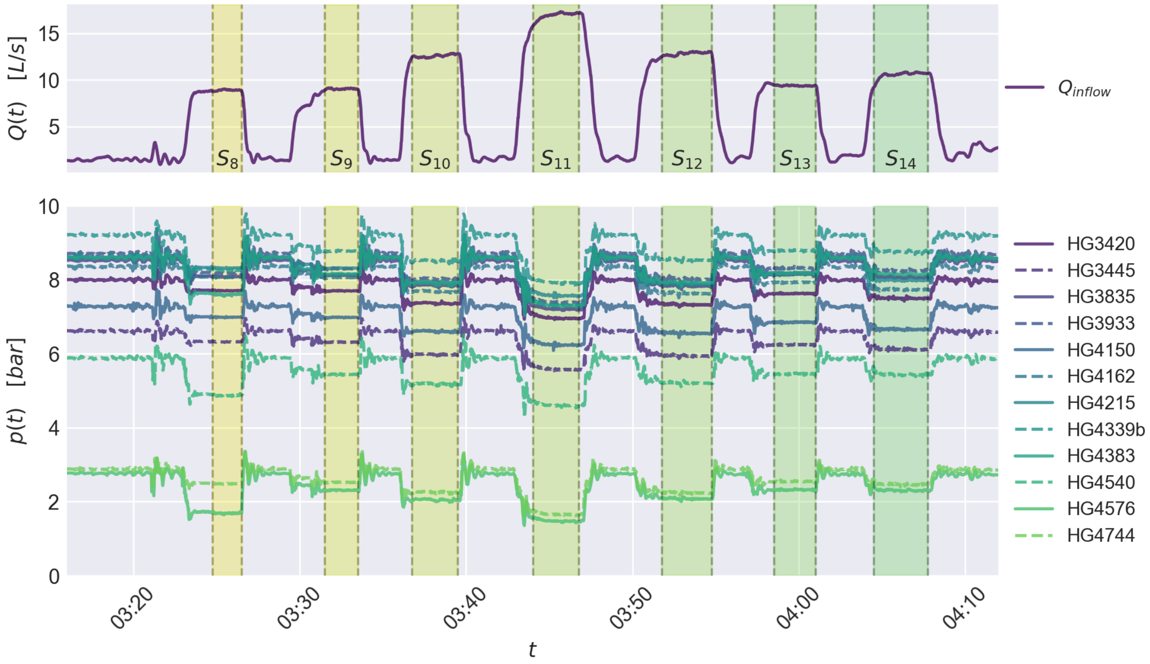


Figure 2.24.: Flow $Q(t)$ (top) and pressure $p(t)$ (bottom) measurements during the second round of the calibration. The scenarios S_8 to S_{14} are marked and the time-spans where the measurements are used for automatic calibration are highlighted

values from the calibration data.

An implicit calibration approach is chosen (Savic et al. 2009). Roughness parameter estimation is formulated as an inverse problem and the roughness parameters are found that fit the measured values the best with a heuristic optimization algorithm from Section 2.1. This approach is similar to the model-based leak localization approach in Chapter 5. The fitness function of the calibration problem is formulated as a sum of least squares

$$f(\mathbf{x}) = \sum_{j=1}^{N_S} \sum_{i=1}^{N_m} (m_{i,j} - \hat{m}_{i,j}(\mathbf{x}))^2 \quad \rightarrow \quad \min_{\mathbf{x}} f(\mathbf{x}) \qquad . \tag{2.161}$$

$m_{i,j}$ is the i-th measurement value in scenario j, $\hat{m}_{i,j}(\mathbf{x})$ is its corresponding value resulting from a hydraulic simulation, N_m is the number of measurements and N_S is the number of calibration scenarios. $\mathbf{x}$ is the parameter vector containing the roughness values and depends on the parametrization of the problem. In general, more roughness values have to be found than measurements exist in the system. For that reason, finding all roughness values accurately is a highly under-determined problem. Consequently, the roughness values have to be subsumed into groups. Criteria for selecting pipe groupings are pipe material, pipe age, diameter, relative locations in the system or the identification of critical pipes influencing pressure heads (Mallick et al. 2002). The developed algorithm is capable of using all these criteria and furthermore also allows combining different groups to increase the number of possible groups. Lippacher 2018 found that one of the best groupings for the Ragnitz network for roughness calibration aiming model-based leak localization was to group the pipes into five groups according to their relative position in the system. The locations of these five groups are depicted in Figure 2.25 with different colors for the groups from G_1 to G_5. Additionally, while analyzing the pressures in the calibration scenarios, a pipe belonging to G_5 with high head loss has been identified in the system prior to calibration. This pipe was expected to either be heavily encrusted[20] or a valve has been partially closed. The affected pipe is highlighted in Figure 2.26 with an arrow. Later on, after the field testing campaign, the WU confirmed that a valve has been forgotten to be fully re-opened after maintenance work in the system. Since this partially-closed valve affected the results of the roughness calibration, it was modeled with a minor loss coefficient k_i (see equation (2.61)) which was also incorporated in the parameter vector $\mathbf{x}$ in the optimization problem defined in equation (2.161).

The optimization problem in equation (2.161) was solved with the DE algorithm as described in Section 2.1.2.2.2. Not all generated scenarios were used due to the missing measurement device at location B^* respectively D^*, because the hydrant outflow could not be estimated accurately. Hence, only scenarios without hydrant openings at these locations are chosen. This scenarios are S_1, S_6, S_7, S_{11}, S_{12} and S_{14} (see Table 2.2 and 2.3). Additionally, a MNF scenario S_0 was taken into account using measurement values between 1:21 and 1:31. The resulting roughness values c_i for the groups are taken from Lippacher 2018 and can be found in Table 2.6. A minimal fitness value of $f(\mathbf{x}) = 20.28$ was found. The average absolute deviation D_i

$$D_i = \frac{1}{N_S} \sum_{j=1}^{N_S} |m_{i,j} - \hat{m}_{i,j}(\mathbf{x})| \tag{2.162}$$

at the measurement points are listed in Table 2.7. m are the measured and $\hat{m}$ are the simulated values for all pressure sensors p_i, N_S is the number of scenarios. The difference between the measured and the simulated values are also shown as a correlation plot in Figure 2.28. The minor loss and the roughness

[20] ... which is not very likely in a system consisting mainly of plastic pipes like the Ragnitz network.

values are depicted in Figure 2.26. Due to the small differences of the values, the logarithm with base 10 of the roughnesses are additionally depicted in Figure 2.27. The minor loss of the partially-closed valve resulted in a very high coefficient of $k_i = 1384$.

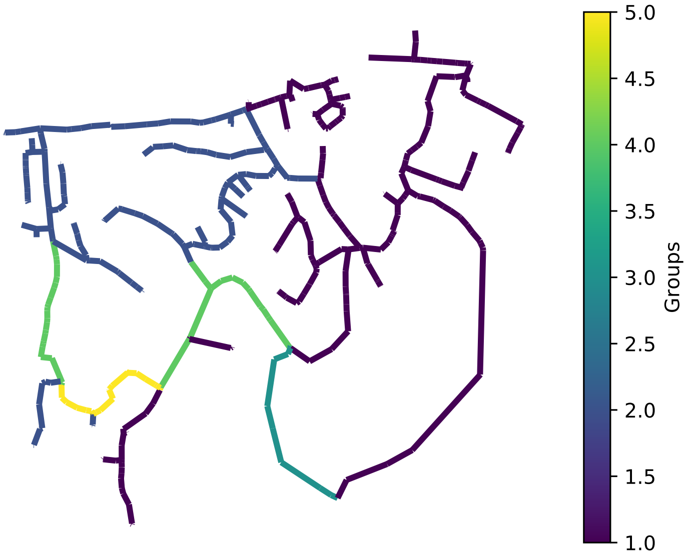


Figure 2.25.: Groups for roughness calibration

Table 2.6.: This table shows the roughness values c_i for the pipe groups from Figure 2.25 resulting from the automatic calibration approach.

Group	$\mathbf{c_i}$ [mm]
G_1	0.0148
G_2	0.0061
G_3	0.2594
G_4	0.9101
G_5	1.9980

Table 2.7.: This table shows the average absolute deviation of measurements and simulations for the optimal roughness parameter set in Table 2.6.

Sensor	D_i [bar]
HG3420	0.059
HG3445	0.043
HG3835	0.021
HG3933	0.020
HG4150	0.038
HG4162	0.016
HG4215	0.016
HG4339b	0.029
HG4383	0.018
HG4540	0.027
HG4576	0.063
HG4744	0.041

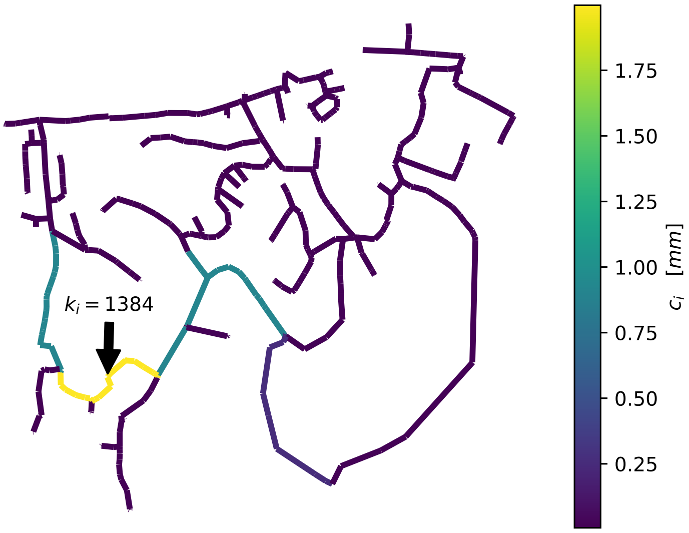


Figure 2.26.: Roughness values after calibration

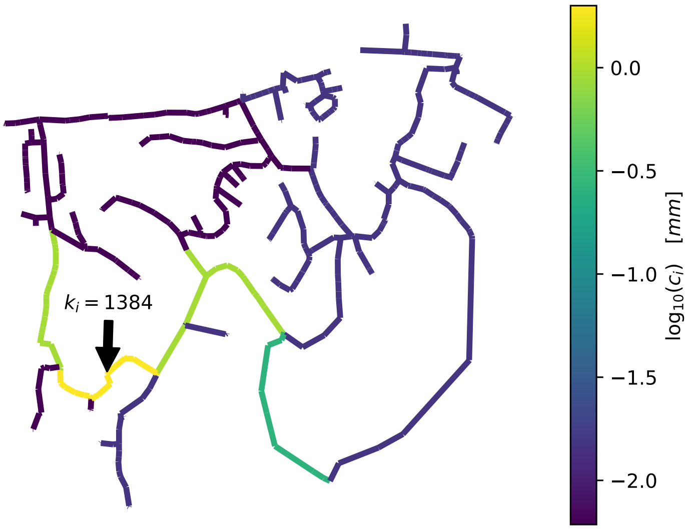


Figure 2.27.: Roughness values in logarithmic scale after calibration

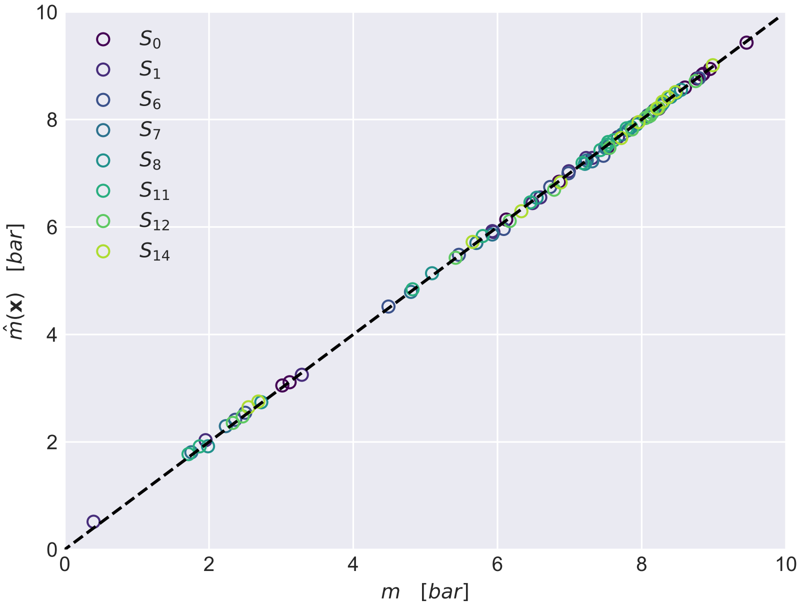


Figure 2.28.: Correlation plot for pressures with calibrated roughness values between measured m and simulated values $\hat{\mathbf{m}}(\mathbf{x})$

2.4.3.3. Leak scenarios

In addition to the calibration scenarios where huge outflows were generated, smaller scenarios have been produced serving as artificial leakages in the system. First, also in the calibration night by smaller openings of the hydrants. Second, long-time leak outflow experiments were generated with a special device allowing to measure small leakage outflows down to $Q_L = 0.25$ L/s and even smaller. This small leak datasets will act as real-world leak detection and localization benchmark scenarios in this thesis.

2.4.3.3.1. Small artificial leaks during calibration night In the calibration night, the hydrant devices measuring the outflow have been also used to generate smaller leakage outflows that are used later for leak localization. The measurement devices had a low flow measurement threshold of 1.25 L/s. The hydrants were opened until the measurement devices registered a leakage outflow, then the outflow was reduced a bit to generate scenarios with approximately $Q \approx 1.0$ [L/s]. The leakages were produced at positions A, E and F (see Figure 2.22). The generated leak scenarios are named S_{15} to S_{17} carrying on the calibration scenario nomenclature. All scenarios, their start t_{start} and end times t_{end}, inflow ΔQ and pressure measurement differences Δp and their leak positions are summarized in Table 2.8. Additionally, the inflow and pressure measurement differences to the MNF of each scenario are depicted in the same manner as before in Figures 2.29 to 2.31. Again, the timespans, where averages are taken, are highlighted in green color. The pressure response at all measurement locations is one to two magnitudes smaller than for the calibration scenarios. This can be seen if Table 2.8 is compared with Table 2.4 respectively Table 2.5. The pressure drops are even hard to see in Figures 2.29 to 2.31.

Furthermore, it can be seen that in Figure 2.30 large flow and pressure fluctuations were produced during Scenario S_{16} at around 03:04. This fluctuations resulted probably from readjusting the leakage outflow manually during scenario generation. It has to be noted that producing a constant leakage outflow out of a hydrant in a system with high pressures of up to 10 bar is obviously not as trivial as expected, because this process has been dynamically than anticipated and the measurement devices were not well suited for this purpose.

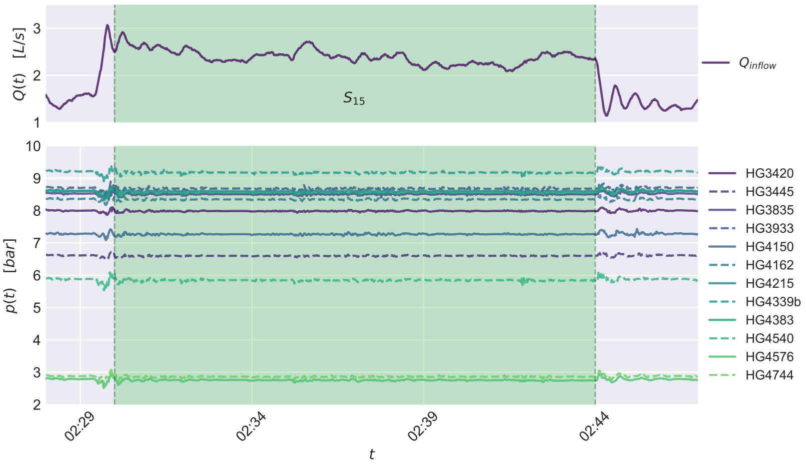


Figure 2.29.: Leak generated in Scenario 15

Table 2.8.: Leak table for smaller leak scenarios during the calibration night with start t_{start} and end times t_{end} and differences in pressure Δp and flow ΔQ to the MNF for different leak positions

	Scenario	S_{15}	S_{16}	S_{17}
	Leak Pos.	A	E	F
	t_{start}	02:30	02:57	04:12
	t_{end}	02:45	03:10	04:23
Sensor				
Inflow	ΔQ [L/s]	1.177	1.138	1.338
HG3420	Δp (bar)	-0.019	-0.018	-0.025
HG3445	Δp (bar)	-0.020	-0.020	-0.026
HG3835	Δp (bar)	-0.029	-0.025	-0.024
HG3933	Δp (bar)	-0.035	-0.024	-0.025
HG4150	Δp (bar)	-0.020	-0.020	-0.032
HG4162	Δp (bar)	-0.020	-0.021	-0.031
HG4215	Δp (bar)	-0.020	-0.020	-0.032
HG4339b	Δp (bar)	-0.047	-0.025	-0.024
HG4383	Δp (bar)	-0.046	-0.025	-0.023
HG4540	Δp (bar)	-0.045	-0.025	-0.023
HG4576	Δp (bar)	-0.041	-0.036	-0.060
HG4744	Δp (bar)	-0.023	-0.022	-0.022

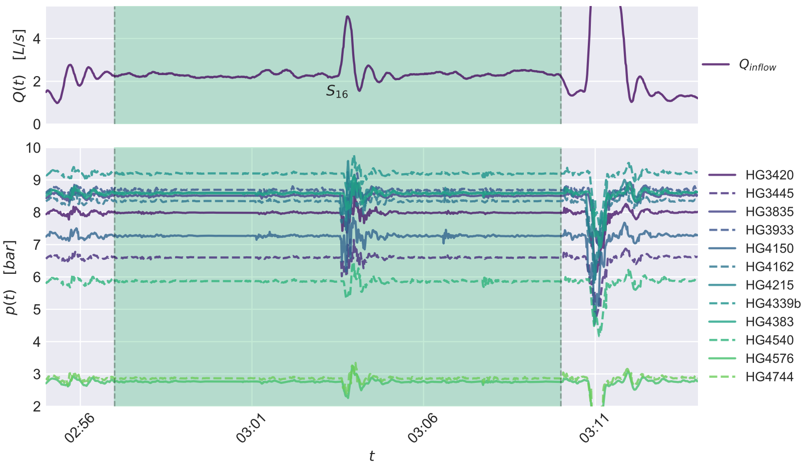


Figure 2.30.: Leak generated in Scenario 16

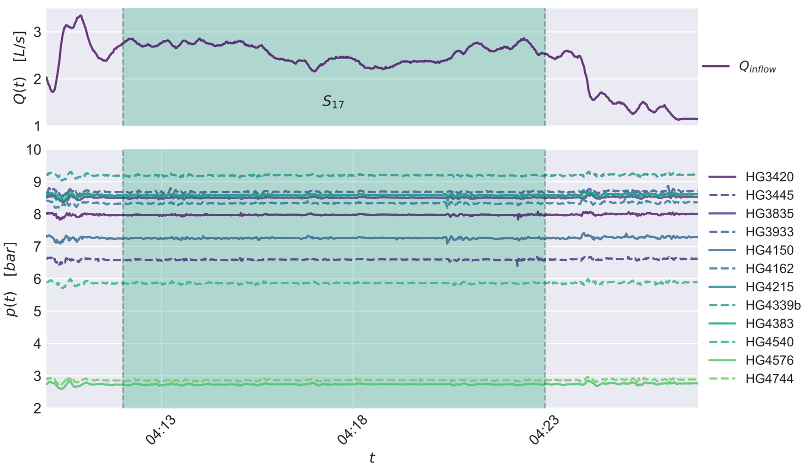


Figure 2.31.: Leak generated in Scenario 17

2.4.3.3.2. Small long-time artificial leaks Soon after the calibration night measurements, artificial leaks with small leakage outflows were introduced in the system—aiming to produce measurement data for small leak scenarios with long run-times. For that reason, a special measurement device has been build to enable the production and measurements of such small leaks. The same pressure and flow sensors as before were used in the WDS of Ragnitz, although approximately after one and a half month, the number of pressure measurements had to be reduced to six sensors, since the WU needed the measurement devices for other daily business operations.

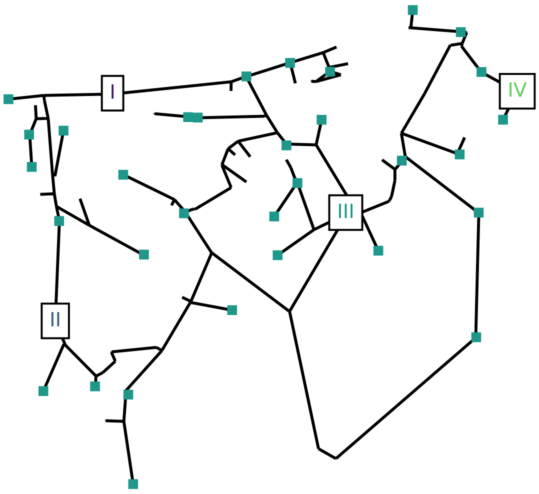


Figure 2.32.: Locations of artificial leaks in Ragnitz-Simple network labeled with Roman numbers from I to IV

The experimental setup for producing and measuring small leaks consisted of a fire hose of 15 meters length attached to the hydrant at the planned leak positions. Further attached to this fire hose was an approximately two meter long PVC pipe with a diameter of 32 mm. After a calming section, an

UFM was clamped on onto this PVC pipe for measuring accurately the leak outflow. The type of this UFM was a "Prosonic Flow 92" with a build in data logger charged by a car battery. It has to be noted that this outflow was only measured for obtaining controlled experimental conditions and that this information was not used for leak detection nor for localization.

The long-time leak experiments took place in a time-frame of three months between April and July 2016. Leaks at four different positions were produced in the Ragnitz network. The leak locations are depicted in Figure 2.32 and labeled with Roman numbers from I to IV. Additionally, different scenarios were produced at these locations with different leak magnitudes. The scenarios are named from S_a to S_l. The magnitudes ranged between 0.25 and 1.0 L/s. Leak run-times were between two days (49 hours) and more than a week (188 hours)—in the long-term scenario S_k at leak position I with a magnitude of 0.7 L/s. All leak scenarios are summarized in Table 2.9. In total, 1952 m^3 of water were dissipated during the long-time leak scenarios. This is approximately six and a half complete fillings of the measurement zone's tank[21]. In general, the leak scenarios were changed approximately every third day. Between the mid of May and the beginning of July a longer period existed, where the leak free system could be observed.

Unfortunately, problems occurred during the first scenario S_a. The internal storage of the pressure measurement devices had been overestimated, leading to a memory overflow in the internal ring memory which resulted in a measurement data loss almost during the whole first scenario. This is why this scenario will not be used for leak localization. Anyway, experiments with the same leak magnitude at the same positions have been repeated in scenario S_d after the storage fault was recognized.

Furthermore, it has to be mentioned that the measurement campaign fell luckily in a period with few National holidays. The leak scenario S_g produced on the National holiday at the first of May has been a Sunday and hence is supposed not to be different to a normal Sunday. Only scenario S_j contains a holiday day (the 16 of May) which has been on a Monday, hence, the customer behavior may differ from a normal Monday.

Table 2.9.: Overview of the long-time artificial leak experiments in the Ragnitz network

	Pos.	**Hydr.**	$\approx$ **Q$_L$**	**t$_{start}$**	**t$_{end}$**	**t$_D$**
Scenario			[L/s]			[hours]
S_a	I	HG3880	0.50	15.04 11:15	17.04 12:32	49
S_b	I	HG3880	0.70	17.04 12:33	20.04 09:49	69
S_c	I	HG3880	0.30	20.04 09:51	22.04 09:47	47
S_d	I	HG3880	0.50	22.04 10:35	25.04 10:51	72
S_e	II	HG4504	0.25	25.04 12:15	28.04 15:16	75
S_f	II	HG4504	0.50	28.04 15:19	01.05 16:12	72
S_g	II	HG4504	0.70	01.05 16:13	04.05 08:29	64
S_h	III	HG3164	0.50	09.05 11:30	12.05 14:37	75
S_i	III	HG3164	1.00	12.05 14:38	15.05 10:40	68
S_j	III	HG3164	0.25	15.05 10:51	18.05 14:38	75
S_k	I	HG3880	0.70	04.07 12:20	12.07 09:15	188
S_l	IV	HG4118	0.90	12.07 10:21	15.07 09:44	71

[21] The tank volume in Ragnitz is 300 m^3.

Chapter 3

Leak Detection

“Time is an illusion. Lunchtime doubly so."

— Douglas Adams, *The Hitchhiker’s Guide to the Galaxy*

3.1. Literature review

Modern sensors and data communication technologies enable—while at the same time tight financial budgets urge—WUs to save on human resources. Whereas in the past, human operators were in charge of small single supply areas, nowadays, modern supervisors are simultaneously responsible for multiple WDS locations (Bakker, Vreeburg, Roer, et al. 2014). The increasing distance between the human operator and the physical WDS leads to a growing risk of failures being unrecognized for long times.

Data analysis methods processing pressure and flow measurements in an automatic way are of particular interest for providing a rapid response to pipe burst and leaks (Romano et al. 2013). These techniques can efficiently handle huge amounts of data in a way that no single human is capable of. The data is preprocessed by extracting only useful and relevant information for making operational decisions.

Several different approaches exist in scientific water related literature addressing this topic—some of these approaches are presented here in this section. Most of them make use of statistical methods, soft computing algorithms like Artifical Neural Networks (ANN) or time series models. For instance, S. R. Mounce and Machell 2006 used ARIMA models to get rid of diurnal effects in the measurement data and to impute missing values. Subsequently, static and time delay ANNs—where the latter performed better—were used. The data was artificially generated by flushing hydrants during the MNF with leak outflows of 1 to 5 L/s. This corresponded to 2 to 10 % of the average inflow in the system. Flow and pressure measurements at the inflow and outflow of the DMA were used with a time resolution of one minute to detect these events. After all, the method was able to detect 75 % of the leaks correctly.

S. R. Mounce, R. B. Mounce, et al. 2011 used Support Vector Machiness (SVMs) to detect unusual flow and pressure fluctuations. The method was able to detect bursts, sensor failures as well as hydrant flushing events of known size (2 L/s) in the system. The outflow corresponded to 6 to 12% of the average inflow. Furthermore, S. R. Mounce, R. B. Mounce, et al. 2011 found that pressure is a less reliable parameter than flow for event detection and that the response of a particular meter is strongly connected to its location in the system.

Ye and Fenner 2011 used an adaptive Kalman filter on pressure and flow data to model normal water usage. The seasonal effects were eliminated by subtracting the measurement values of the past week. Residuals of the filter indicated abnormal water usage caused by a burst. The method was applied on (i) engineered burst tests by opening hydrants between 2.5 and 6.2 L/s; and on (ii) data from 10 real DMAs verified by using customer complaint and repair logs of the WU. Most of the bursts were detected within 15 minutes. The burst had large burst sizes of 10 % to 50 % of the average inflow. The results showed that flow measurement data is more sensitive to a burst or a leak than pressure data. Furthermore, combinations of pressure and flow data was not successful since it did not improve the detection performance at all.

Eliades and Polycarpou 2012 used an adaptive forecasting methodology by updating Fourier coefficients. Subsequently, a detection logic utilizing the CUSUM method was used on the first Fourier coefficient. This coefficient represents the offset of the data and is linked to a leak in the system. The methodology was applied on simulations based on real flow data where leaks were simulated with sizes of 1.5 % to 10 % of the average inflow. The algorithm was able to detect leaks of sizes down to 0.4 L/s. However, Eliades and Polycarpou 2012 stated that the detection of this small leak sizes might not be possible in practice because of uncertainties in customer demand and measurement noise. Furthermore, the detection time for leaks was slow taking 10 days in average, yet, 15 % of the small leaks were detected after three weeks.

Romano et al. 2013 made use of ANNs with multivariate Gaussian mixture and Bayesian inference models. Additionally, the approximate location of the leak is retrieved through application of geostatistical techniques. The method was tested on burst data generated by hydrant openings in a real-world DMA (see Chapter 5).

Bakker, Vreeburg, Roer, et al. 2014 developed a heuristic burst detection method utilizing an adaptive water demand forecasting model for flow data together with a dynamic pressure drop-demand relation estimator (Bakker, Vreeburg, Schagen, et al. 2013). One point forecasts were made for the next timestep 15 minutes into the future. Historical data of five years was used of three rather large areas (larger than usual DMAs) with average zonal demands ranging from 9 L/s for the smallest zone up to 640 L/s for the largest zone. The smallest detected burst in the smallest zone was 2 L/s and 42 L/s for the largest zone. Bakker, Vreeburg, Roer, et al. 2014 stated that the method is applicable anywhere provided that one year of historical data is available to construct the adaptive forecast model. Additionally, they found that pressure was not very valuable for burst detection, because the effect of burst on pressure was much less profound than its effect on flow measurements. Nevertheless, the model was able to detect critical bursts in the historical data at an earlier stage than customer reports or the WU detection capabilities. Especially, small bursts and bursts occurring during the nighttimes resulted in long run-times.

D. Jung, D. Kang, et al. 2015 utilized different univariate and multivariate statistical process control methods, for example, western electric company rules, CUSUM or exponentially weighted MA, Hotelling T^2 methods and mutlivariate versions of CUSUM and exponentially weighted MAs. The methods were tested on artificial generated data sets and on the same historical data as Bakker, Vreeburg, Roer, et al. 2014. It was shown that univariate exponentially weighted MA performed best. Furthermore, D. Jung and Lansey 2015 used a non-linear Kalman filter with nodal group estimation on artificial generated data for burst detection and showed that this approach performed better than a linear Kalman filter.

Y. Wu, Liu, et al. 2018 developed a clustering-method utilizing cosine distance to find dissimilarities in data obtained from multiple flow sensors. Nevertheless, the method was only sensitive to relatively large bursts.

Of course, other data driven and soft computing approaches can be found in literature, including self-organizing maps coupled with ANNs (Aksela et al. 2009), principal component analysis (Palau et al. 2011), fuzzy interference with ANN (S. R. Mounce, Boxall, et al. 2010) or Bayesian demand forecasting systems (Hutton and Kapelan 2015), just to name a few. An excellent, recent and more detailed review on this topic summarizing various data-driven approaches for pipe bursts detection can be found in Y. Wu and Liu 2017.

3.2. Methodology

3.2.1. Overview of methods

All methods used in this chapter have already been introduced in Section 2.3. This section sets the methods into context for model-based leak detection and gives a short overview how the methods are used in the results section. Model-based means in the connection to leak detection not that a hydraulic model is involved, it means that the measurement data is modeled with TSA techniques. TSA is used to extract relevant information from the measurements by decoupling the leak signal from measurement noise and random fluctuations. An overview of the used techniques and the corresponding sections in the results can be found in Figure 3.1. The methods are depicted as squares, the corresponding sections are depicted in the background with colors, round edges and numbered; the single parts of the time series data that is used for—or results from—specific methods are depicted as circles.

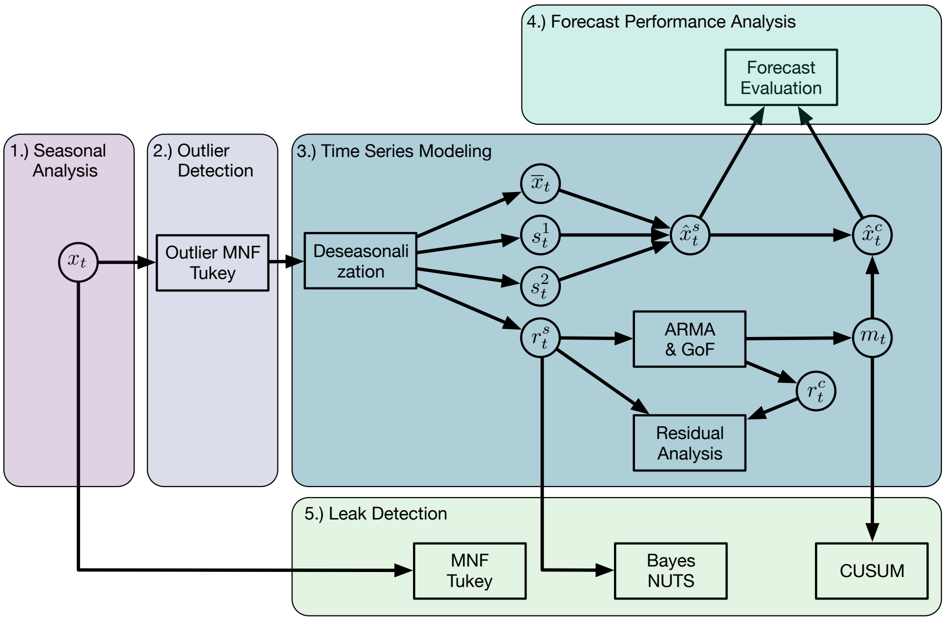


Figure 3.1.: Overview of the leak detection methodologies

First, the original time series x_t represented by the plain measurement values (flow or pressure) is examined in detail—especially the seasonal effects are exposed in a **seasonal analysis** (see Section 3.3.1)—through applying the cyclic operators defined in Section 2.3.5.1 in equation (2.95).

Second, outliers are identified in the measurement data on the example of unusual demand during MNF hours (see Section 3.3.2 for the results). The **outlier detection** is performed with Tukey's range test as defined in Section 2.3.3.

Third, the measurement data is modeled with TSA techniques. At first, the **time series modeling** is performed through deseasonalizing x_t as already discussed in Section 2.3.5.3. x_t is split up in components: its mean $\overline{x_t}$, two seasonal components s_t^1 and s_t^2—forming together a simple time series model $\hat{x}_t^s$—and the residual component r_t^s. The residuals are further analyzed by looking at the ACF and PACF defined in Section 2.3.5.2. As a result of this analysis, the residuals are further modeled through ARMA models (see Section 2.3.5.5) to extract correlation effects between subsequent data points. The best ARMA model is chosen by GoF statistics (see Section 2.3.4), namely, AIC and BIC. This results in a division of the modeled time series in (i) m_t which can be seen as a trend function, and (ii) a random component r_t^c. m_t together with the seasonal model $\hat{x}_t^s$ forms the complex model $\hat{x}_t^c$—describing the behavior of x_t in a more advanced way than $\hat{x}_t^s$.

Fourth, the simple and the complex model are evaluated in a **forecast performance analysis**. For that reason, the models are used to forecast x_t by utilizing data that has not been used priorly for fitting the models. $\hat{x}_t^s$ and $\hat{x}_t^s$ are evaluated using the forecast metrics described in Section 2.3.5.7.

Finally, the different parts of the models are used for **leak detection** utilizing different algorithms. A focus of this analysis is on the algorithms described in Section 2.3.6—answering following relevant questions: (i) how fast is the algorithm, (ii) how sensitive, (iii) do certain parameters have to be tuned and how complicated is the parameter tuning and (iv) is the algorithm capable of finding small leaks in flow and/or pressure measurements. These questions are relevant for the research question of this chapter—if an automated detection of a small leak is possible.

3.2.2. Measurement data

The measurements start at the 30^{th} of May 2016 and end at the 12^{th} of July and is depicted in Figure 3.2. A long leak-free period is followed by a small leak—starting at the 4^{th} of June within the data recorded during the field-study. This leak serves as a test for the detection algorithms.

The data is partitioned in three phases: (i) the training, (ii) the test and (iii) the leak phase.

(i) The first phase between the 30^{th} of May and the 26^{th} of June is the **training** phase, which lasts for four weeks in total. This phase is used for analyzing the measurements in detail, for finding the optimal time series model with TSA algorithms and for fitting this model to the measurement data.

(ii) The second phase between the 27^{th} of June and the 3^{rd} of July is called the **test** phase. It lasts for one week. Within this phase the models retrieved in the first phase are tested regarding their forecast performance respectively their forecast error under normal conditions.

(iii) The third phase between the 4^{th} of July and the 12^{th} of July is the **leak** phase. An artificial leak is introduced in the system. In this phase different leak detection algorithms are applied on the data to test if and when they detect the leak—both on flow and pressure measurement data.

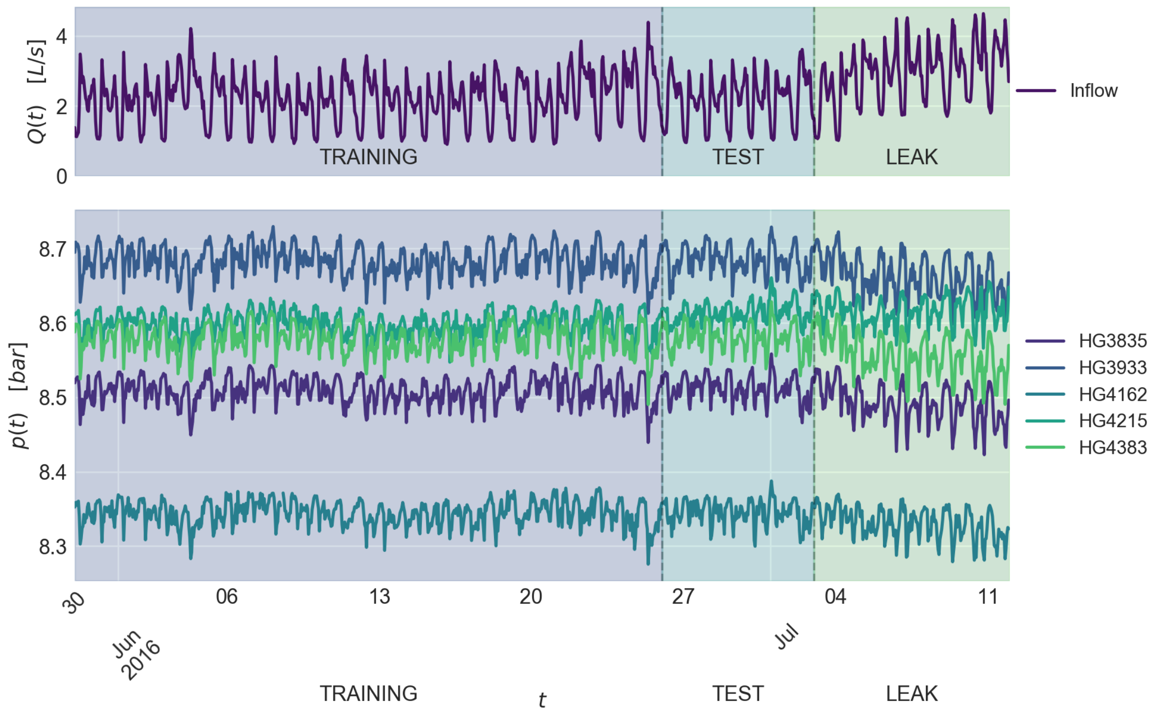


Figure 3.2.: Overview of flow (top) and pressure (bottom) data used for TSA and leak detection. The training, the test and the leak data sections are highlighted

All three phases—training, test and leak—are additionally highlighted in Figure 3.2. The inflow measurements are on the top, the pressure measurements are at the bottom of the figure. Six pressure sensors were installed in the system during this period. The sensor positions result from the optimal sensor placement algorithm of Casillas in Section 4.2.2.3.3 and the SPUDU algorithm in Section 4.2.2.4—for five sensors each. Only six sensors were necessary for both placements, since four sensor positions of the Casillas and SPUDU algorithm coincide. The pressure data is of high time resolution with a measurement each minute. The flow measurements have the same resolution of one minute as the pressure measurements. The measurement values are accumulated by building the mean over a 15 minute time period—all leak detection algorithm experiments use this 15 minute data. Additionally, results of experiments with higher (5 minutes) and lower (1 hour) time resolution are discussed.

The sensors were positioned at following hydrants: HG3420, HG3835, HG3933, HG4162, HG4215 and HG4384 (see Figure 2.21 in Section 2.4 for the locations corresponding to the hydrant numbers). Unfortunately, the sensor at HG3420 malfunctioned during this period. Thus, HG3420 is not depicted in Figure 3.2. Additionally, sensor HG4215 started to malfunction shortly after this measurement period. Even before, the sensor delivered false measurements—it is the only sensor where the pressure increases when the leak was introduced in the system. Consequently, this sensor is not used for leak detection and localization. In total, four well-functioning pressure sensors remain for the leak experiments.

The artificial leak is introduced in the system on the 4^{th} of July at 12:20 at hydrant HG3880 with a magnitude of approximately $Q_L \approx 0.7$ (L/s)—see leak scenario S_k in Table 2.9 in Section 2.4. The leak location at HG3880 corresponds to position I in Figure 2.32.

3.2.3. Pressure tank level correction

The water level fluctuations of the tank are in a scale of 2.5 meter—or approximately a quarter bar. This is in the magnitude (or even a magnitude higher) of the pressure drop caused by the leak. The fluctuations have an additive effect on the pressure measurements in the system. Thus, the tank level units are transformed from meter to bar and, subsequently, subtracted from the pressure measurements to remove this additive effect.

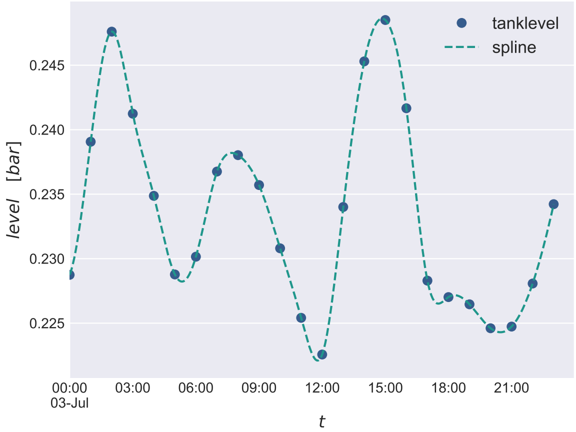


Figure 3.3.: The water level of the tank converted from meter to bar and approximated with a cubic splines

Unfortunately, the temporal resolution of the tank level measurements is not as high as the resolution of the pressure measurements. Periods exist where a level measurement is available once an hour. Thus, the water level data is approximated through cubic splines to estimate the data at timestamps between the measurements. An example can be seen in Figure 3.3 where the data is already transformed to bar with the procedure described in Section 2.4.1.1.

3.3. Results and discussion

This subsections in the results section strictly follow the numbering scheme in Figure 3.1.

3.3.1. Seasonal analysis

Figure 3.4 shows the inflow $Q(t)$ over time in the Ragnitz system in a time resolution of one minute during the model training period of three weeks between the 30th of May 2016 and the 26th of June 2016. Additionally, the moving average MA_{1H} over one hour and over one day MA_{24H} is depicted to decrease random fluctuations in the data. The figure shows that wider moving average windows w follow the measurements slower.

The inflow measurements summarize the collective water consumption of all customers in a measurement zone plus an offset resulting from leakage outflows of not detected leaks. Figure 3.4 shows that the variations of the demand over time possess different fluctuating time scales (Walski et al. 2003).

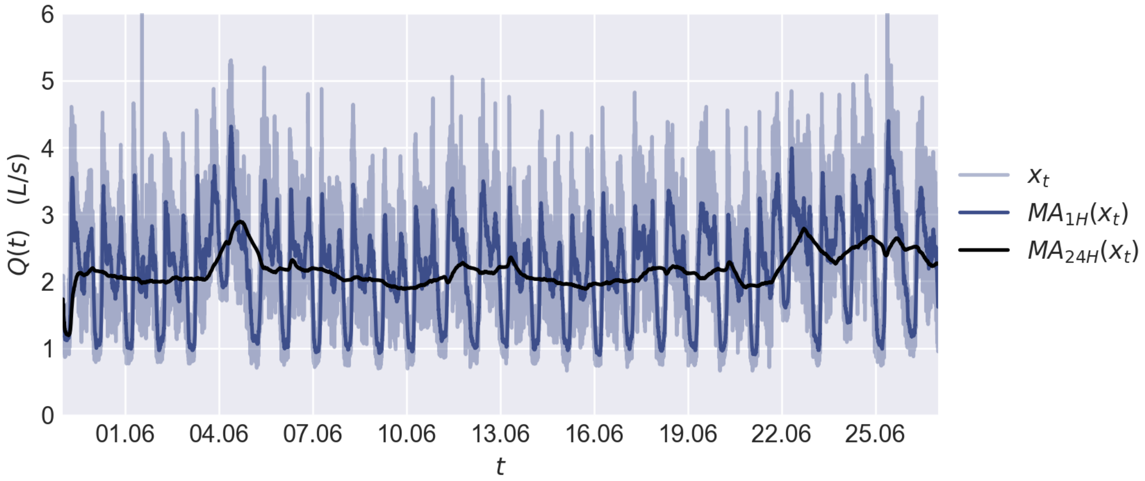


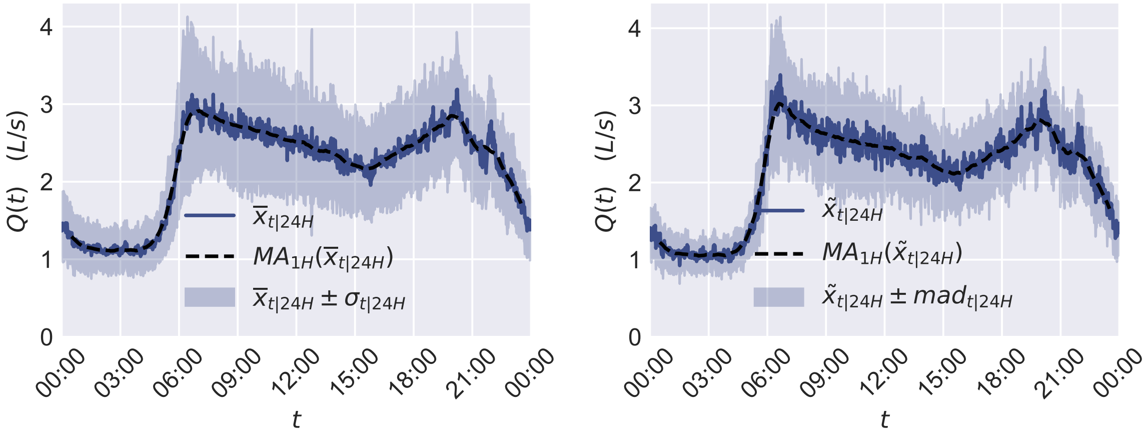
Figure 3.4.: Inflow in of the Ragnitz network

The first time scale is a **daily** variation of demand according to differences in water usage of customers over the course of one day. By applying cyclic operators as defined in equation (2.95) with a period length of $P = 24$ hours, this effect is revealed (see Figure 3.5). Figure 3.5a shows the daily cyclic mean $\bar{x}_{t|24H}$ and the corresponding cyclic standard deviation $\sigma_{t|24H}$, while Figure 3.5b depicts the cyclic median $\tilde{x}_{t|24H}$ and the cyclic $\text{mad}_{t|24H}$. Additional moving averages with a window size of $w = 1H$ are shown. Figure 3.5 shows that the median and mad is more robust to fluctuations and outliers than the mean and the standard deviation σ. The measurements reveal that water consumption is low during nighttime. The MNF is approximately $\overline{Q}_{MNF} \approx 1.0\ (L/s)$. In the morning, after inhabitants in the measurement area awake, the consumption reaches its maximum. The reasons for this morning peak are people taking showers, using toilets or preparing breakfast. Subsequently, most people leave the measurement zone and go to work. Note that the zone is a residential neighborhood and not a commercial area. The inflow in the system decreases until the inhabitants return in the afternoon, which results in a second peak. This customer behavior is nearly the same every day.

The second important periodical effect in the inflow time series can be seen on a **weekly** time scale. The diversity arise from differences in consumption on weekdays and weekends. Consequently, this effect is revealed by splitting the cyclic daily averages into the corresponding days of the week. Figure 3.6 shows that the morning peak starts later at weekends. Already at Friday afternoon, the inflow in the system shows a different behavior compared to other weekdays.

As a consequence, an accurate time series model has to account at least for these two time scales—daily and weekly. Additional seasonal effects exist in water demand. For example, water consumption during dry weather periods in summer (gardens, swimming pools) is substantially higher than water consumption during winter. However, the field test took place during spring and summer 2016. Thus, these seasonal effects can be neglected. However, the discussed TSA methods are capable of integrating additional seasonalities without additional effort.

To obtain accurate pressure measurements, the data was preprocessed in this chapter. This preprocessing consisted of subtracting water level measurements (see Section 3.2.3). The subtraction isolates the pressure measurements from tank water level influences. Figure 3.7 compares the inflow $Q(t)$ and the pressure $p(t)$ averaged over all measurement station in the zones. The pressure shows an



(a) Mean and standard deviation

(b) Median and median absolute deviation

Figure 3.5.: Daily (cyclic) inflow pattern in Graz Ragnitz

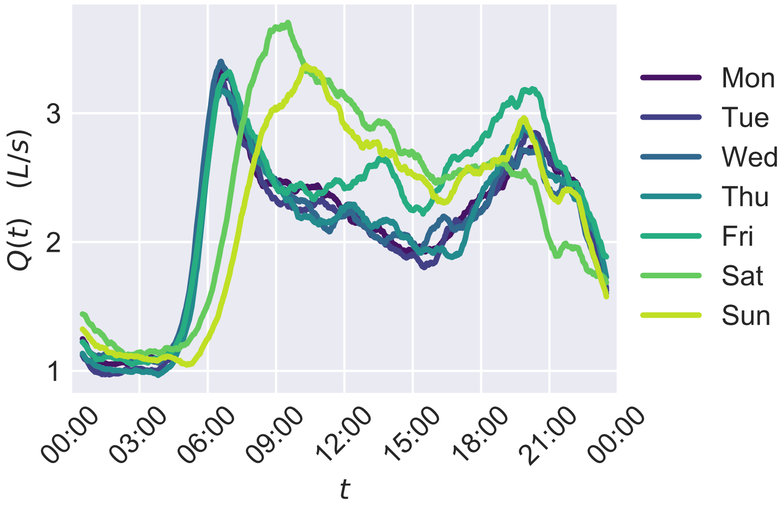


Figure 3.6.: Differences in the hourly moving average of cyclic daily pattern $\mathrm{MA}_{1H}(\bar{x}_{t|24H})$ on different weekdays

inversely proportional behavior to the inflow. $p(t)$ is highest during MNF hours—during the morning peak it possesses its minimum. A second minimum occurs in the afternoon. Additionally, the pressure also possesses the same weekly periodical effects as the inflow, but these seasonalities are not shown here.

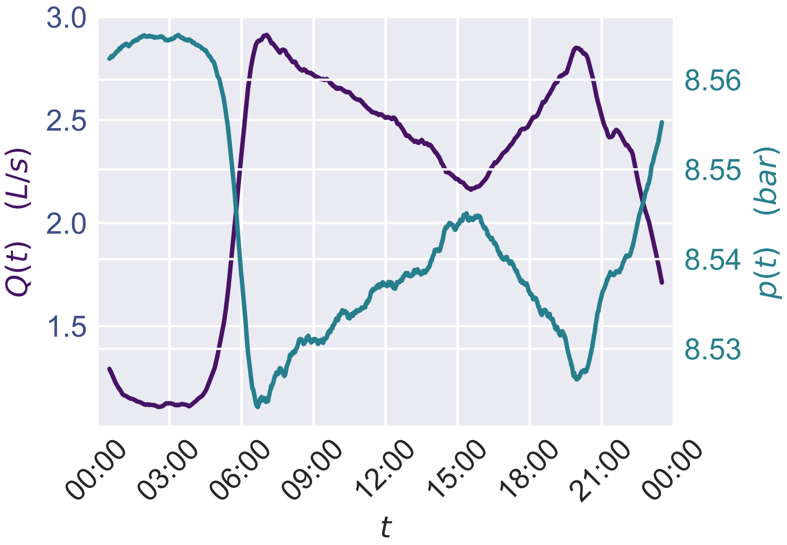


Figure 3.7.: Relationship between inflow $Q(t)$ (left y-axis) and pressure $p(t)$ (right y-axis) averaged over all pressure measurement stations—depicted as $\mathrm{MA}_{1H}(\bar{x}_{t|24H})$

3.3.2. Outlier detection

Water demand is not the same every day (see Figure 3.4). Besides daily and weekly variations, water demand differs, for two consecutive Mondays, for example. Figure 3.8 emphasis this effect by depicting the inflow in the system as boxplots for each day. The upper part of the figure shows the inflow data as hourly mean $Q(t)$, the middle part shows boxplots of the inflow during the whole day and the boxplots at the bottom show the inflow during MNF. The MNF was found to be between 2:00 and 4:00 in the morning in the measurement zone.

There are days with unusual high water demand. For calibration and leakage localization it is important to separate days with normal water consumption from those days. This section presents an automatic procedure to identify those days based on Tukey's fence test (Section 2.3.3). This test is applied on inflow measurements to automatically identify days with high demand.

Figure 3.9 presents the results of Tukey's test on the inflow data. The data points are the mean values over the MNF $\overline{Q}_{MNF}(t)$. The quartiles of the data result in $q_1 = 1.011$ (L/s) respectively $q_3 = 1.118$ (L/s). This leads to the outlier boundaries of $[0.852, 1.278]$ (solid line), respectively, to the far outlier boundaries of $[0.692, 1.437]$ (dashed line) in Figure 3.9. Two data points surmount the far outlier boundaries—three the normal outlier boundaries. All remaining data points are within the boundaries. The dates of these data points and their corresponding $\overline{Q}_{MNF}$ are listed in Table 3.1. Interestingly, the outliers are all on different days of the week. Consequently, it does not result from weekly seasonality.

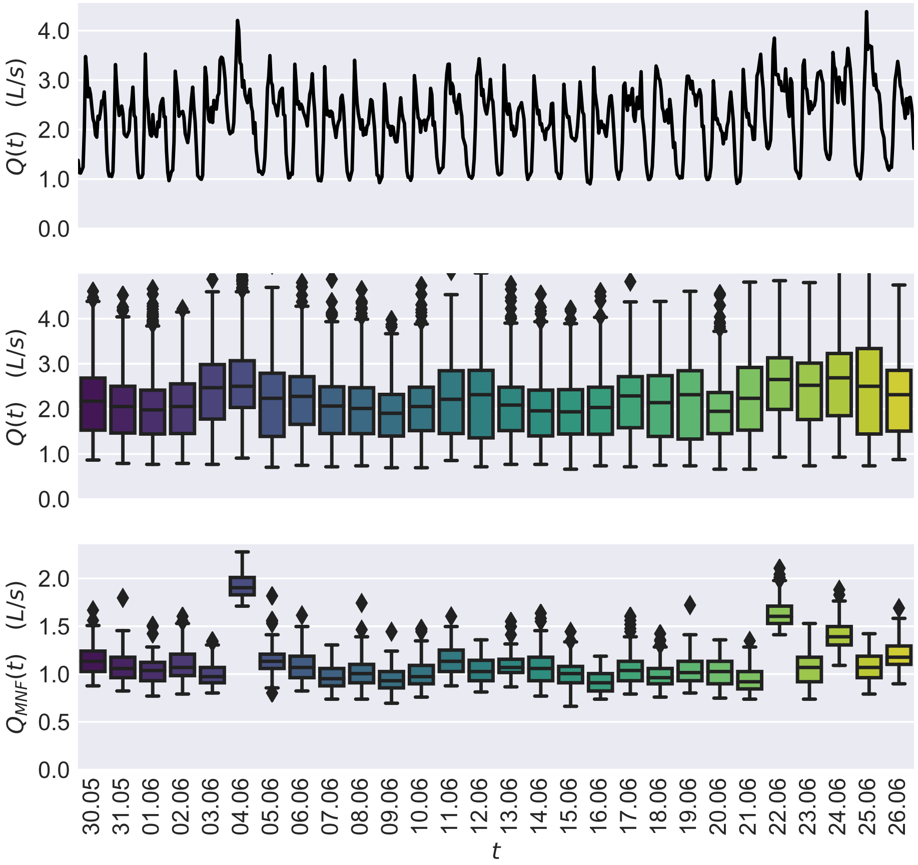


Figure 3.8.: Boxplots showing outliers

The effect when those outliers are deleted from the data can be seen in Figure 3.10. The minutely measured inflow with outliers is depicted in Figure 3.10a as daily mean $\overline{x}_{t|24H}$. Additionally, the sample standard deviation $\sigma_{t|24H}$ and a moving average $\mathrm{MA}_{1H}(\overline{x}_{t|24H})$ with a one hour window size are shown. Figure 3.10b shows the data without outlier days. As a result, the standard deviation becomes smaller without outliers. Figure 3.11 shows the data during the MNF. Again, the standard deviation during the MNF is smaller. Additionally, $Q(t)$ slightly decreases in the outlier free data (Compare Figure 3.11a and 3.11b).

For pressure data (measured at hydrant HG3933), the effect of high inflow data is negligible, as can be seen by comparing Figure 3.10c and 3.10d, respectively Figure 3.11c and 3.11d. Consequently, outlier have to be removed when inflow data is used (e.g. for generating inflow pattern for hydraulic

Table 3.1.: Outliers identified by Tukey's test in the daily mean values of the MNF inflow data

date	$\overline{Q}_{MNF}$ (L/s)	Weekday
2016-06-04	1.927	Sat
2016-06-22	1.644	Wed
2016-06-24	1.405	Fri

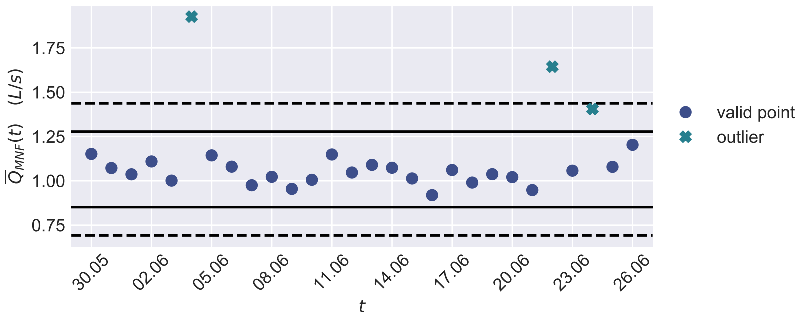


Figure 3.9.: Tukey test on the mean of the MNF data $\overline{Q}_{MNF}(t)$

simulations) and not for pressure data. The pressure seems more robust to high demands.

3.3.3. Time series modeling

Time series analysis techniques are most commonly used for water demand forecasting (Donkor et al. 2014; Saludes et al. 2017).

In this section, a model is developed that describes the flow and pressure time series in the measurement zone Graz Ragnitz in an optimal way. This model is capable of dividing the measurement time series in trend, seasonal and random components. Finding the optimal model and fitting the parameters is an optimization task, solved with methods described in Section 2.1. First, the section will start by dividing the measured time series in seasonal components, followed by reducing autocorrelations in the remainder. With the proposed method, the deterministic effects will be separated from random fluctuations.

3.3.3.1. Deseasonalization

The deseasonalization method—as described in Section 2.3.5.3—is applied on the inflow data x_t to split the time series in its mean $\overline{x}_t$, its seasonal components—the daily seasonality s_t^1 respectively the weekly seasonality s_t^2—and the remainder or residual component r_t^s. The days with unusual demand found in Section 3.3.2 (the 4^{th}, 22^{nd} and 24^{th} of June) are not considered for computing the seasonalities.

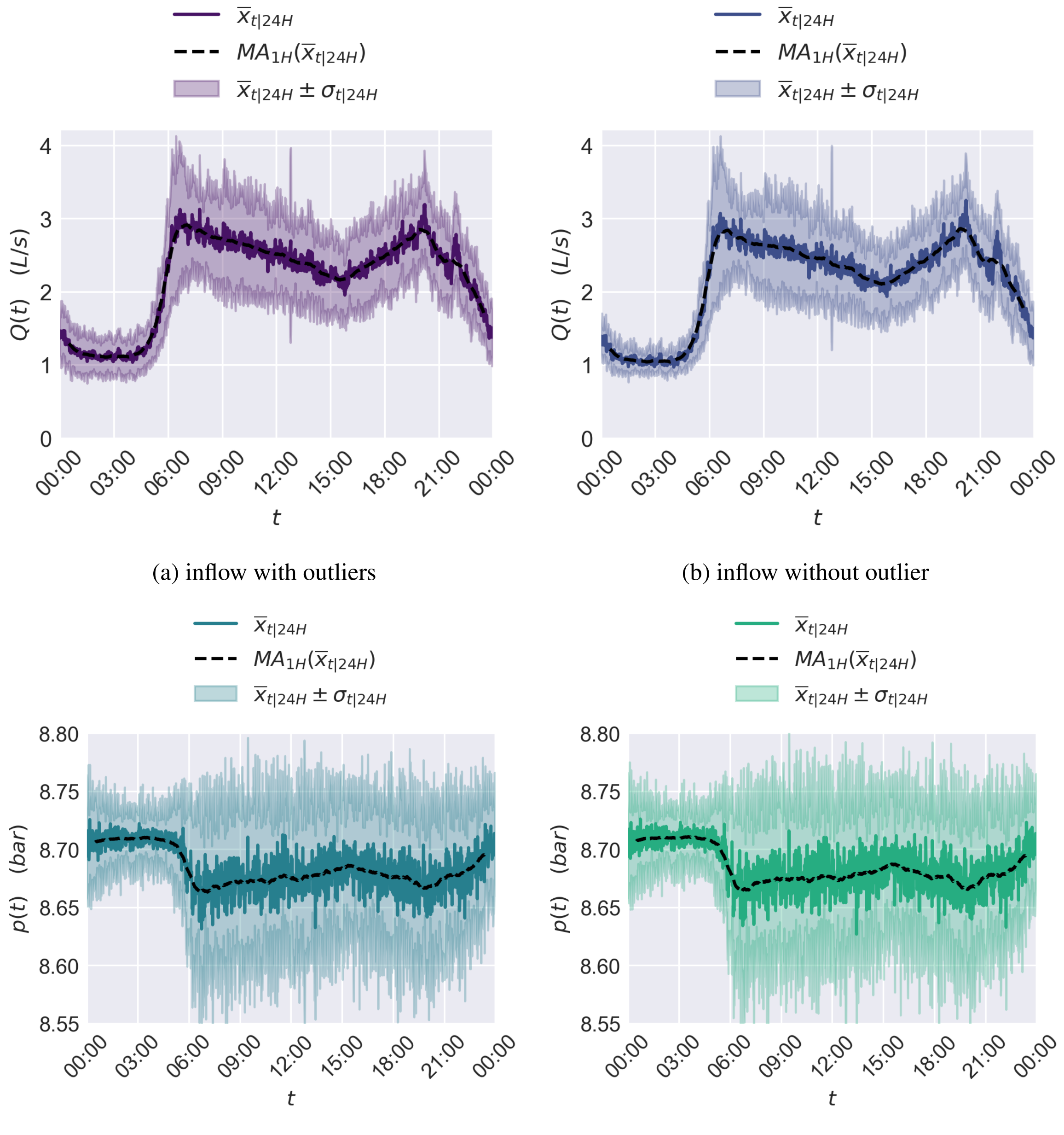
The three former parts can be combined to obtain a simple seasonal model $\hat{x}_t^s$ of the measurement data

$$\hat{x}_t^s = \overline{x}_t + s_t^1 + s_t^2 \quad . \tag{3.1}$$

Consequently, the residuals of the simple model result from the mismatch of the data and the model

$$r_t^s = x_t - \hat{x}_t^s \quad . \tag{3.2}$$

Figure 3.12 shows the components resulting from the deseasonalization procedure (x_t, s_t^1, s_t^2, $\hat{x}_t^s$ and r_t^s) of the inflow time series $Q(t)$ in the measurement during the training period. Despite just being the summation of the two seasonalities, $\hat{x}_t^s$ shows a similar behavior than the original time series x_t. $\hat{x}_t^s$ can be applied for producing weekly demand patterns used for EPS simulations. These simulations can be



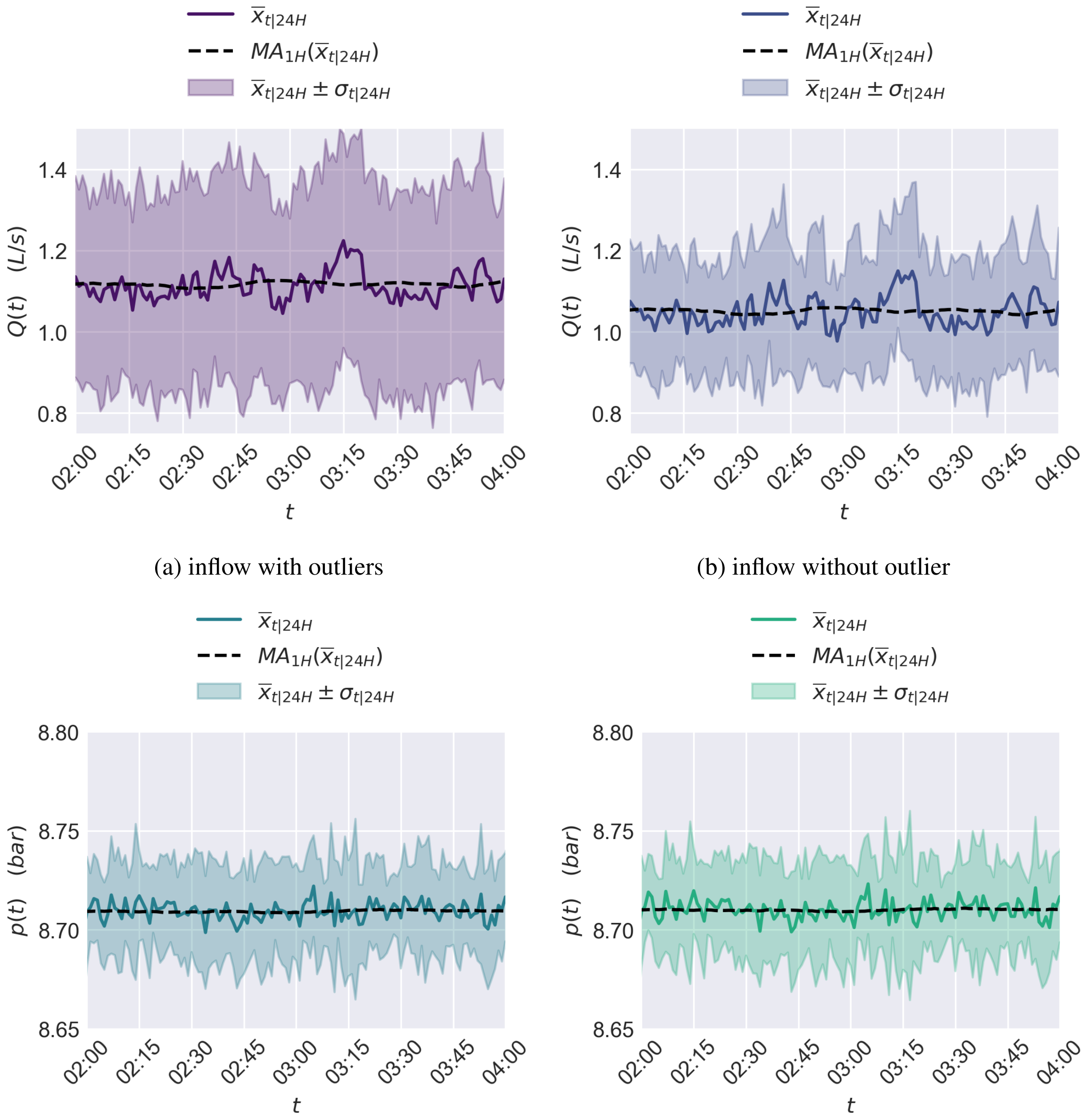
(a) inflow with outliers

(b) inflow without outlier

(c) pressure with outliers

(d) pressure without outlier

Figure 3.10.: Comparison of inflow and pressure measured at hydrant HG3933 with and without outliers



(a) inflow with outliers

(b) inflow without outlier

(c) pressure with outliers

(d) pressure without outlier

Figure 3.11.: Comparison of flow and pressure measured at hydrant HG3933 with and without outliers during MNF (2:00 - 4:00)

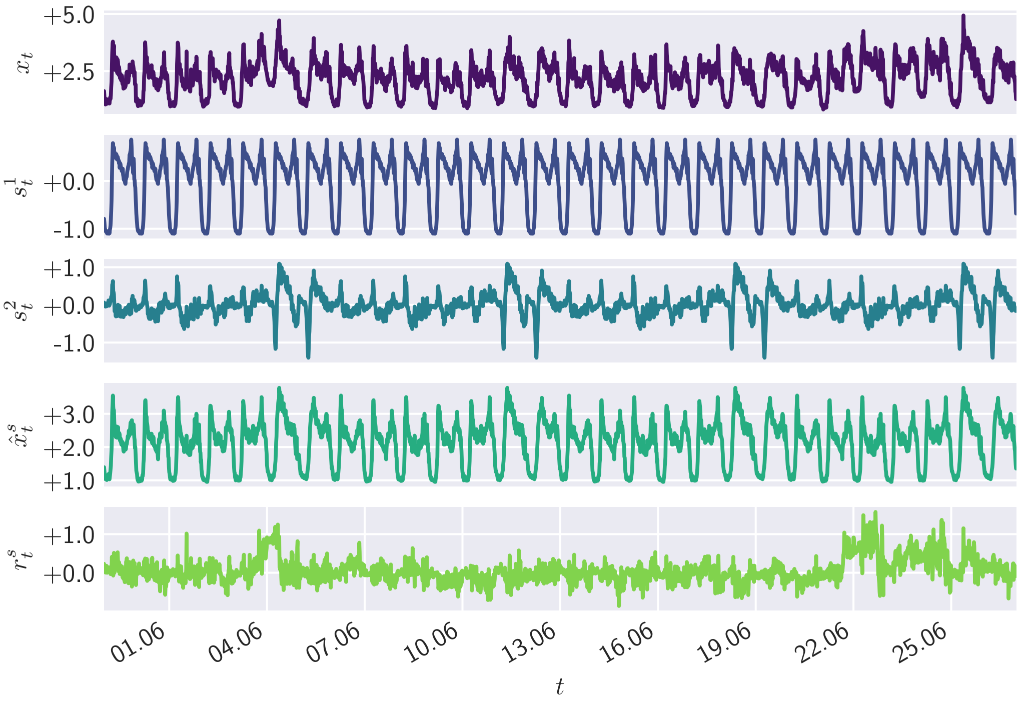


Figure 3.12.: Components of the deseasonalized time series of Ragnitz inflow measurement $Q(t)$. The units in all graphs are L/s

used, for instance, for evaluating the performance of a WDS during normal demand loads—especially for analyzing the impact of future system adaptations or optimization of pump schedules to improve the WU's energy savings and carbon footprint.

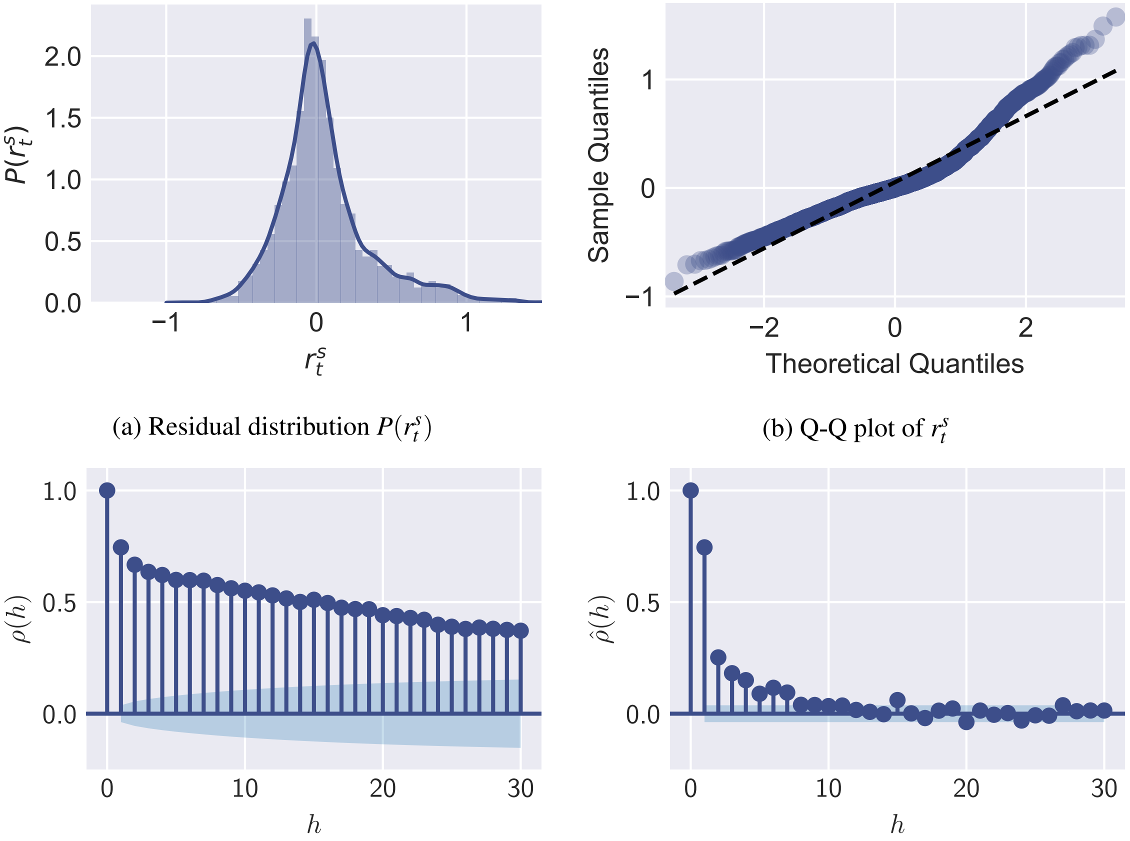
Higher water consumption trends can be seen in the residual component r_t^s, although the signal is additionally perturbed by high noise. Section 3.3.3.3 shows how to separate this noise from the trend which will lead to the better but more complex time series models $\hat{x}_t^c$. But first, r_t^s is analyzed in more detail.

3.3.3.2. Residual analysis—simple model

A closer look at the residual component r_t' reveals further insights in the observations x_t. For this closer look, the ACF and the PACF from Section 2.3.5.2 are used. But first, the distribution of the residuals is examined (see Figure 3.13).

The distribution of the residuals $P(r_t^s)$ can be found in Figure 3.13a. Additionally, a Q-Q plot of residuals of the simple model is depicted in Figure 3.13b. Further statistics of the r_t^s are in Table 3.4.

Clearly, the residuals do not show white noise behavior (see in Section 2.3.5.4), because the non-zero mean ($\bar{r}_t = 0.053$). A t-Test for zero mean resulted in $t > 9$ and a p-value of $p = 3 \cdot 10^{-19}$. The



(a) Residual distribution $P(r_t^s)$

(b) Q-Q plot of r_t^s

(c) AutoCorrelation Function (ACF)

(d) Partial AutoCorrelation Function (PACF)

Figure 3.13.: Residual analysis plots of r_t^s

skewness γ_1 of r_t^s is

$$\gamma_1 = E\left[\left(\frac{r_t^s - \bar{r}_t}{\sigma_r}\right)^3\right] = 1.14 \quad , \tag{3.3}$$

hence, the mass of the distribution is concentrated on the left—the distribution is right-skewed. This results from the days with higher demand. Additionally, r_t^s possesses heavy tails. This can be calculated through the kurtosis γ_2 in Fisher's definition

$$\gamma_2 = E\left[\left(\frac{r_t^s - \bar{r}_t}{\sigma_r}\right)^4\right] - 3.0 \quad . \tag{3.4}$$

Positive values indicate heavy tails. For r_t^s the kurtosis results in $\gamma_2 = 2.23$. The distribution of r_t^s is right-skewed and possesses heavy tails (see Figure 3.13b).

Dependencies exist between current x_t and former observations x_{t-i}, resulting in a non-white noise behavior. This can be seen in the ACF depicted in Figure 3.13c. Additionally, the PACF is shown in Figure 3.13d. The ACF and PACF clearly show that autocorrelations exist in the time series. In the next section a method is applied removing these correlations.

3.3.3.3. ARMA modeling and GoF

The Box-Jenkins method is applied as described in (Box et al. 2015) to model the residual's dependency r_t^s on previous values m_t. The goal is to ensure that the remaining part r_t is a white noise process and, thus, a real random variable without any dependencies to be further modeled

$$r_t^c = r_t^s - m_t \quad . \tag{3.5}$$

m_t is chosen to be an ARMA model (see Section 2.3.5.5) since it can be concluded from ACF and PACF in Figure 3.13 that the residuals are not fully describable by a single AR nor a single MA process (according to Box et al. 2015). The ARMA model is implemented in the state space form described in Section 2.3.5.6 using Python's statsmodels package for enabling the computational highly efficient Kalman filter formulation later on to forecast future time series values.

Subsequently, the total deterministic behavior of the time series—the resulting complex time series model $\hat{x}_t^c$ is described by

$$\hat{x}_t^c = m_t + \hat{x}_t^s = m_t + \bar{x}_t + s_t^1 + s_t^2 \quad . \tag{3.6}$$

Table 3.2 presents the GoF statistics for ARMA models fitted on r_t^s. It has to be noted that models with $p = 0$ represent pure MA models whereas models with $q = 0$ are plain AR models. Hence, the GoF statistics in the table also assist in deciding if a mixed model is necessary for describing the autocorrelations in the residuals. Both, the AIC statistics in Table 3.3a as well as the BIC statistics in Table 3.3b, show minimal values for the ARMA(2,1) model

$$x_t = c + \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + w_t + \theta_1 w_{t-1} \quad . \tag{3.7}$$

Hence, this model is chosen to describe the behavior of r_t^s.

Table 3.2.: GOF results for ARMA(p,q) model fits of r_t^s

	(a) AIC				(b) BIC			
p/q	0	1	2	3	0	1	2	3
0	-	-174.95	-603.69	-756.54	-	-163.16	-586.00	-732.96
1	-1108.13	-1474.69	-1545.73	-1548.12	-1096.34	-1457.00	-1522.15	-1518.63
2	-1287.18	**-1549.78**	-1547.90	-1546.19	-1269.49	**-1526.19**	-1518.41	-1510.81
3	-1384.09	-1547.91	-1546.74	-1544.24	-1360.50	-1518.42	-1511.36	-1502.97

The fitting coefficients in equation (3.7) result in $c = (0.00 \pm 0.05)$, $\varphi_1 = (1.23 \pm 0.02)$, $\varphi_2 = (-0.25 \pm 0.02)$ and $\theta_1 = (-0.81 \pm 0.02)$ and the w_t values have a standard deviation of $\sigma = (0.033 \pm 0.001)$.

The constant c is zero due to the fact that the mean value $\bar{x}_t$ of the time series is already part of the model since equation (2.105). The high φ_1 coefficient indicates a strong dependency of the time series on previous values, which can be already seen in the high ACF in Figure 3.13. The negative θ_1 coefficient indicates that the series is smoother than a pure white noise process. Figure 3.14 shows the results of the ARMA(2,1) fit. Consequently, m_t is smoother than r_t^s since the random component r_t^c is extracted from it. Furthermore, r_t^c shows the desired white noise behavior.

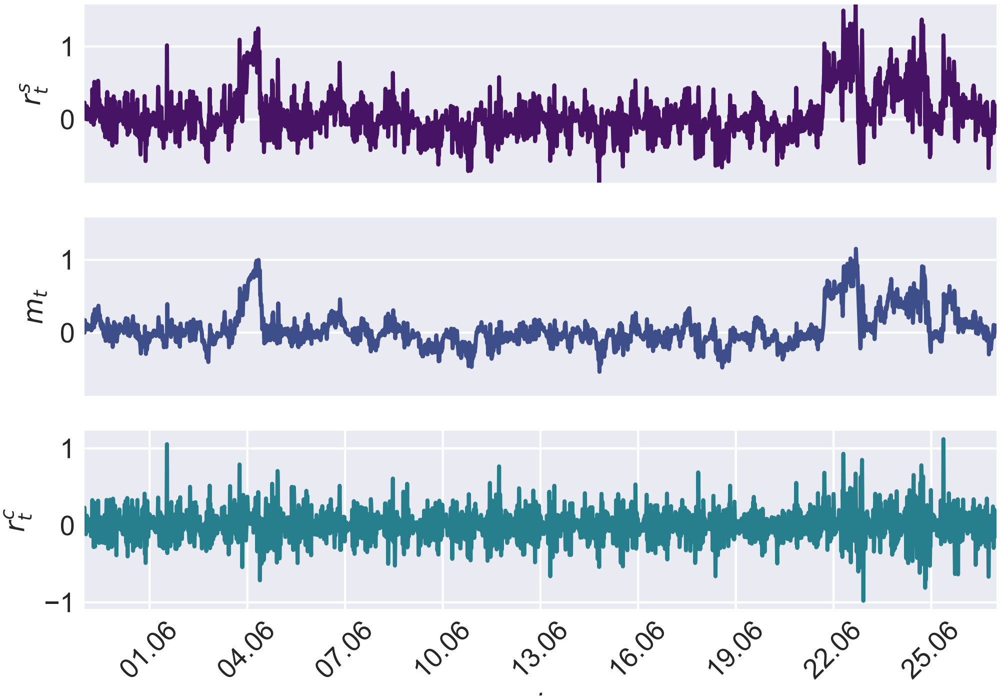


Figure 3.14.: Results of the ARMA$(2,1)$ fit on r_t^s

Assembling the complete model together—as described in equation (3.6)—leads to $\hat{x}_t^c$, which is shown in Figure 3.15. The comparison of x_t with $\hat{x}_t^c$ and $\hat{x}_t^s$ for the last week of the training phase shows that the model $\hat{x}_t^c$ fits the observations x_t and certainly follows the measurements better than $\hat{x}_t^s$.

3.3.3.4. Residual analysis—complex model

In fact, the autocorrelations in r_t^c of the complex model are now completely removed, as can be seen in Figure 3.16. The ACF in Figure 3.16c as well as the PACF in Figure 3.16c of r_t do not show any significant lags besides $h = 0$, thus, the residuals r_t^s are independent of former function values.

The residuals follow now a white noise distribution with a mean of $\overline{r}_t^c = 0.004$, small skewness of $\gamma_1 = 0.18$, although, it still possesses heavy tails with a kurtosis of $\gamma_2 = 2.48$ (see Figures 3.16a and 3.16b). At least there is a probability of 30 % retrieved from t-Test statistics that the true mean of $P(r_t^c)$ is zero. For forecasting purposes later on this will be sufficient. Comparing the residual analysis plots of r_t^s in Figure 3.13 with r_t^c in Figure 3.16 shows that the time series is modeled better than using the simple model. Additionally, some statistical properties of both residual distributions can be found in Table 3.4.

3.3.3.5. Comment on higher time resolution and pressure sensors

Higher time resolutions of the measurement data result in higher ARMA model orders. This is reasonable, because if the distance between subsequent time stamps decreases, the temporal influence is

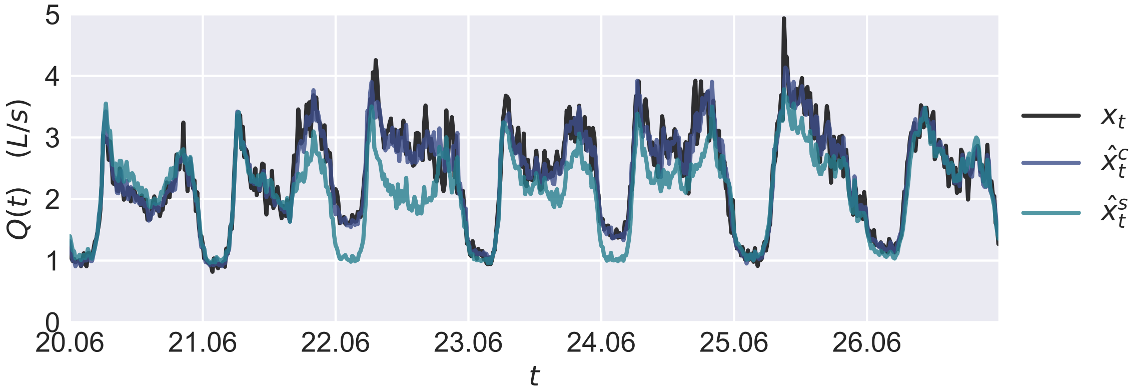


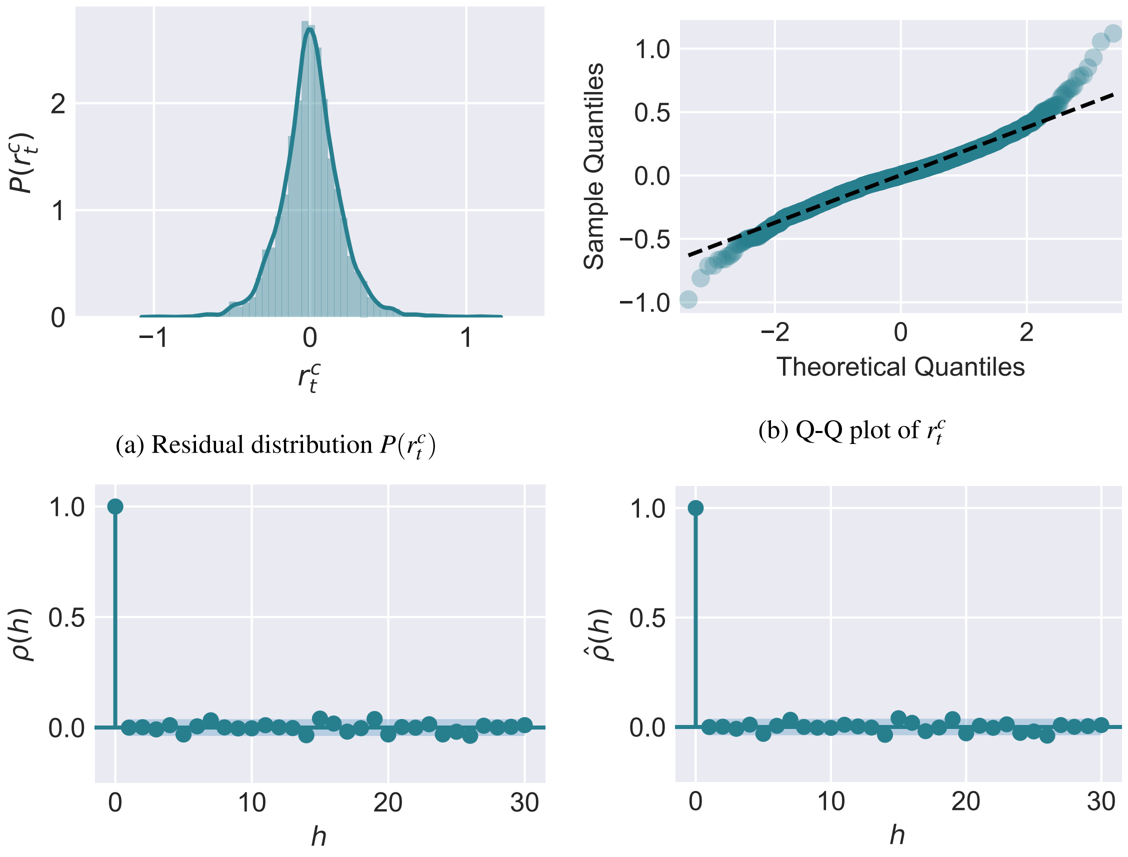
Figure 3.15.: Comparison of x_t versus the model fits $\hat{x}_t^s$ and $\hat{x}_t^c$

Table 3.4.: Statistics of the residuals r_t of the simple s and complex c time series model

	r_t^s	r_t^c
N	2688	2688
$\bar{x}_t$	0.053	0.004
σ_t	0.305	0.188
min	-0.862	-0.978
25%	-0.120	-0.102
$\tilde{x}_t$	0.002	0.003
75%	0.162	0.106
max	1.576	1.120

noticeable over more lags. For example, applying the same procedure as in Section 3.3.3.3 on measurements with higher time granularity of five minutes, the best ARMA model with the lowest AIC and BIC values is ARMA(2,2). This effect works also in the opposite direction. Lower time resolutions of one hour result in an ARMA(1,1) model to be optimal.

The ARMA modeling of the pressure measurements seem to be less influenced by past measurement values. The optimal ARMA models in dependency of the time for the different installed sensors can be found in Table 3.5. For the 15 minute time period, the ideal model is identified as an ARMA(1,1) model for three out of five sensors. Only HG3933 with ARMA(2,1) and HG4162 with ARMA(1,3) differ. However, the AIC and BIC values of the corresponding ARMA(1,1) models are very close to the ideal one. Anyway, for a time resolution of one hour, the ARMA(1,1) model is optimal for all sensors. Furthermore, for the highest time resolution of five minutes, the ARMA(1,1) model is identified to be the best for two sensors. Again, the AIC and BIC values are very close between the other model orders and the ARMA(1,1) model. Consequently, for all following simulations, an ARMA(1,1) is chosen for simplicity to describe the pressure measurements independent of the time resolution. For the inflow measurements, the ARMA order as stated in Table 3.5 is used, because the flow seems to be more sensitive to the modeling order. Nevertheless, the forecast performance for the different models do not significantly depend on the ARMA order.



(a) Residual distribution $P(r_t^c)$

(b) Q-Q plot of r_t^c

(c) AutoCorrelation Function (ACF)

(d) Partial AutoCorrelation Function (PACF)

Figure 3.16.: Residual analysis plots of r_t^c

Table 3.5.: Optimal ARMA model orders p/q for different pressure sensors and different time resolutions

Sensor	[1 H]	[15 Min]	[5 Min]
Inflow	1/1	2/1	2/2
HG3835	1/1	1/1	1/2
HG3933	1/1	2/1	2/2
HG4162	1/1	1/3	1/1
HG4215	1/1	1/1	1/1
HG4383	1/1	1/1	1/2

3.3.4. Forecast performance analysis

Fitting models on measurement data is one task, but the interesting question is if the model can estimate future measurement values—and if, how accurate these predictions are. This section deals with this by testing the two models—the simple seasonal $\hat{x}_t^s$ and the complex ARMA model $\hat{x}_t^c$—on their ability to predict the near future. If future states of the WDS are predicted with a high accuracy, it

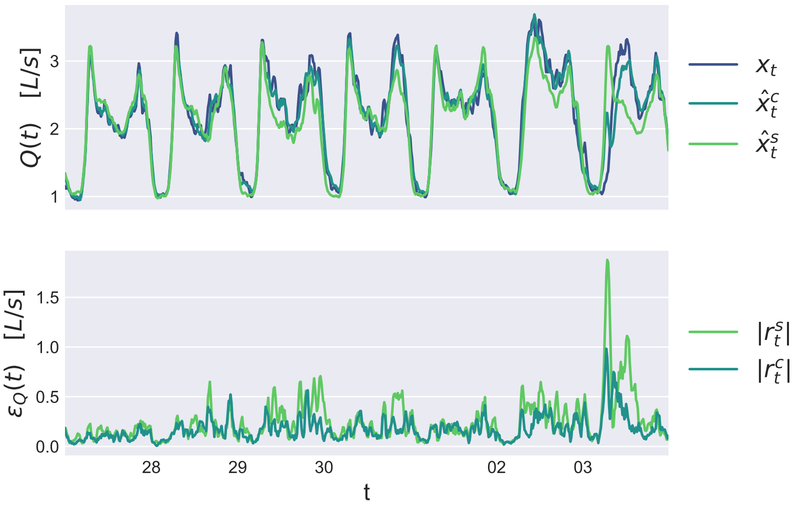


Figure 3.17.: Forecast analysis plot for the inflow

is also possible to tell, if the future diverges from the past and to find reasons for that. This will make the detection of faults in the system possible.

The training phase in Figure 3.2 is already used for analyzing the measurements and building respectively fitting time series models on top of this data. That is why the second phase—the test phase—is used now for the forecast performance evaluations. The forecast of $\hat{x}_t^s$ and $\hat{x}_t^c$ is evaluated with the criteria discussed in Section 2.3.5.7—MAE, RMSE, MAPE and finally the forecasting skill SS to compare the models directly.

$\hat{x}_t^s$ is fitted on new data once every week. Then this data will be used to forecast the values for the next week. When the next week ends, the data is again used to fit the model. Then the whole procedure is repeated.

$\hat{x}_t^c$ is assumed to use the weekly forecast data from $\hat{x}_t^s$ and build an ARMA model on top of the residuals r_t^s to extract correlation effects (Section 3.3.3.3). The ARMA model is fitted on the training data, not on the data in the testing phase. Subsequently, the ARMA model is used to generate one-point forecasts. In other words, the model produces—at each timestep—a 15 minutes future prediction and compares this to new values. The values used in the forecast performance evaluation are these values. Subsequently, the ARMA model takes the current measurement and incorporates it into the model parameters through the Kalman filter approach discussed in Section 2.3.5.6. Then it produces the next one-point forecast based on this new data. Thus, it is capable of handling and incorporating the data in real-time. Numerically, it is highly efficient. The fitting of the model on the training data ($\approx$ 3000 data points) takes less than 1.5 seconds. This has to be done once. Subsequently, the forecast for the next data point takes just a tiny fraction of a second.

The forecast of the time series is tested on both—flow and pressure data. Figure 3.17 shows the forecast for the flow measurement data compared to the original time series x_t. It can be seen that both models are capable of representing the unknown data very well. However, the complex model is superior to the simple model. This can also be seen in the statistical parameters in Table 3.6. Interestingly—looking at the MAPE value—the simple model is already capable of describing nearly 89 % of the time series correctly (100 - MAPE). This value is furthermore increased by the ARMA

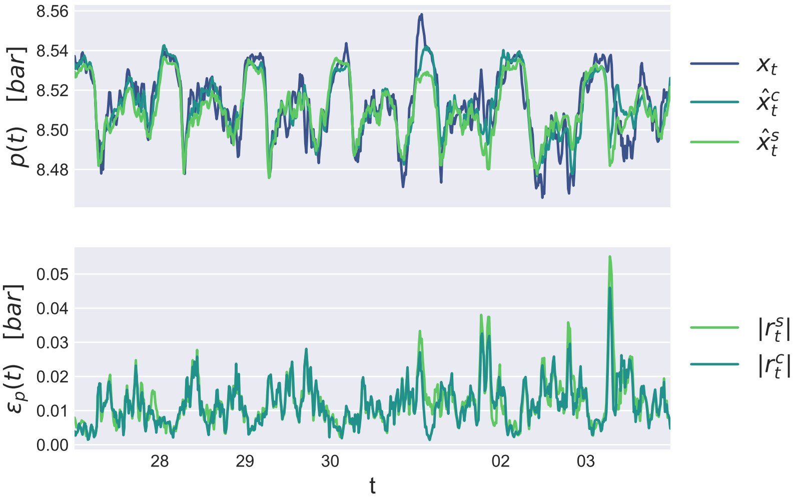


Figure 3.18.: Forecast analysis plot for pressure sensor HG3835

model of up to 92 %. Especially, the last day in the forecast shows a different behavior to the previous four weeks. However, the ARMA model is capable of detecting this different behavior and adapts to it.

Concerning the forecast performance, the pressure measurements show similar behavior to the flow measurements. Again, the complex model performs better than the simple one (Figure 3.18 and Table 3.6). The forecast skill measure reveals the same picture—SS= 0.57 for flow and SS= 0.15 for pressure.

The pressure measurements are very well described by both models, because demand fluctuations are not affecting them as strong compared to flow measurements.

Table 3.6.: Forecast metrics evaluated for $\hat{x}_t^s$ and $\hat{x}_t^c$ on the data of the test phase

	Flow		Pressure	
	$\hat{x}_t^s$	$\hat{x}_t^c$	$\hat{x}_t^s$	$\hat{x}_t^c$
MAE	0.25	0.14	0.012	0.012
RMSE	0.36	0.24	0.018	0.017
MAPE	11.33	7.84	0.016	0.015

3.3.5. Leak detection

Leak detection algorithms have the advantage that the procedure of finding abnormalities in the measurement data can be automatized. Thus, expensive human surveillance is not needed anymore. Nowadays, this is of special importance since the amount of data in WDSs grows exponentially.

If algorithms assume a leak in the system, notifications containing relevant information can be send to decision makers. Subsequently, the decision maker can decide on strategies to handle the extraordinary situation in the system. Additionally, automatic reports can be generated for decision makers on a regular basis.

One of the biggest challenges in automatic leak detection is to distinguish leaks from big customers. Both extract water from the system and thus influence the mass balance in the system. They only differ in the fact that the customer follows a certain pattern and usually stops its consumption after a while. A leak, in contrary, will extract water until it is repaired[22]. It is recommended to measure large customers and customers with different water usage behaviors—water consuming industries in a primary residential zone or hospitals, for instance—in a system with extra demand measurement devices. That devices are ideally smart water meters—able to send the demand data automatically to the WU at near real-time.

Leak detection with hydraulic sensors has the big advantage that it is cheap—because a low number of sensors is needed compared to, for example, acoustic noise logging or transient leak detection—and enables fast responses to leaks compared to other leak detection methods. The physical principles behind hydraulic leak detection are the mass and energy conservation laws. When a leak occurs, flow upstream of the leak increases due to the conservation of mass. This enlarged upstream flow leads to a pressure drop downstream of the leak due to the higher friction losses (energy conservation). Downstream of the leak, the flow stays the same according to mass balance equation. Additionally, the pressure gradient downstream of the leak also stays the same as in the leak free scenario. Comparing the flows and pressures to measured values of the leak free system (values recorded in the past before the leak occurred) enables conclusions to be drawn[23] if a leak developed in the system.

In this section, we make use of plain statistical models describing past measurements as described in the previous sections. These models are compared to new measurements to detect leaks as discrepancies between the forecasted values of the time series model and the current measurements. For that reason, only data-driven time series models and not complicated hydraulic simulations are necessary.

For testing leak detection algorithms, an artificial leak was introduced in the system through opening a hydrant. The leak was opened on 4^{th} of June at 12:20 at hydrant HG3880. The leak outflow was set to meet approximately $Q_L \approx 0.7\ (L/s)$, which is roughly a third of the system's mean inflow (see Figure 3.19). Although, the occurrence of leak can clearly be seen by human eyes in this case, this section deals with developing mathematical methods to provide computers with the same skill. Furthermore, it has to be noted that this is a very small leak compared to leakage outflows in literature—usually a magnitude higher (> 5 L/s). Especially, detecting leaks with pressure sensors is harder for small leak sizes since friction losses are a function of the power of the velocity, e.g. velocity squared (see equation (2.53)).

3.3.5.1. Minimum night flow Tukey

Tukey's test (described for outliers in Section 2.3.3) can be used for detecting leaks as well. Applying Tukey's test on the data represented in Figure 3.19 during the MNF results in Figure 3.20. The leak can be detected in the measured inflow in the system with this simple test. Although, the response time to the leak is in the order of several days. If the test detects an outlier on one night, it might just be due to higher consumption in the system. If the test instead leads to outliers on several subsequent nights, a leak can be assumed in the system and counter measures can be taken.

It has to be noted that every new measurement containing an outlier for the Tukey test also elevates the outlier threshold barriers. Comparison of Figure 3.9 with Figures 3.20 shows that the upper boundary

[22] ...or the system is not pressurized anymore.

[23] Conclusions can also be drawn on the leaks position in the system. See Chapter 5 for a detailed description.

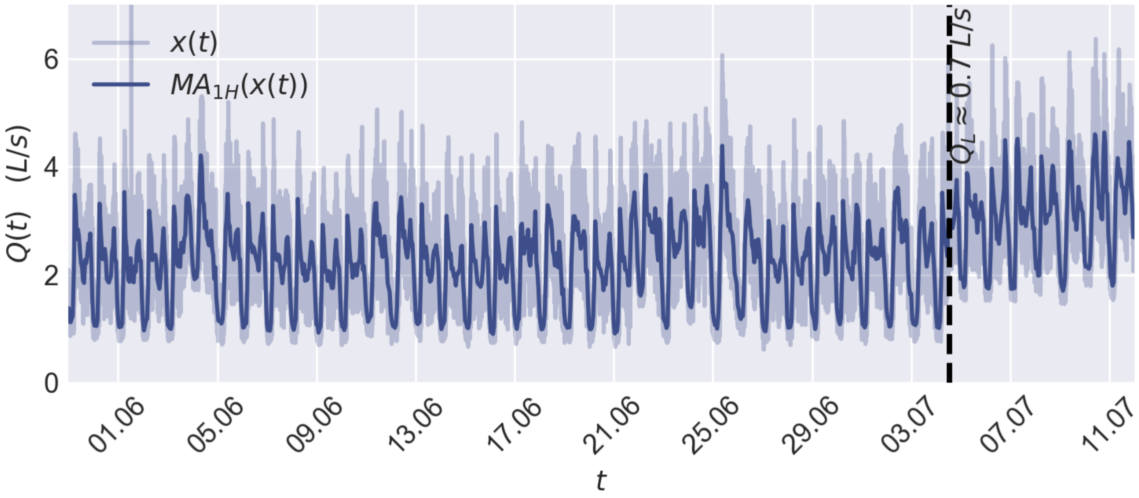


Figure 3.19.: Leak occurring at 2016-07-04 at 12:20 with a magnitude of $Q_L \approx 0.7L/s$

for the outlier test grew—outliers that were found in Figure 3.9 are not outliers anymore in Figure 3.20 (See the 24th of June for example). Longer leak run-times result in higher probabilities that outliers will be overshadowed by the leak. Especially, leaks that start small and grow over time are very difficult to detect with Tukey'a outlier test.

Pressure measurements provide a cheap alternative to flow measurements for fault detection in WDS. Hence, Tukey's outlier test has been applied to pressure measurement data measured at the hydrants in the system as well. The results for Tukey's test on MNF pressure data can be found in Figure 3.21. Obviously, the pressure differences between the leak data and the leak free system are not big enough to trigger alarms in the outlier test. Hence, this method fails in predicting leakages on pressure measurements. It has to be noted that the result is the same for all pressure sensors in the system. Other methods have to be used to reveal faulty states in the system in pressure measurement data, as described in the next sections.

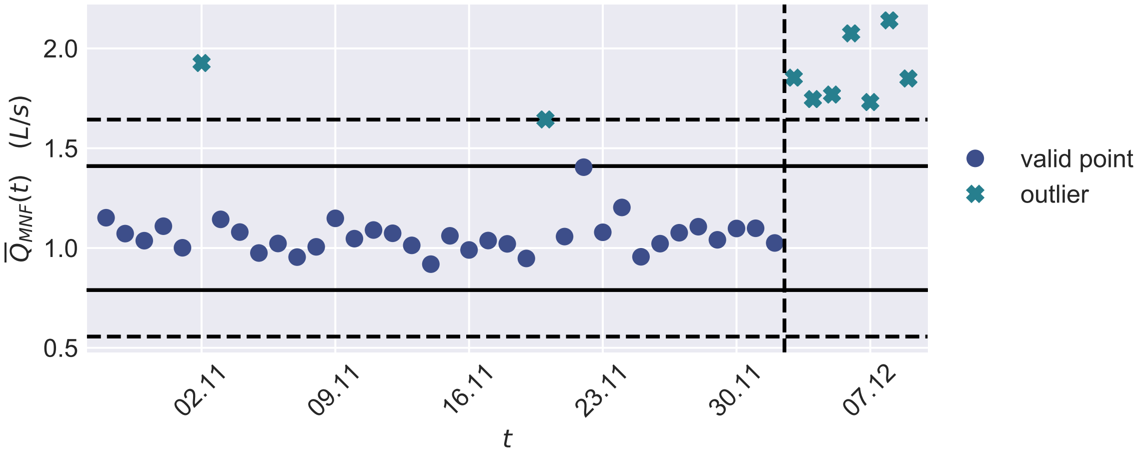


Figure 3.20.: Tukey test for leak detection

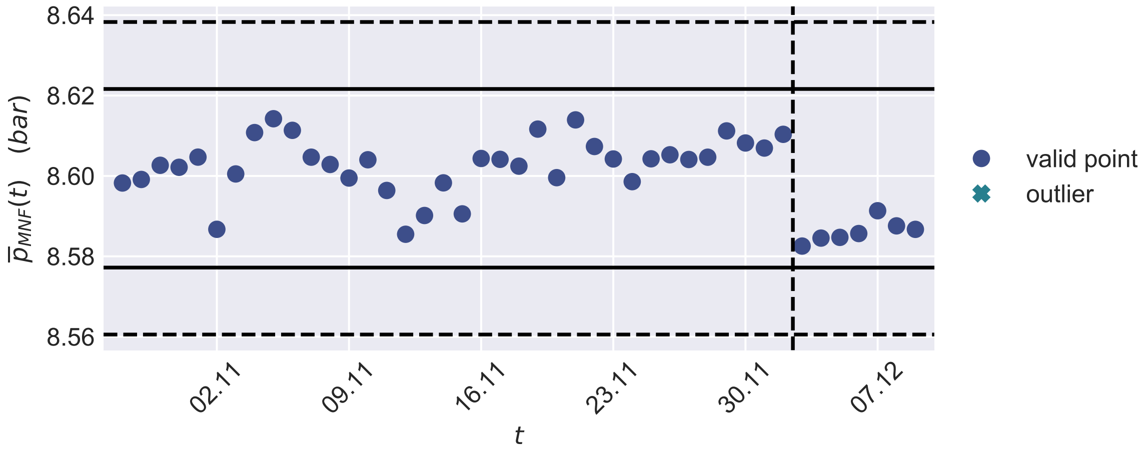


Figure 3.21.: Tukey test for leak detection with pressure sensor HG3835

3.3.5.2. CUSUM

The CUSUM algorithm (described in Section 2.3.6.1) is applied on m_t of the flow and pressure measurements. The implementation is an extension to the code of Duarte 2015. For the flow measurements, only the positive values g_t^+ are used, since a leak leads to an increase in the inflow. For the pressure measurements, g_t^- is used to detect anomalies. The threshold values τ and the drift values d have to be chosen once for the CUSUM algorithm based on historic measurements. They do not change over time. The training data serves as these historic measurements. Since the scales of flow and pressure measurements are different, particular values are used for flow and pressure data. The values are chosen in such a way, that no false alarm is triggered in the historic data. The flow threshold is $\tau = 0.55$ with a drift value of $d = 0.02$. The pressure threshold is $\tau = 0.0358$ with a drift value of $d = 10^{-4}$.

The results of the CUSUM test applied on the flow measurements can be found in Figure 3.22. The blue line at the top of the figure shows the m_t time series resulting from a one-point forecast—a prediction for the next 15 minutes updated each time when a new measurement value arrives. In that way, the CUSUM algorithm works as a real-time and online test, since only the past measurements are involved in computing $g_t^\pm$ and no future data is needed to evaluate the current value. It has to be noted that the computation of $g_t^\pm$ is instantaneously, since it solely consists of a summation of two values and an if clause. Obtaining the forecasted value of m_t is computationally highly efficient through the state space implementation of the ARMA model and takes only few milliseconds for all flow and pressure measurements. This is why this test enables an online leak detection. The black points in Figure 3.22 represent the time instance when an alarm is triggered. Once an alarm is triggered, the algorithm additionally allows to compute, when the event started. This is depicted as a green triangle with the tip showing to the right. Also, the estimated event end is computed as a green triangle pointing to the left. The bottom of the figure shows the computed $g_t^\pm$ values of the time series and the threshold depicted as black line.

Figure 3.22 shows that high demands at the 4.6 and around the 22.6 do not trigger an alarm. Only an event occurring in the morning before the artificial leak—caused possibly by a hydrant flushing due to repair work on a leaking hydrant—leads to a false alarm on the 3.7. at 08:00. The algorithm estimated this event to start on the 3.7. at 06:45 and to end on the 3.7. at 09:00. The first correct leak alarm is triggered at the 4.7 at 14:15, approximately two hours after the leak is produced in the system. The

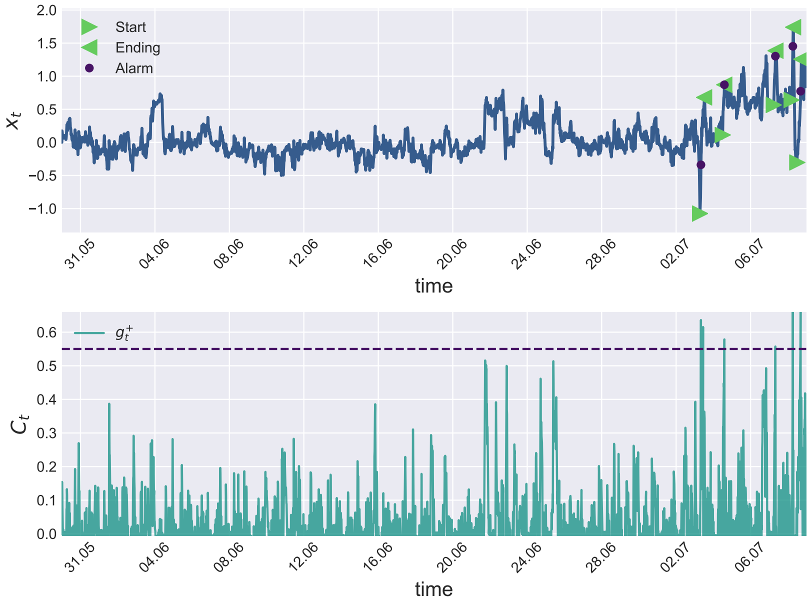


Figure 3.22.: CUSUM test on m_t of the inflow data. A leak alarm is triggered on the 3.7. 08:00 (false alarm) and on the 4.7. 14:15—1 hour and 55 minutes after the leak occurred

algorithm estimates that the event started at 12:00, which is very close to the actual leak start at 12:20. After this first event, a lot of alarms are triggered subsequently, because of the change caused by the leak, further indicating that something has happened in the system.

An example of the application of the CUSUM test on the pressure measurements can be found in Figure 3.23. A leak alarm is initiated at 15:00—also very close to the introduction of the real leak. A summary of the application of the CUSUM method on all sensors can be found in Table 3.7. No false positives occurred in the pressure measurements and four out of five sensors were able to detect the leak at the same day it happened. Only HG4162—which is the farthermost sensor—was not able to detect the leak with the CUSUM method in time—causing a detection time delay of almost three days.

Table 3.7.: CUSUM test results applied on m_t for all sensors showing the time when a leak alarm is triggered first with estimated event start and end times and preliminary false positive (FP) alarms and the detection delay in hours.

Sensor	Alarm	Start	End	FP	Delay
Inflow	4.7. 14:15	4.7. 12:00	4.7. 14:15	1	1.9
HG3835	4.7. 15:00	3.7. 07:00	6.7. 20:00	0	2.7
HG3933	4.7. 20:30	3.7. 11:00	7.7. 08:45	0	8.2
HG4162	7.7. 08:30	3.7. 07:30	7.7. 08:30	0	68.2
HG4383	4.7. 15:00	3.7. 07:00	7.7. 09:15	0	2.7

The CUSUM algorithm was also applied on r_t^s of the simple model $\hat{x}_t^s$. However, better results with less false alarms were obtained using m_t of the complex time series model $\hat{x}_t^s$.

Obviously, the advantage of the CUSUM method is a very fast and timely detection of leaks—also

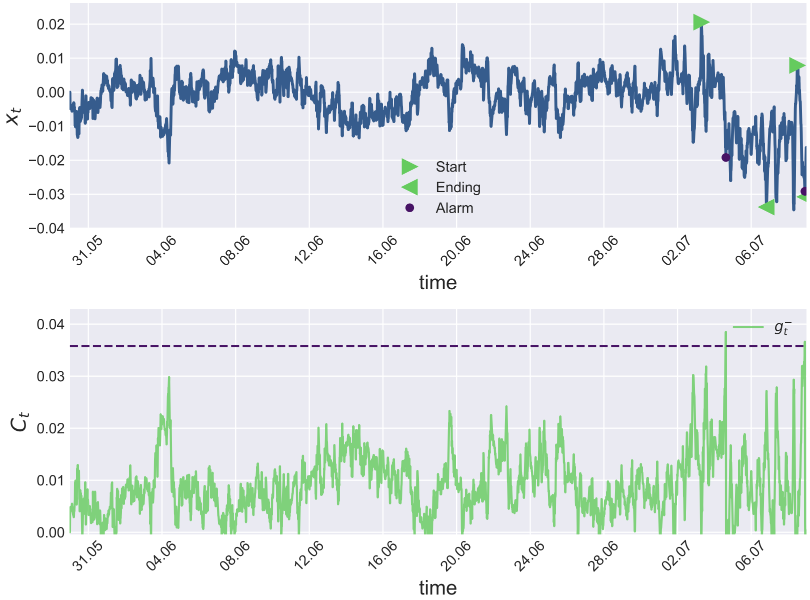


Figure 3.23.: CUSUM test on m_t of pressure sensor HG3835. A leak alarm is triggered on the 4.7. 15:00—2 hours and 40 minutes after the leak was introduced in the system

possible with pressure sensors. Furthermore, the algorithm estimates approximate times of leak. The disadvantages are the parameters settings τ and d. These parameters have to be found for the measurements based on historic data and are very sensitive—even to small changes. Additionally, the estimated beginning of the leak event somewhere between the proposed start time and the triggered alarm are imprecise. For instance, an accurate estimation of the leak time is very important for water balance calculations.

3.3.5.3. Likelihood-ratio

Another method of detecting leaks—not as sensitive on threshold values as the CUSUM algorithm—is the likelihood ratio $\mathcal{G}$ as described in Section 2.3.6.2. To make the computation more efficient, the likelihood-ratio equation (2.150) is not computed over the whole time series, but over the past week of the actual measurement value. The computation takes 0.03 seconds for an actual measurement, thus, this test can also be applied online. Additionally, this method allows to estimate the size of the leak respectively the pressure drop caused by the leak. This information is very important for computing the location of the leak later on in Chapter 5. The size of the change is estimated through computing the differences of the means after and before the estimated change-point $\hat{\tau}$

$$\Delta\mu(\hat{\tau}) = \underbrace{\frac{1}{(N-\hat{\tau}-1)}\sum_{j=\hat{\tau}+1}^{N} x_j}_{\mu_2} - \underbrace{\frac{1}{\hat{\tau}}\sum_{i=1}^{\hat{\tau}} x_i}_{\mu_1} \quad . \tag{3.8}$$

The Likelihood-ratio method is applied on all sensors, the inflow and the four pressure sensors, using the simple $\hat{x}_t^s$ as well as the complex model $\hat{x}_t^c$. For the simple model, the method is applied on the residuals r_t^s—for the complex model, the method is applied on m_t. Different threshold values g_{τ^*} are defined for the different models and sensors. The results and the corresponding threshold values can be found in Table 3.8.

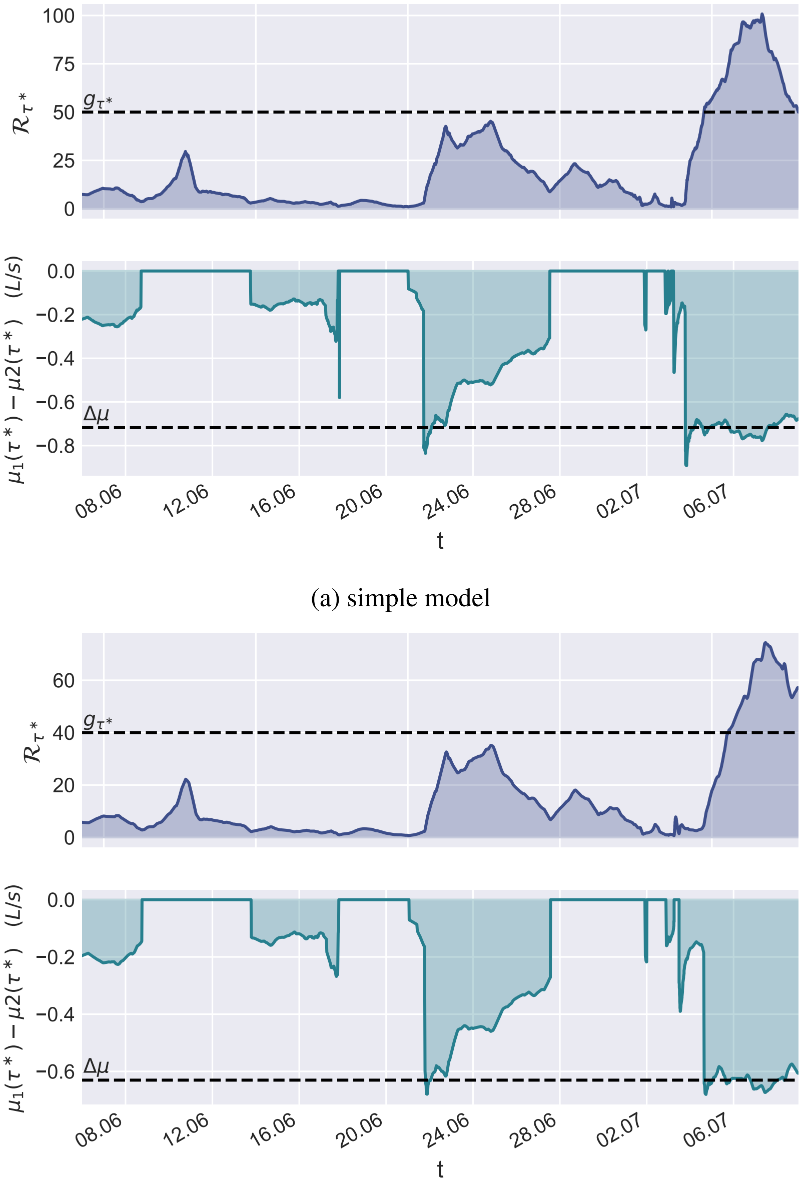
Table 3.8.: Results for the likelihood-ratio leak detection method applied on the simple and the complex model, the corresponding threshold values, the time when an alarm is triggered, the estimated change in the measured variables and the detection delay in hours.

Sensor	Model	Threshold	Alarm	$\Delta\mu$	Delay
Inflow	$\hat{x}_t^s$	$g_{\tau^*} > 50$	4.7. 15:15	0.7178	2.9
	$\hat{x}_t^c$	$g_{\tau^*} > 40$	5.7. 18:00	0.6305	29.7
HG3835	$\hat{x}_t^s$	$g_{\tau^*} > 0.04$	4.7. 14:00	-0.0222	1.7
	$\hat{x}_t^c$	$g_{\tau^*} > 0.02$	5.7. 13:00	-0.0173	24.7
HG3933	$\hat{x}_t^s$	$g_{\tau^*} > 0.10$	5.7. 21:00	-0.0240	32.6
	$\hat{x}_t^c$	$g_{\tau^*} > 0.02$	6.7. 20:00	-0.0111	55.7
HG4162	$\hat{x}_t^s$	$g_{\tau^*} > 0.035$	4.7. 23:15	-0.0161	10.9
	$\hat{x}_t^c$	$g_{\tau^*} > 0.017$	5.7. 23:15	-0.0116	34.9
HG4384	$\hat{x}_t^s$	$g_{\tau^*} > 0.070$	4.7. 23:00	-0.0260	10.7
	$\hat{x}_t^c$	$g_{\tau^*} > 0.045$	6.7. 07:00	-0.0201	42.7

It can be seen that the simple model is faster in detecting the leak. This is due to the smoothing effect of the ARMA model caused by the incorporated dependency on the past values in the complex model. Additionally, the complex model seems to underestimate the flow and pressure changes—also caused by the smoothing effect. A comparison between the simple model and the complex model on the inflow data can be found in Figure 3.24. Every plot shows the likelihood-ratio on top—a measure that a change is likely in the data. The threshold value g_{τ^*} is also depicted in these plots. The estimated changes are depicted at the bottom together with the estimated $\Delta\mu$ values. Note that the flow change is also depicted negative so it corresponds to the pressure measurements in Figure 3.25. In addition, to detect the leak, this method also shows periods of higher demand in the flow as well as the pressure measurements. The water extraction around the 22^{nd} of June is equal to the size of the leak. However, the water withdrawal did not take as long as the leak. Consequently, no alarm is triggered. The leak can be clearly seen as a high probability in all figures. The peak even gets higher and more distinct from the other peaks, if a longer time period is taken into account.

The results of the loglikelihood-ratio test applied on pressure measurements of the simple model are shown in Figure 3.25. The pressure sensor responding quickest to the leak is depicted in Figure 3.25a. A distinct leak peak can be seen in the pressure. The farthermost hydrant HG4162 from the leak is depicted in Figure 3.25b. Using the CUSUM method it was not possible to timely detect a leak with a sensor at this positions. Now, with the likelihood-ratio method, a detection is possible within 12 hours. The latest leak alarm is triggered by sensor HG3933 within 36 hours.

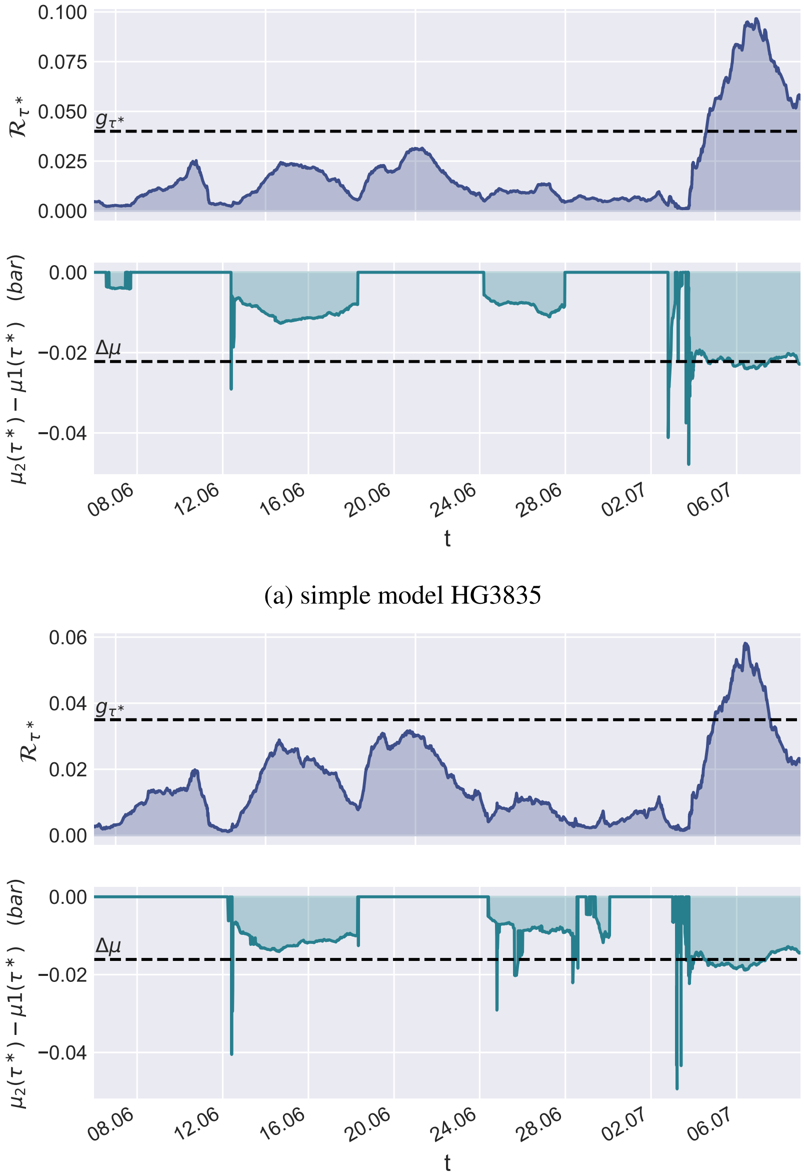
In general, this method is slower in detecting leaks than the CUSUM method, but it is assumed to result in less false alarms. It has to be noted that higher leakage outflows will result in faster leak response times for this method as well as for the CUSUM method.



(a) simple model

(b) complex model

Figure 3.24.: Log-likelihood-ratio test for the inflow measurements



(a) simple model HG3835

(b) simple model HG4162

Figure 3.25.: Log-likelihood-ratio test for different pressure sensors using the simple model

3.3.5.4. Bayes analytical

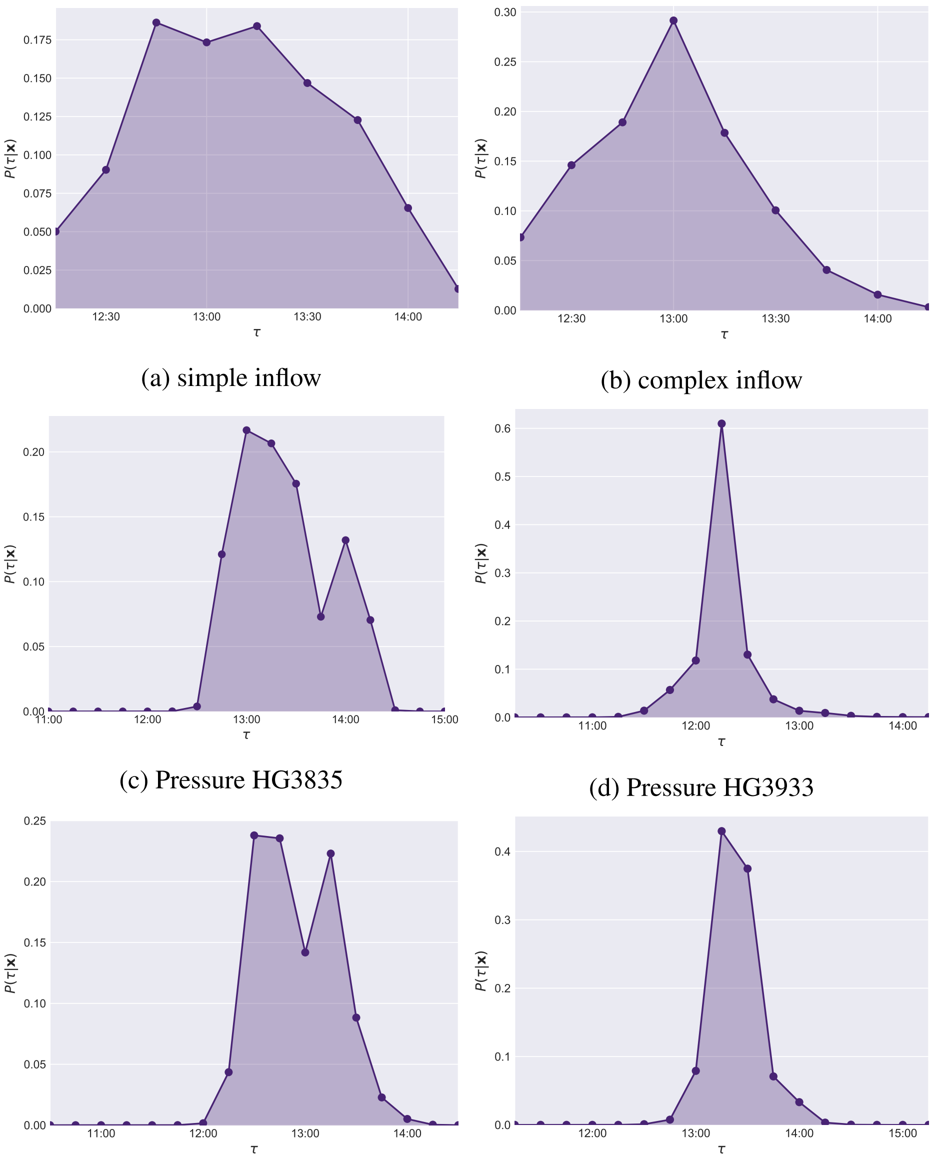
Bayes method (Section 2.3.6.3) can be used to detect leaks, as well. It provides similar results as the likelihood-ratio method from the former section. An additional benefit of this method is that the time when the leak occurred can be isolated with relatively high precision. This is especially important for computing water losses for the water balance, but also for retrieving more accurate pressure estimates before and after the leak occurred by reducing the uncertainty in the leak start. This is useful, for example, for leakage localization after the detection. The time frame for computing the occurrence of the leak is taken from the results of the CUSUM method. The event start τ_s and the event end-time τ_e are used for the better estimation of $\hat{\tau}$. In Figure 3.26, the computed leak time probabilities $P(\tau|\boldsymbol{x})$—resulting from equation (2.158)—are depicted. The maximum value of the probability is taken as leak time estimate $\hat{\tau}$ within the time frame

$$\hat{\tau} = \arg\max_{\tau_s \leq \tau \leq \tau_e} P(\tau|\boldsymbol{x}) \quad . \tag{3.9}$$

Figure 3.26a shows the $P(\tau|\boldsymbol{x})$ for the inflow data using the simple model, whereas Figure 3.26b shows $P(\tau|\boldsymbol{x})$ using the complex model. The maximum is more distinct using the complex model than the simple model, hence, this model is also used for the pressure leak time estimation. $P(\tau|\boldsymbol{x})$ computed with the complex model can be seen in Figures 3.26c to 3.26f for all pressure sensors. All figures show that the leak is estimated around the same time. This is fascinating since every sensor is treated completely independently. Hence, good leak start time estimates are found with every sensor independently of the hydraulic variable that is measured. The leak start time estimates are summarized in Table 3.9. For getting a combined measurement, the mean over all estimates is build on the bottom of the table. Furthermore, it has to be noted that the Bayesian method described in Section 2.3.6.3 leads to a time-shift of the estimates to be actually located between two consecutive time stamps. For that reason, half of the measurement interval has to be subtracted from the estimate, which is 7.5 minutes for a 15 minute interval. Thus, the leak is estimated to have occurred at 12:35—only 15 minutes after the leak actually was introduced. In terms of leak outflow error due to the deviation of leak time estimates, this equals a $\Delta Q_L \approx 600L$ for a leak of 0.7 L/s in the water balance calculation, compared to 5670 L resulting for the uncertainties in the inflow measurement introduced by the CUSUM method.

Table 3.9.: Estimated leak start times $\hat{\tau}$ using the Bayesian method

Sensor	τ^*
Inflow	13:00
HG3835	13:00
HG3933	12:15
HG4162	12:30
HG4383	13:15
$\overline{\tau}^*$	12:42:30



(a) simple inflow

(b) complex inflow

(c) Pressure HG3835

(d) Pressure HG3933

(e) Pressure HG4162

(f) Pressure HG4383

Figure 3.26.: Leakage times estimated by Bayes' theorem

3.3.5.5. Bayes NUTS

This method uses Bayesian statistics as described in Section 2.3.6.3 together with the NUTS-MCMC algorithm described in Section 2.3.2. The algorithm is implemented with the help of Python's PyMC3 package (Salvatier et al. 2016). The huge advantage of this method—and all methods making use of Bayes' theorem in general—is that the algorithms don't need any further information than the measurement data itself. The only assumptions are that a change might have happened in the system and that the measurement data obeys a certain probability function. The method in Section 3.3.5.4 assumes that the data is Gaussian distributed, but other distributions can be used. It has to be noted that for certain probability distributions, finding an analytical solution may be hard or even impossible. Using MCMC methods promise remedy, since any distribution of the measurement values can be used—integration over the state space follows straight numerics. Furthermore, this method is capable of sampling the whole probability function of the problem from the measurement data additionally to $P(\tau|\boldsymbol{x})$, for example, $P(\mu_1|\boldsymbol{x})$, $P(\mu_2|\boldsymbol{x})$ and $P(\sigma|\boldsymbol{x})$. From these distributions, the necessary flow changes and pressure drops for leak localization can be calculated.

To reduce the possible state space and to make the simulations comparable to each other, it is assumed that a leak has happened between τ_s =9:15 and τ_e =15:15 for all simulations. The upper boundary is chosen to be the moment when the second sensor in Table 3.8 discovers a leak using the likelihood-ratio—also a Bayesian method. The cooperation of multiple sensor triggering a leak alarm also reduces the probability of producing false alarms. The lower boundary is set to the last six hours before the alarm.

For first experiments, only the data until 15:15 is used. Thus, only 13 new measurements are taken since the leak occurred (concerning a sample time of 15 minutes). Later, additional experiments with a longer period are made using data until the 8^{th} of June till 12:00. The average flow and pressure values need some time to converge to their real mean values. This can be seen in Figure 3.27, where the difference of the mean value until the current timestamp is subtracted from the mean value after the 10^{th} of June. Huge fluctuations exist in the approximated mean—around the 8^{th} these fluctuations smooth out. The fluctuations are correlated to fluctuations in the other pressure measurements, hence, correlations between the data caused by demand fluctuations in the whole system might be the reason for that. More advanced time series modeling—using correlated time series models over the all measurements—might lead to faster responses and may result in faster and better mean estimates with less fluctuations.

The computation time for the marginalized distributions for Markov chains with 2000 steps is approximately one minute for each sensor. μ_1, μ_2 and σ are sampled with NUTS. Because τ is a discrete variable and thus has not a meaningful gradient, it is sampled from a random uniform integer distribution between τ_s and τ_e, using an adaptive Metropolis step method. The results for the inflow can be found for the simple model in Figure 3.28a, and in Figure 3.28b for the complex model. The median (solid black line) and the mean absolute deviation (dashed black line) are additionally depicted in the figures. The simple as well as the complex model for the short and the long period estimate the leakage occurrence time between 11:45 (simple short and long) respectively 12:00 (complex short and long) with a standard deviation of four to maximum nine minutes. This are very good estimations concerning the short measurement period.

The measurements depicted in Figure 3.28 and 3.29 are generated by simulating 2000 samples, where the first 500 samples are assumed to be the burn in phase and, thus, are neglected. In general, the computed marginal probability distributions are smoother when taking the longer measurement period

Table 3.10.: Flow and pressure difference estimates resulting from Bayes MCMC simulations for the simple and the complex model for the short period till the 4.7. 15:15 and the long period till the 8.7. 12:00

	Short Period		Long Period	
Sensor	$\hat{x}_t^s$	$\hat{x}_t^c$	$\hat{x}_t^s$	$\hat{x}_t^c$
Inflow (L/s)	0.86077	0.68647	0.73929	0.66287
HG3835 (bar)	-0.02019	-0.01314	-0.01850	-0.01476
HG3933 (bar)	-0.02926	-0.00974	-0.02108	-0.00961
HG4162 (bar)	-0.01903	-0.01053	-0.01280	-0.00977
HG4383 (bar)	-0.02563	-0.01268	-0.02207	-0.01724

into account. This can be seen by comparing Figure 3.28a and Figure 3.28c, respectively, Figure 3.28b and 3.28d. Furthermore, the complex model tends to underestimate the differences in pressure and flow before and after the leak as already discussed in Section 3.3.5.3. The median values of the simulation results are summarized in Table 3.10. This results will be used later on for localizing the leak. The results for all pressure sensors using the long period and the simple model can be found in Figure 3.29.

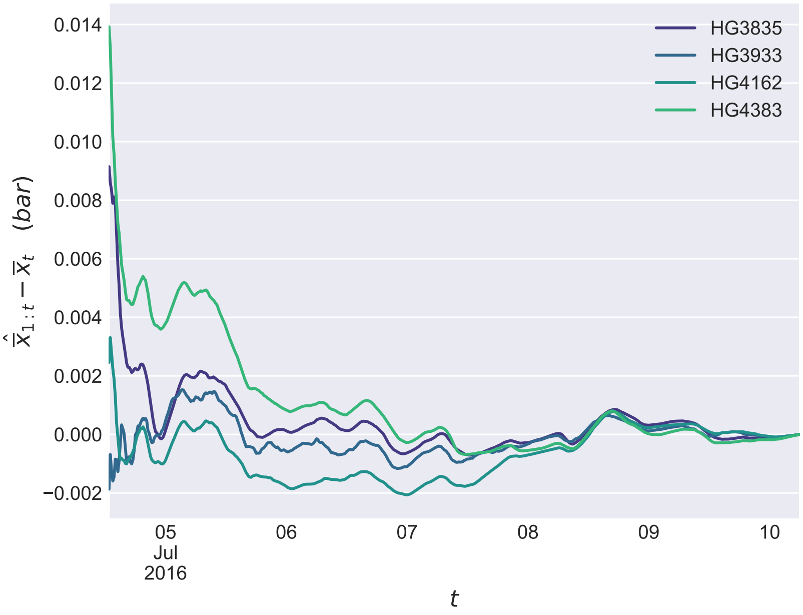
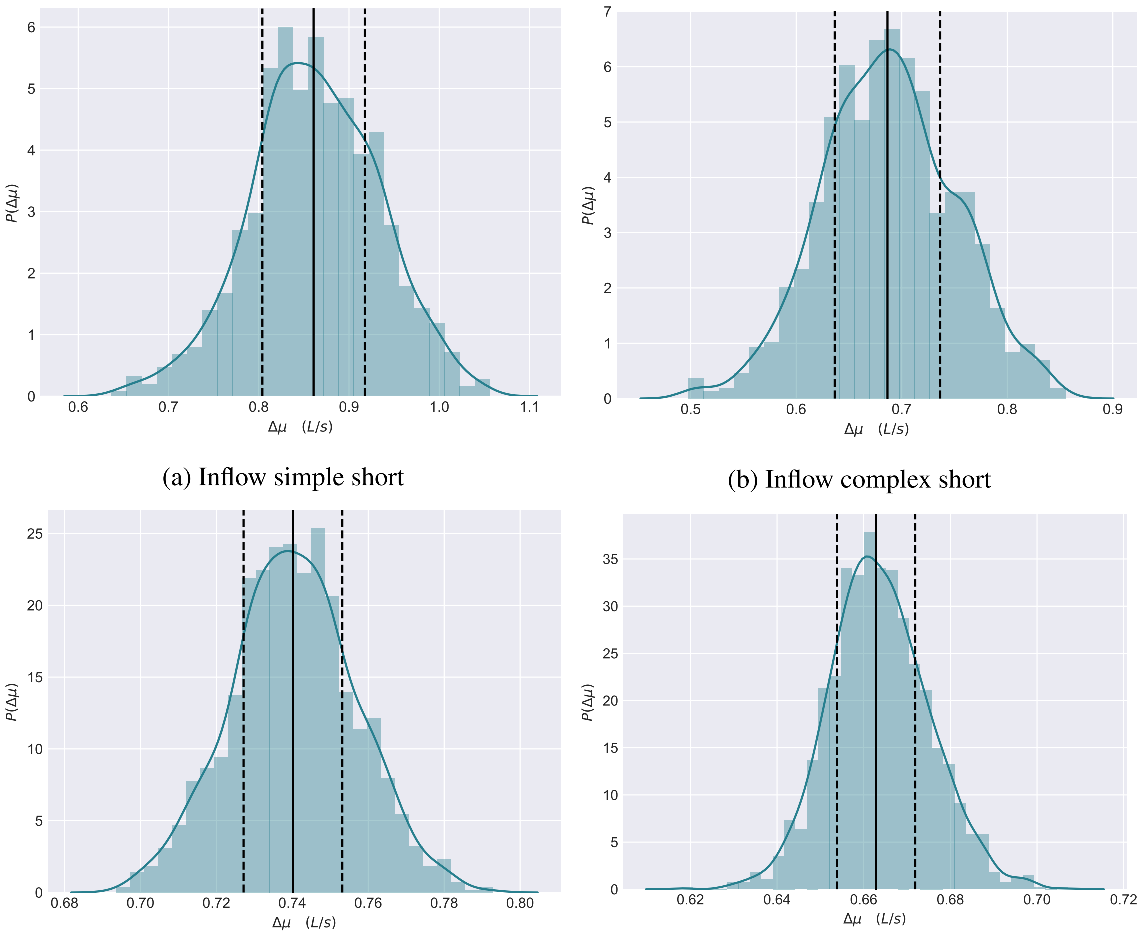


Figure 3.27.: Convergence of the mean values over time

3.4. Conclusion

The overall goal of this chapter was to answer following research question:

Q.1.1 Is it possible to early detect small leaks (<1 L/s) in the real world with pressure sensors in an automatic way?



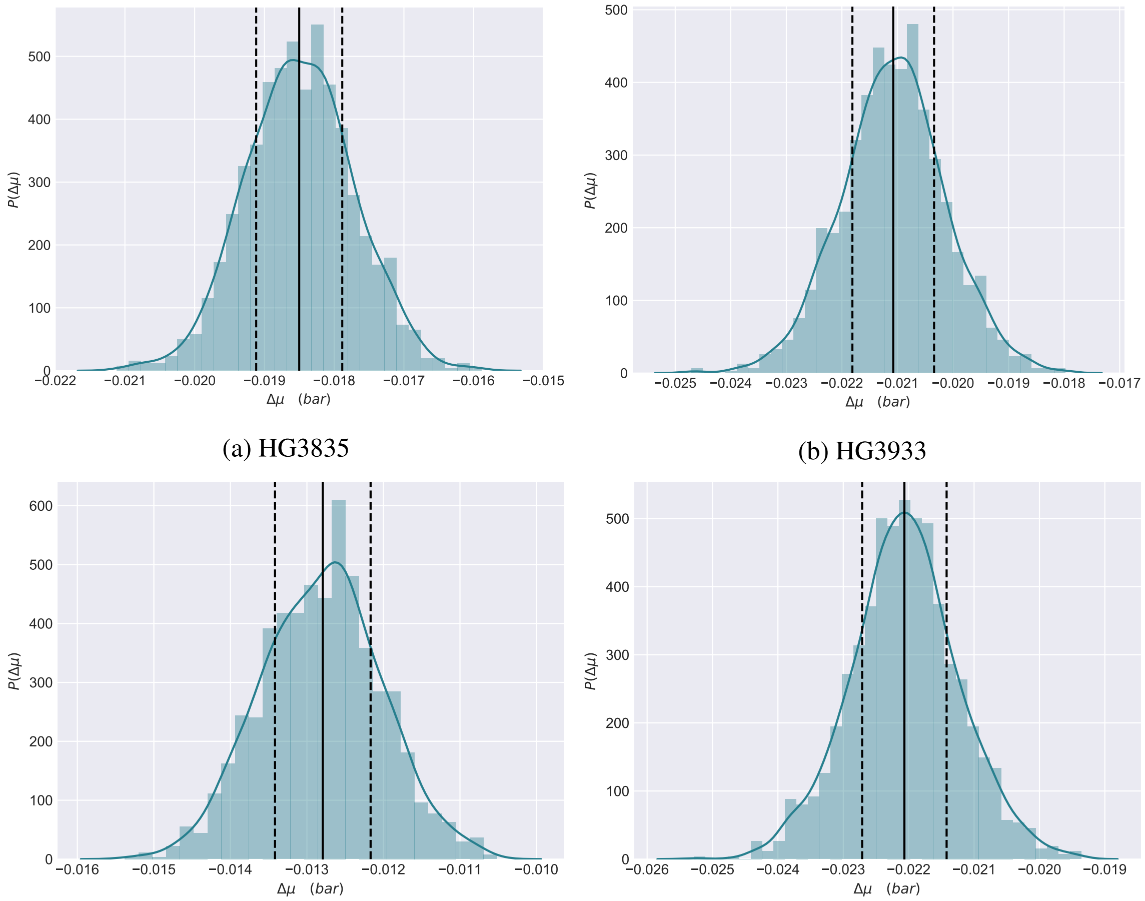
(a) Inflow simple short

(b) Inflow complex short

(c) Inflow simple long

(d) Inflow complex long

Figure 3.28.: NUTS Bayes inference for inflow for simple and complex model and short and long time period



(a) HG3835

(b) HG3933

(c) HG4162

(d) HG4383

Figure 3.29.: Marginalized pressure distributions for the simple model on the long period

This is of high relevance, since pressure sensors cost a fraction of flow sensors, are easier to install, more energy efficient and are capable to measure changes very fast. However, most scientific literature till now stated that flow sensors are superior to pressure sensors and that even pressure information in addition to flow measurements does not lead to any improvements for leak detection at all. This is exaggerated by the fact that most literature deals with the detection of huge pipe bursts. The physical effect, that makes burst detection possible in WDS, is the introduced pressure drop by higher fluid velocities and hence higher friction losses. While this effect can be large for high velocities introduced by large bursts, this effect exponentially decreases with smaller leaks causing lower velocities. Only with advanced statistical techniques, those tiny pressure changes can be detected in the expected noisy WDS environment—which is fraught with measurement noise and uncertainties and hidden under different consumption patterns evolving on at least two time scales.

Furthermore—while removing days with unusual high demand from flow measurement data reduced the fluctuations in the flow measurements—this technique revealed no improvements for pressure measurements. It seemed unimportant to remove these days at all, since the improved time series model was capable to model demand variations very well.

Times series modeling of hydraulic measurement data was introduced in this chapter, starting with a simple model. This model was generated by extracting the two seasonal components from the signal—daily and weekly seasonalities. Surprisingly, this simple model showed already good forecasting performances, capable of describing future data with almost 88 % accuracy. A better model—capable of describing correlations between subsequent data points—was developed using ARMA modeling. Intentionally, no ARIMA model is used (as proposed by Hutton and Kapelan 2015), since the differentiating term in the time series model would extinguish the trend component. For that reason, simple ARMA modeling was chosen. The best ARMA model order was found to be ARMA(2,1) for flow and ARMA(1,1) for pressure data measured with a time granularity of 15 minutes. Increasing the time resolution enlarges also the model order, while decreasing had the opposite effect. By using the more advanced model, the forecast performance increased additional five percent compared to the simple model. The complex model was also able to follow days with extraordinary high demand. Concerning pressure data, the forecast performance of both models was even better than for flow, since pressure does not possess such a high variability. Of course, the low variability is a constraint for leak detection with pressure sensors.

As expected, simple statistical tests like Tukey's fence test applied on MNF data were not able to detect leaks in pressure data. Furthermore, for flow measurements the leak response times were very slow. Hence, more advanced statistical methods were applied in combination with TSA techniques. The CUSUM algorithm was able to detect leaks shortly after they appeared with pressure sensors. The time series model already filtered the seasonal effects and random noise from the measurement data, making timely detection in pressure signals possible. Compared to the work of Eliades and Polycarpou 2012—that make use of the CUSUM method applied on Fourier transformations of the time series—the approach within this thesis combining seasonal and ARMA models with the CUSUM method seems superior resulting in much lower detection time delays. Furthermore, the algorithm showed the fastest response time to the leak of less than two hours. Certainly, the drawbacks of this method are that parameters have to be tuned on historical data and that the algorithm response in an indirect and not intuitively way to these parameters.

The likelihood-ratio test promised a method with less parameters to tune, although, this improvement is at the expense of a slightly slower detection time. However, the likelihood-ratio test is more robust to false alarms than CUSUM. A further advantage of the likelihood-ratio test is, that estimates of the leakage outflow—respectively the pressure drop—can be retrieved at run-time of the algorithm. Furthermore, these estimates also seemed quite stable over time, hence, early and good approximations are expected for subsequent leak localization.

In principle, Bayesian statistics can also be used to detect leaks. In fact, they equally perform as the likelihood-ratio test, which is fundamentally also utilizing Bayes' theorem. Another big advantage of the Bayesian method is that the start-time of the leak event can be estimated with high accuracy. Furthermore, this is possible without setting any parameter. Good estimates of the start of a leak are especially important for computing the total leak outflow used for (i) water balance calculations or (ii) retrieving good estimates for leak localization. The estimate of the leak was within 15 minutes—using all sensors for a leak with an outflow of $Q_L = 0.7$ (L/s). It has to be noted that for the computation of the total leak outflow, the leak end-time is also necessary. Logically, this is the time when the leak is finally repaired and thus known by the WU's staff.

Maybe the strongest point of the Bayesian formulation of the problem is the utilization of MCMC methods to retrieve probability distributions of the problem without using basically any input parameters. Additionally, any prior information can be incorporated in these algorithms. It is also imaginable, that the outcomes of a Bayesian leak detector can be used for feeding a probabilistic state estimators, e.g. a non-linear Kalman filter (D. Jung and Kim 2018), with whole probability functions for subsequent leak localization.

A drawback of the leak detection methods through time series modeling, in general, is that pressure measurements—at least in the example presented in this chapter—take some time to converge to their long time behavior. Hence, an early detection of leaks might be possible, unfortunately, the subsequent model-based leak localization might take some time until enough measurements are taken to stochastically retrieve proper estimates of the pressure values so that the leak can finally be localized. In the example presented in the chapter, the pressure fluctuations seemed more like a collective behavior caused by higher random demand coincidently at the same time the leak was introduced. More advanced methods—using not only correlations in the measurement data of a sensor itself, but using correlation between all measurement devices—might lead to faster and better estimates of the pressure-drop. This will be done in future work based on this thesis.

Simulations have shown that the detection time is also dependent on the model that is used. In general, the complex time series model is more robust to false alarms, but is also slower, because the ARMA model terms lead to a delay in the signal following changes in the time series. Furthermore, the ARMA model also seems to underestimate the changes in the signal. For that reason, the simple seasonal model is expected to lead to better leak localization. This problem is similar to the threshold setting problem. If one wants to detect a leak very early, then low thresholds are set, which leads to a lot of false alarms. If one wants to reduce the false alarms, higher thresholds must be set leading to longer detection times. Similarly, the simple seasonal model leads to earlier detection, but the complex ARMA model is more robust concerning false alarms.

All algorithms in this chapter work in real-time—if real-time means that arbitrary computations are faster than the time it takes to retrieve new measurements. That was achieved on the one hand through the efficient state space formulation of the ARMA models using a Kalman filter approach. The forecasting step and the updating step, which is made by only incorporating the current measurement, can be computed almost instantaneously. Additionally, the computational burden of Tukey's test, CUSUM method and the likelihood-ratio method as well as the analytical Bayes method was lower than the fraction of a second. Only the more sophisticated state estimation using MCMCs methods resulted in computation times in the order of a minute per sensor, which is still real-time given the definition at the beginning of this paragraph. Furthermore, the computations per sensor are independent of each other. Hence, they can be computed in parallel which will drastically reduce the computation time of all methods discussed in this section of less than a minute.

To finally conclude, the research question (see Q.1.1. in Section 1.5) if a small leak can be detected with pressure sensors only in the real-world—can be certainly answered in the affirmative. This is a promising result for WUs, that want to save money, but also do not want to give up early detection of leaks. Additionally, this argues in favor of installing pressure sensors in DMAs since they can indeed be useful for detecting abnormalities. Furthermore, they can support algorithms working with flow measurements to reduce false detection alarms.

Chapter 4

Optimal Sensor Placement

“Ford!" he said, “there’s an infinite number of monkey’s outside who want to talk to us about this script for Hamlet they’ve worked out."

— Douglas Adams, *The Hitchhiker’s Guide to the Galaxy*

This chapter is based on following three publications: Steffelbauer, Neumayer, et al. 2014, Steffelbauer and Fuchs-Hanusch 2016a and Fuchs-Hanusch and Steffelbauer 2017. The main focus of this thesis lies on Chapter 5—developing a model-based approach for locating small leaks and applying it on a real-world example. Certainly, the success of this technique goes hand in hand with the chosen (pressure) measurement positions in a system (D. Kang and Lansey 2010). Without doubt, the best possible locations of pressure sensors will significantly improve the localization performance. Therefore, this chapter focuses on how to find these optimal measurement positions for leak localization within a DMA. The problem itself is called an Optimal Sensor Placement (OSP) or sampling design problem (Savic et al. 2009).

An additional focus of this chapter is to examine effects of uncertainties on the OSP. A novel method is introduced which enables incorporation of uncertainties of all kind in an OSP algorithm by extending the projection based approach of Casillas, Puig, et al. 2013. The methodology is tested on simulations of a real-network (Linz-Pichling introduced in Section 2.4.2). Especially, the effect of demand uncertainties on pressure measurements and how this influences the ideal measurement positions is analyzed. For that reason, the problem is solved (i) for different numbers of sensors and (ii) for different magnitudes of uncertainties. Furthermore, a relation between the number of sensors and the leak localization performance is introduced defining a cost-benefit curves for the OSP problem. Furthermore, this chapter will compare six different OSP algorithms in a real-world network (the Graz-Ragnitz network introduced in Section 2.4.3)—two using structural information only, four utilizing leak sensitivity matrices, one algorithm is the proposed one capable of incorporating uncertainties. Seven different criteria are used to analyze the different placements resulting from the six OSP algorithms. Subsequently, sensors are installed at the resulting positions in the real-world. The leak localization performance of the different placements will be tested on real-leaks introduced in this network in Chapter 5.

4.1. Literature review

A large and growing body of literature has investigated the OSP problem for WDSs. Much of the existing literature on OSP in WDNs payed particular attention (i) on OSP for contaminant detection (Hart and Murray 2010; Rathi and Gupta 2014)—especially since the terrorist attacks in New York in 2001—and (ii) on WDS model calibration (see for example Savic et al. 2009). Surprisingly, only few studies are investigating the OSP problem for leak localization.

Only recently in the past 10 years, studies have directly addressed the OSP problem for leak localization—starting in 2008 with the work of (B. Farley et al. 2008, 2010, 2013) in England; and in Spain with the work of (Pérez, Puig, Pascual, Peralta, et al. 2009) nearly at the same time in 2009. Both concepts have in common that they use a so-called leak sensitivity matrix (see Section 4.2.2.1 for a detailed description) generated by hydraulic simulations. Furthermore, both approaches binarized this matrix leading to a loss of valuable information (Quevedo et al. 2011). An approach preventing this information loss was introduced by the work of Casillas, Puig, et al. 2013, where the OSP problem was based on the computations of projections with a non-binarized leak sensitivity matrix. Subsequently, a semi-exhaustive search strategy and a GA were used to solve the so formulated integer optimization problem.

Pérez, Cugueró, et al. 2014, on the other hand, formulated the problem as a minimization between the maximum distance to pre-calculated leak scenarios and a gravity center of nodes with projections larger than 99% of the maximum projection value. The optimization problem was solved with a greedy-search algorithm producing more robust sensor placements than the former binarization approaches. The approach of Casillas, Puig, et al. 2013 was enhanced by Cugueró-Escofet et al. 2017 by using a relaxed isolation index which enabled more practical considerations like an acceptable isolation distance in the problem formulation.

Sarrate, Nejjari, et al. 2012 focused on pure structural analysis of WDS. A leak isolability index was defined and maximized with a Depth-First Search (DFS) algorithm. Unfortunately, this approach only worked for medium sized networks. That is why Sarrate Estruch et al. 2013 respectively Sarrate, Blesa, Nejjari, and Quevedo 2014 utilized graph clustering techniques to reduce the problem complexity and made it applicable in real-world WDNs. Of course, focusing only on the structure of WDNs without considering hydraulic principles stated a radically simplification of the OSP problem. Thus, this approach can not alone ensure good leak localization performances in real networks due to the simple description. That is why Sarrate, Blesa, and Nejjari 2014 developed a method to combine the approaches with the leak sensitivity matrix by using projections again.

The recent approach of Casillas, Garza-Castañón, et al. 2015 proposed as OSP performance measure overlapping signatures in a leak signature space. Minimizing these overlaps lead to ideal sensor positions. Similarly, Nejjari et al. 2015 suggested as a sensor placement performance measure the minimization of the average worst leak expansion distances calculated from sensitivity matrices. Subsequently, the problem was solved through exhaustive search.

Dissimilar to the above mentioned approaches, Christodoulou et al. 2013 offered a different way of placing sensors by maximizing the total entropy—the information retrieved from a certain placement of sensors—in the network. Already in 2000, Schaetzen et al. 2000 made use of the Shannon entropy for OSP, although, the problem was formulated for model calibration.

Besides the problem of placing sensors in an optimal way itself, the problem gains additional complexity by taking uncertainties into account. Unfortunately, hydraulic models and measurements are fraught with several sources of uncertainty (Hutton, Kapelan, et al. 2014). Not surprisingly, these uncertainties may affect the optimality of sensor positions. Therefore, these effects should be taken into account in the OSP problem to obtain more robust placements under uncertainties. Unfortunately, this topic was even less examined in past studies than the OSP problem for leak localization.

Blesa et al. 2014 studied the robustness of the work of Sarrate, Blesa, and Nejjari 2014 against sensitivity matrix uncertainties. They found that sensor positions were not sensitive on the leak magnitude, but they heavily depend on the working point (boundary conditions) of the WDS. Furthermore, Blesa et al. 2015 formulated a multi-objective problem for OSP—minimizing the average and the worst leak isolability index. Recently, Soldevila et al. 2018 used hybrid feature selection for a classifier-based sensor placement algorithm that was robust to demand uncertainties and measurement noise. The different OSP approaches for model-based leak localization are summarized in Table 4.1.

Table 4.1.: This table gives a short overview on the literature of OSP for model-based leak localization and methodologies are listed in alphabetical order.

Literature	**Description**
Blesa et al. 2015	Multi-objective minimization of (i) average and (ii) worst leak isolability index
Casillas, Puig, et al. 2013	Projection-based non-binarized sensitivity matrix
Casillas, Garza-Castañón, et al. 2015	Reduction of leak sensitivity space overlaps in leak signature space
Christodoulou et al. 2013	Maximum entropy
Cugueró-Escofet et al. 2017	Relaxed leak isolation index
B. Farley et al. 2008, 2010, 2013	Binarized sensitivity matrix
Nejjari et al. 2015	Minimization of average worst leak expansion distance
Pérez, Puig, Pascual, Peralta, et al. 2009	Binarized sensitivity matrix
Pérez, Cugueró, et al. 2014	Minimization of maximum distance to gravity center with projections $> 90\%$
Sarrate, Nejjari, et al. 2012	Structural analysis with DFS
Sarrate Estruch et al. 2013	Graph clustering of structural analysis
Sarrate, Blesa, and Nejjari 2014	Combination of graph clustering and sensitivity matrix
Soldevila et al. 2018	Classifier-based hybrid feature selection

4.2. Methodology

4.2.1. The optimal sensor placement problem, parameter space size and solution strategies

It is important for leak detection that sensors are able to observe signals of all possible leaks in a system. Additionally, to localize the leak, these signals have to be distinguishable by the sensors corresponding to the leak's location in a system. This is called leak isolability. Thus, an optimal sensor placement should fulfill both criteria, the (i) leak detectability and (ii) the leak isolability. All former mentioned sensor placement approaches contain both criteria directly or indirectly in their formulation. Furthermore, all different OSP algorithms are expressed in terms of an objective function $f(\mathbf{x})$ which has to be minimized (or maximized), either by exhaustive search methods (for small and medium sized systems) or by the optimization methods already described in Section 2.1. Most of the time the OSP problem is solved with heuristics like e.g. GAs. The parametrization of the OSP problem is based on a vector $\boldsymbol{x}$

$$\boldsymbol{x} = (n_1, \ldots, n_N)^T \quad . \tag{4.1}$$

The length N of this vector corresponds to the number of sensors that are intended to be placed. The elements x_i of $\boldsymbol{x}$ contain the sensor positions encoded as integer values n_i in the range of $n_i \in [1, \ldots, M]$ where M is the number of possible measurement locations. Each number is unequivocally assignable to a potential measurement node in the system.

In general, the OSP problem is a combinatorial optimization respectively an integer programming problem (see Section 2.1.1). This property results through the combination of discrete sensor positions which build the parameter space. The search spaces for OSP problems are huge. Solving $f(\mathbf{x})$ for every possible combination of sensors in a WDS results in an almost countless[24] parameter respectively solution space. For example, if there are M different possible measurement positions for pressure sensors in a network, there are $\mathcal{C}$ possible sensor combinations for placing N sensors

$$\mathcal{C} = \binom{M}{N} = \frac{M!}{N!(M-N)!} \quad . \tag{4.2}$$

Hence, $\mathcal{C}$ independent computations of the objective function $f(\mathbf{x})$ have to be evaluated. Furthermore, this number grows exponentially with M: using the Stirling approximation $n! \propto \sqrt{2\pi n}(n/e)^n$ for $n \to \infty$ reduces equation (4.2) to

$$\mathcal{C} = \binom{M}{N} \approx \frac{M^N}{N!} \quad \text{for} \quad N << M \quad . \tag{4.3}$$

The constrained $N << M$ is always the case for OSP in real WDS. Placing $M = 5$ sensors, for instance, in a medium sized network with $N = 200$ nodes results in $\mathcal{C} \approx 2.5$ billion possible sensor combinations, respectively, 2.5 billion possible solutions for the problem. This is why exhaustive search methods are unrewarding.

If the optimal solution can be retrieved by placing one sensors after another—called from now on a

[24] Countless as in the sense of the human perception of a gigantic entity, not in the sense of the strict mathematical definition of not being countable, e.g. uncountably infinite.

greedy sensor placement algorithm—the search would be drastically reduced to

$$M(M-1)\dots(M-N+1) = \frac{M!}{(M-N)!} \approx M^N \qquad . \tag{4.4}$$

Additionally, all possible solutions for all sensor numbers from 1 to M would be retrieved on the way to find the optimal solution for M sensors. Unfortunately, already Kapelan et al. 2005 showed that the OSP problem is not solvable by a greedy algorithm for hydraulic calibration problems.

For model-based leak localization, the malfunctioning of greedy-algorithms has not been shown, yet. But if that is the case, meta-heuristics like GAs are the only alternative for finding optimal solutions respectively near-optimal solutions in reasonable time for problems with this enormous search space sizes. Fortunately enough, they are very effective for finding sub-optimal solutions in high dimensional and large parameter spaces.

4.2.2. Sensor placement algorithms for comparison

To compare different sensor placement algorithms, two fundamentally different approaches are studied within this thesis: (i) topology-based optimal sensor placement algorithms (see Section 4.2.2.2) and (ii) sensitivity-based methods (see Section 4.2.2.3).

The first methods utilize algorithms from the broad field of graph theory. As already mentioned in Section 2.2.4, a WDN can be seen as a mathematical graph. The second class of methods takes advantage of the leak sensitivity matrix to find optimal sensor positions. That is why the sensitivity matrix and how to obtain it is explained in more detail in Section 4.2.2.1.

4.2.2.1. The leak sensitivity matrix

The sensitivity matrix S is the key in deciding where to install pressure measurement devices in a WDS (Pudar and Liggett 1992) and used in a lot of OSP algorithms. The reason for that is that S represents the effect of leaks on pressure measurement points. The leak sensitivity matrix S is computed in following way

$$S = \begin{pmatrix} s_{11} & \dots & s_{1n} \\ \vdots & \ddots & \vdots \\ s_{m1} & \dots & s_{mn} \end{pmatrix} \qquad , \tag{4.5}$$

where m is the number of leak scenarios and n is the number of possible measurement positions.

The individual elements s_{ij} of S represent the change of pressure at point j due to a leak at location i in the network

$$s_{ij} = \frac{\partial p_j}{\partial f_i} \qquad . \tag{4.6}$$

Real WDS modeling is a large scale problem involving lots of non-linear equations that depend on numerous parameters (see Section 2.2). That is why computing S analytically is not practicable (Pérez, Puig, Pascual, Quevedo, et al. 2011). Hence, a first order Taylor series approximation of the analytical non-linear leak-pressure behavior is performed. These first order derivatives can numerically be resembled by using the forward-difference (FD) approximation (Naumann 2012)

$$\frac{\partial f(\boldsymbol{x})}{\partial x_i} = \frac{f(\boldsymbol{x}+h\boldsymbol{e_i}) - f(\boldsymbol{x})}{h} + O(h) \quad . \tag{4.7}$$

$n+1$ function evaluations are necessary to build the derivative, because one function evaluation results from the working point evaluation $f(\boldsymbol{x})$. Alternatively, the central-difference (CD) approximation (Naumann 2012) can be used

$$\frac{\partial f(\boldsymbol{x})}{\partial x_i} = \frac{f(\boldsymbol{x}+h\boldsymbol{e_i}) - f(\boldsymbol{x}-h\boldsymbol{e_i})}{2h} + O(h^2) \quad . \tag{4.8}$$

This approximation needs $2n$ function evaluations to differentiate $f(\boldsymbol{x})$. In this thesis, the first derivatives are numerically approximated by using the forward-difference method from equation (4.7). Linking this to WDS simulation, the single elements of the sensitivity matrix s_{ij} are calculated by subtracting $\hat{p}_j$—the leak free pressure—from the pressures which are calculated at the same position $p_j^{f_i}$ under leak scenario i with a leak of magnitude f_i and normalizing the resulting value by the leak size

$$s_{ij} = \frac{\partial p_j}{\partial f_i} = \frac{p_j^{f_i} - \hat{p}_j}{f_i} \qquad . \tag{4.9}$$

The computational burden for computing S using equation (4.7) is given by solving the hydraulic model $m+1$ times in the system. The pressures at all measurement positions m are retrieved from a single EPANET evaluation for a specific leak scenario. The additional simulation emerges from a hydraulic simulation of the leak free case to obtain $\hat{p}_j$. Using the more accurate approximation in equation (4.8) will double the computational cost, but would also decrease the approximation error. Nevertheless, for small leaks in the system, the analytical and this approximated sensitivity converge (according to Pérez, Quevedo, et al. 2011).

It has to be mentioned that S—respectively the sensor placements making use of S—heavily depend on the working point of the system defined by the demand loading and boundary conditions in the hydraulic simulation (Pérez, Puig, Pascual, Quevedo, et al. 2011). The influence of the leak magnitude is negligible (Blesa et al. 2014).

The above presented method is also called influence coefficient method or parameter perturbation method (Sanz Estapé 2016). Obviously, the method is computationally slow. More advance methods exist to retrieve S through a single[25] hydraulic simulation (see for example the equation perturbation method by Cheng and He 2011).

However, the parameter perturbation approach is chosen, because of its simplicity to implement. Furthermore, the computation time can be reduced on modern multi-core computer environments by parallel computation of the single elements of S. For the sensor placement problem, most computational effort arises due to the incredible huge size of the search space (see Section 4.2.1) and the computation of S plays only a minor part. Thus, more sophisticated approaches to calculate S are neglected in this thesis. Further methods how to compute S can be found in (Kapelan et al. 2003) or (Sanz Estapé 2016), for the interested reader.

In order to estimate the general sensitivity of a potential measurement point with respect to all possible

[25] respectively two, again one simulation is needed for the leak free case.

leak scenarios, the mean over the column of S is calculated in the following way

$$\bar{s}_j = \frac{1}{m} \sum_{i=1}^{m} s_{ij} \qquad . \tag{4.10}$$

This calculation serves for the graphically representation of the sensitivity of a certain measurement locations to leaks in general (see for example Figure 4.2a) and is closely related to the leak detectability.

Residual matrix Some leak localization algorithms and OSP approaches also rely on the computation of residuals, hence they are described here. The residuals express the difference between real-world measurements and the expected hydraulic system behavior. Consequently, the residuals r_{ij} are computed similarly to the sensitivity matrix, but in general with different leak sizes f_i and by omitting the normalization

$$r_{ij} = p_j^{f_i} - \hat{p}_j \qquad . \tag{4.11}$$

In total, the R matrix represents the measurements in a system. Later on in this thesis, the i-th column of R will be called i-th residual vector $\mathbf{r_i}$.

4.2.2.2. Graph-based optimal sensor placement

The first two algorithms—Shortest Path 1 and 2—have been proposed by Schaetzen et al. 2000 originally for hydraulic model calibration. Both algorithms make use of the network graph only without utilizing any hydraulics. Consequently, both algorithms employ shortest path algorithms to retrieve the ideal sensor positions. Thus, "these rankings are not optimal in a hydraulic sense but are practicable and correlate well with the information required in the measurements for estimating parameters of a hydraulic network model" (according to Schaetzen et al. 2000). Basically, both algorithms prefer measurement points that are (i) dispersed over the whole network, (ii) having a high coverage in a shortest path sense per definition, (iii) as wide as possible away from sources (e.g. tank) and therefore are possibly highly sensitive to pressure changes. That is the reason why they are used for leak localization as well in this thesis.

4.2.2.2.1. Shortest Path 1 The first algorithm proposed by (Schaetzen et al. 2000) chooses pressure measurement points by utilizing the shortest path algorithm by Dijkstra 1959 on the network graph, where the weight ω associated to a pipe is its corresponding length l (see Section 2.2.4). The shortest path from the water source (tank, reservoir or DMA inlet) to each node in the network is calculated. The node with the longest shortest path length— a min-max optimization problem—is chosen as measurement location. Afterwards, a pipe with length zero is added between the source and the measurement node, consequently changing the shortest path lengths in the network. Then this procedure is repeated until the demanded number of sensors is placed. As a consequence, this OSP method prefers sensor locations near the interior of the network (according to Schaetzen et al. 2000),

4.2.2.2.2. Shortest Path 2 The second algorithm operates in a similar manner, but instead adding a pipe of zero length, all lengths of the pipes belonging to the longest shortest path are set to zero.

According to Schaetzen et al. 2000, this method prefers sensor locations in the outer edges of the network.

If a WDS possesses several sources, then this sources are connected with pipes of length zero prior to estimating the optimal sensor positions. Logically, the first optimal sensor position is always the same for both algorithms. The differences between the algorithms start as recently as the first measurement point is identified and different pipe lengths are added to the network. Especially in branched pipe networks, the ideal positions between the two algorithms differ.

Finally, it has to be mentioned that both algorithms—Shortest Path 1 as well as Shortest Path 2—are in fact greedy sensor placement algorithms per definition as described in Section 4.2.1 equation (4.4). Because of their greediness as well as their utilization of highly efficient graph theoretic algorithms, they are very economical in terms of computational power. Thus, they are applicable even on huge networks consisting of thousands of nodes without any problems—even on ordinary desktop computers.

4.2.2.3. Sensitivity-based optimal sensor placement

This section covers the later-used sensitivity-based approaches. All concepts have in common that they make more or less use of the leak sensitivity matrix presented in Section 4.2.2.1. Whereas all approaches utilize the sensitivity matrix, they all differ in how they use it to find optimal pressure measurement locations. Hence, all OSP algorithms lead to different "optimal" measurement positions. The optimality of the sensor positions resulting from the different algorithms will be examined later on in this thesis by testing them in the real-world.

4.2.2.3.1. Shannon-Entropy This approach is also presented in Schaetzen et al. 2000 together with the shortest path algorithms already described in Section 4.2.2.2.1 and 4.2.2.2.2. Although, this approach is originally formulated for roughness calibration based on pressure measurements in the paper of Schaetzen et al. 2000. It is reformulated to fit the OSP methodology for model-based leak localization. As the name already indicates, this OSP method makes use of the Shannon entropy. Shannon introduced this term as a probabilistic measure of uncertainty tightly connected to the information about system. In mathematical terms, the Shannon entropy is formulated in following way

$$S(p) = -\sum_{j=1}^{m} p_j \ln p_j \quad \text{with} \quad \sum_{j=1}^{m} p_j = 1 \quad \text{and} \quad 0 \leq p_j \leq 1 \quad \forall j \in [1,m], \tag{4.12}$$

where p_j is a probability distribution function. Finding the distribution that will result in the maximum entropy will result in the most unbiased information. Thus, a sensor placement with maximum entropy would lead to an even spread of measurement locations in a system (according to Schaetzen et al. 2000).

Schaetzen et al. 2000 formulates two objectives to be maximized. They are later combined in a single objective. Both objectives are based on the sensitivity matrix and the Shannon entropy function derived from it. The sensor positions of a placement are given by the parameter vector $\boldsymbol{x} = (n_1, \ldots, n_N)^T$ containing the N measurement locations decoded as integers $n_i \in [1,n]$ with n the number of possible measurement locations.

The first objective maximizes the sensitivity

$$F_1(\boldsymbol{x}) = \sum_{j=1}^{m} s_j, \quad \text{where} \quad s_j = \max_{i \in \boldsymbol{x}}(s_{ij}) \quad . \tag{4.13}$$

m is again the number of leak scenarios and s_{ij} is the sensitivity matrix in equation (4.5). Maximizing $F_1(\boldsymbol{x})$ leads to high detectability of leaks by placement $\boldsymbol{x}$.

The second objective maximizes the entropy function

$$F_2(\boldsymbol{x}) = S(p) = -\sum_{j=1}^{m} p_j \ln p_j \quad \text{with} \quad p_j = \frac{s_j}{\sum_{j=1}^{m} s_j} \quad . \tag{4.14}$$

Subsequently, the two objectives are combined in following way

$$F(\boldsymbol{x}) = \sqrt{\frac{1}{2}\sum_{i=1}^{2}\left(\frac{F_i(\boldsymbol{x}) - \max(F_i(\boldsymbol{x}))}{\min(F_i(\boldsymbol{x})) - \max(F_i(\boldsymbol{x}))}\right)^2} \quad \rightarrow \quad \min_{\boldsymbol{x}}(F(\boldsymbol{x})). \tag{4.15}$$

This optimization problem introduced by minimizing equation (4.15) is solved using a GA.

4.2.2.3.2. Perez Binarization approach This OSP methodology—in fact one of the first OSP method for model-based leak localization—is based on the standard theory of model-based diagnosis. Leakage localization is based on consistency checks using the residual vectors $\mathbf{r_i}$ defined in Section 4.2.2.1. The OSP itself again originates from the leak sensitivity matrix S defined in Section 4.2.2.1. Since some potential sensor positions are more sensitive to leaks than others, S is normalized to assure comparability between the different columns

$$s_{ij} = \frac{s_{ij}}{\max(s_{i1}, \ldots, s_{in})} \quad . \tag{4.16}$$

Consequently, the maximum normalized sensitivity should lie on the diagonal. The normalization is followed by a binarization process to obtain the binary leak signature matrix ϕ_{ij}

$$\phi_{ij} = \begin{cases} 0 \text{ if } |s_{ij}| \leq \tau \\ 1 \text{ if } |s_{ij}| > \tau \end{cases} \quad . \tag{4.17}$$

τ is a threshold that has to be defined. It has to mentioned that the binarization of S goes hand in hand with an information loss of the investigated system (Quevedo et al. 2011).

Finding the optimal τ is important, since small τ lead to all elements becoming 1 and large $\tau \approx 1$ lead to all elements becoming zero except for the diagonal. The ideal τ is found through exhaustive analysis of the behavior of ϕ_{ij} on τ in the following way: A leak signature is a unique column in S. The number of all signatures is compared to the maximum number of nodes having the same signature. The ideal τ is found where these two numbers are equal. Additionally, the maximum number of signatures corresponds to the isolability of a certain placement and can be used for a cost-benefit analysis similar to Section 4.2.3 (Pérez, Puig, Pascual, Quevedo, et al. 2011).

Subsequently to binarization, the objective of the OSP algorithm is to find a sensor distribution $\boldsymbol{x}$ that minimizes the number of nodes for the largest set of leaks with the same signature. This is again an optimization task and can either be solved through exhaustive search, semi-exhaustive search like e.g. branch and bound search methods, or as always in this thesis with a heuristic optimization method, a GA as described in Section 2.1.2.2.1.

4.2.2.3.3. Casillas Projection approach The sensor placement algorithm described in Casillas, Puig, et al. 2013 manages the OSP without the information loss through binarization of S as the approach described in Section 4.2.2.3.2. This OSP approach makes full use of S by calculating projections ψ_{ij} between the residual vecors $\mathbf{r}_i$ and $\mathbf{s}_j$

$$\psi_{ij}(\mathbf{q}) = \frac{\mathbf{r}_i^T Q(\mathbf{q})\mathbf{s}_j}{|\mathbf{r}_i^T Q(\mathbf{q})^T||Q(\mathbf{q})\mathbf{s}_j|} \quad . \tag{4.18}$$

$Q(\mathbf{q})$ is a diagonal matrix constructed from a binary vector $\mathbf{q}$ with the length of the possible sensor positions, where q_i is 1 if a sensor is placed at node i respectively 0 for no sensor.

Mathematically, $\boldsymbol{q}$ and $\boldsymbol{x}$ (see the definition in equation (4.1)) are connected through following relation

$$q_i = \begin{cases} 1 \text{ if } i \in \boldsymbol{x} \\ 0 \text{ if } i \notin \boldsymbol{x} \end{cases} \quad . \tag{4.19}$$

A translation operator $\hat{T}$ is defined following the relationship between $\boldsymbol{q}$ and $\boldsymbol{x}$ in equation (4.19) to keep consistent with the notations of the OSP objective functions for the other placements which are all dependent on $\boldsymbol{x}$

$$\boldsymbol{x} = \hat{T}(\boldsymbol{q}) \quad \text{resp.} \quad \boldsymbol{q} = \hat{T}^{-1}(\boldsymbol{x}) \quad . \tag{4.20}$$

The largest projection value represents the found leak spot. The correct leak spot is found by a sensor configuration $\mathbf{x}$ if the maximum value is part of the matrix's diagonal. Subsequently, $\varepsilon_i(\mathbf{x})$—an error index corresponding to a certain sensor placement for a specific leak scenario—is defined as following

$$\varepsilon_i(\mathbf{x}) = \varepsilon_i(\hat{T}(\mathbf{q})) = \begin{cases} 0 \ \ldots \text{ if } \psi_{ii}(\mathbf{q}) = \max(\psi_{i1}(\mathbf{q}), \ldots, \psi_{im}(\mathbf{q})) \\ 1 \ \ldots \text{ otherwise} \end{cases} \quad . \tag{4.21}$$

The objective function of the OSP algorithm $f(\boldsymbol{x})$ is defined as the mean over $\varepsilon_i(\mathbf{x})$ for all leak scenarios taking every possible leak into account

$$f(\mathbf{x}) = \sum_{i=1}^{m} \frac{\varepsilon_i(\mathbf{x})}{m} \quad \rightarrow \quad \min_{\mathbf{x}} f(\mathbf{x}) \quad . \tag{4.22}$$

Minimization results in the optimal sensor configuration $\mathbf{x}^*$. The lower bound of $f(\mathbf{x})$ is 0, denoting that every leak in the system is located correctly by the sensor configuration $\mathbf{x}$. The upper bound is $f(\mathbf{x}) = 1$. That is the case if no leak scenario is identified correctly. The optimization problem can be solved again with the optimization strategies defined in Section 2.1. In this thesis, GAs will be used to solve this problem (see Section 2.1.2.2.1).

4.2.2.4. SPUDU - Sensor placement under demand uncertainties

The optimality of measurement locations in WDS do not only depend on the detectibility and isolability of leaks, it also depends on uncertainties arising from (i) model inaccuracies or (ii) random fluctuations in measurements over time (as, for example, originating from uncertainties of customer demand over time[26]). Therefore, a sensor placement methodology has been developed allowing to incorporate uncertainties in the OSP (Steffelbauer and Fuchs-Hanusch 2016a).

The idea behind this methodology is that uncertainties σ_i depend on the measurement locations. These uncertainties can be model output uncertainties retrieved, for example, from MCS. Some measurement locations are more sensitive to uncertainties than other points. Hence, points that are highly sensitive to uncertainties are less ideal to place sensors at, since the measured quantities can be concealed by these uncertainties. This is why points with high uncertainties should be omitted by an OSP algorithm.

Incorporating uncertainties using optimization algorithms, this can be done in two ways: (i) explicitly or (ii) implicitly in the optimization problem.

(i) The first approach can be formulated as punishing the fitness values of measurement locations by adding a penalty function g proportional to the uncertainties at these points

$$f(\mathbf{x}) \rightarrow f(\mathbf{x}) + g(\sigma(\mathbf{x})) \quad . \tag{4.23}$$

(ii) The second approach incorporates the uncertainties σ implicitly in the OSP problem itself

$$f(\mathbf{x}) \rightarrow f(\mathbf{x}, \sigma) \tag{4.24}$$

In this thesis, the latter approach has been chosen by extending the method developed by Casillas, Puig, et al. 2013 which was already discussed in Section 4.2.2.3.3. This has the advantage that instead of two functions—$f(\mathbf{x})$ and $g(\sigma(\mathbf{x}))$—only one function has to be evaluated.

The projections in the OSP method of (Casillas, Puig, et al. 2013) can be seen as a scalar product between the residual vectors and the corresponding column of S. The residuals are intended to be perturbed by subtracting weighted uncertainty vectors σ_{p_i} corresponding to the measurement points. This results in small deviations of the direction of the residual vector. The variation is proportional to the strength of the uncertainty[27]. The so caused mismatch between the unperturbed sensitivities and the perturbed residuals leads to a penalization of measurement locations associated with high uncertainties. Mathematically, this implicit approach is formulated in following way

$$\psi_{ij}(\mathbf{q}) = \frac{\left(\mathbf{r}_i^T - \omega\sigma_{p_i}\right) Q(\mathbf{q})\mathbf{s}_j}{|\left(\mathbf{r}_i^T - \omega\sigma_{p_i}\right) Q(\mathbf{q})^T||Q(\mathbf{q})\mathbf{s}_j|} \quad . \tag{4.25}$$

ω is a weighting factor controlling the strength of the influence of σ and will be investigated in more detail later. High distortion of the residual vectors $\mathbf{r}_i$ result in smaller $\psi_{ij}(\mathbf{q})$. Therefore, points with sensors at these positions will be less optimal.

Subsequently, the OSP approach will follow strictly the procedure already described in Section 4.2.2.3.3 continuing from equation (4.21) to build $f(\boldsymbol{x})$ which is again optimized by GAs.

[26] Measurement uncertainties originating from the sensors itself are depending on the device and not on the location.

[27] Note, that the approach is applicable for any kind of uncertainties.

4.2.3. Sensor placement cost-benefit analysis considering uncertainties

Up to now, most previous studies have investigated OSP problems were the number of measurement devices to be deployed was assumed to be known in advance. Thus, the number of sensors that are going to be placed was treated as a fixed parameter in OSP algorithms. However, for WUs, it is very interesting to know how many sensors are needed in a WDN to achieve a certain quality of leak localization. Furthermore, they are interested in the expected benefit of installing further k sensors in their system. A few studies exist that have investigated the behavior of the sensor number on the sensor placement quality parameter defined in the respective methodology. See, for example, Pérez, Puig, Pascual, Peralta, et al. 2009, Casillas, Puig, et al. 2013 and Sarrate, Nejjari, et al. 2012. However, all of the former mentioned studies have in common that they plainly state the pure simulation results without aiming to find a mathematical law behind the quality as a function of the sensor number, nor did any of this studies consider uncertainties.

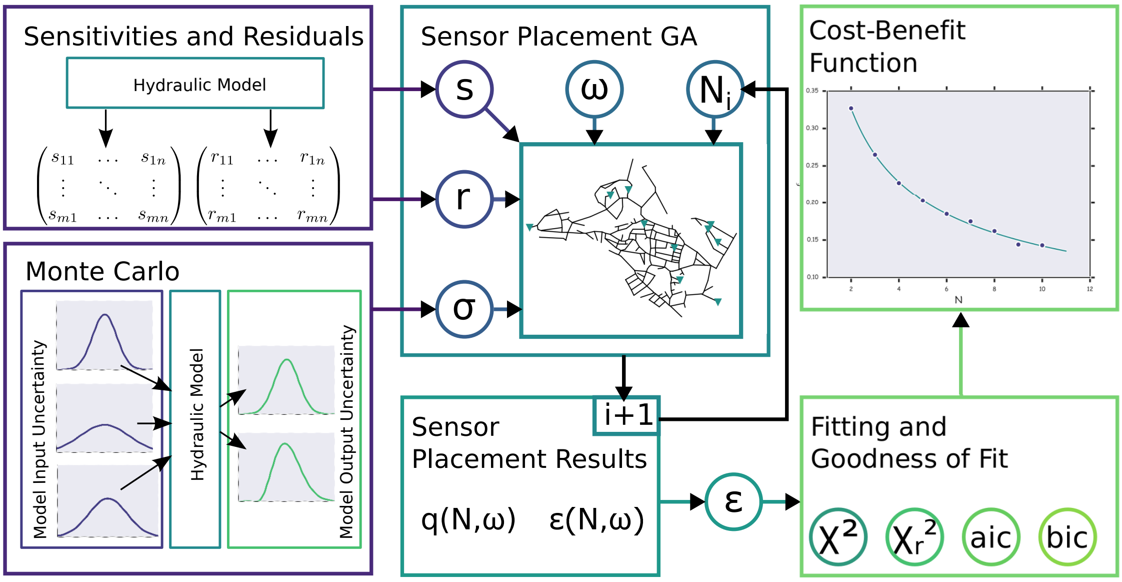


Figure 4.1.: Flowchart of the SPUDU methodology (taken from Steffelbauer and Fuchs-Hanusch 2016a)

The following questions—(i) how many sensors are needed and (ii) how do uncertainties influence the number of sensors—are tried to be answered with the proposed methodology in this section. It consists of a closer look on fitness function outcomes of the OSP computations in dependency of the sensor number and in the presence of uncertainties of certain strength.

An overview of the followed approach is depicted in Figure 4.1 (taken from Steffelbauer and Fuchs-Hanusch 2016a). First, the sensitivity matrices S and the residuals $\mathbf{r}$ (see section 4.2.2.1) are computed through hydraulic simulations. S and r are later used as inputs for the OSP algorithm described in Section 4.2.2.4. The hydraulic model is also used to calculate the effect of uncertain input parameters on the outputs of the hydraulic model (σ) through MCS (see Section 2.3.1). Subsequently, the computed σ is integrated into the SPUDU OSP algorithm described in Section 4.2.2.4. The number N of sensors, a weighting factor ω to adjust the strength of the uncertain demands, S, r and σ are used for finding optimal measurement locations. These locations possess a leak localization with a certain minimal fitness value $f(\mathbf{x}^*)$.

Running an GA for a fixed ω and different number of sensors N results in different $f(\mathbf{x}^*|N,\omega)$ values. The fitness values can additionally serve as a sensor placement quality parameter, since they are connected to the number of leak scenarios which have not been located accurately (see equations (4.21) and (4.22)). The percentage $L_{\%}$ of theoretically accurate located leaks can be defined through following equation

$$L_{\%}(\mathbf{x}^*|N,\omega) = 100 \cdot (1 - f(\mathbf{x}^*|N,\omega)) \qquad . \tag{4.26}$$

The calculation of $f(\mathbf{x}^*|N,\omega)$ for different N and ω allows the derivation of sensor placement cost-benefit functions. For reasons of simplicity, the cost is simplified by the sensor number and the benefit is given as the quality of a certain placement in terms of leak scenarios located correctly or $f(\mathbf{x})$. In this thesis, different functions are tried to describe the behavior of $f(\mathbf{x}^*)$ as a function of N. Certainly, $f(\mathbf{x}^*)$ should decrease with increasing N. Logically, more sensors should lead to better leak localization quality.

The cost-benefit analysis is expensive regarding computational costs. For every N and ω pair a time consuming optimization problem has to be solved. That is why a general cost-benefit functions describing the behavior of $f(\mathbf{x})$ on N potentially promises computational shortcuts and might lead to better insights into OSP problem itself.

Concerning the cost-benefit evaluation, two questions arise: (i) how to fit a function optimally to $f(\mathbf{x})$ and (ii) how to decide which function is best?

(i) States and optimization problem which is solved with the Levenberg-Marquardt algorithm as described in Section 2.1.2.1.4.

(ii) The GoF statistics described in Section 2.3.4 are used to ascertain which is the true cost-benefit behavior of the OSP problem.

4.3. Results and discussion

This section summarizes the results of the OSP analysis. First, the developed methodology of how to incorporate uncertainties in a OSP algorithm—the SPUDU method (see Section 4.2.2.4)—is presented in detail and evaluated on the Linz-network (see Section 2.4.2) for different sensor numbers as well as different strengths of uncertainties. A special focus will be laid on the effect of demand uncertainties on optimal sensor locations. Subsequently, the results are used to conduct a cost-benefit analysis of the OSP problem additionally with different sizes of uncertainties using the methodology described in Section 2.3.4. The results are already published in the related journal paper to this thesis (Steffelbauer and Fuchs-Hanusch 2016a).

Second, the different OSP algorithms described in Section 4.2.2 will be applied on the Ragnitz-Detailed network (see Section 2.4.3) for placing $N = 5$ sensors. The dissimilar "ideal" OSP results will be discussed in detail. The resulting OSP positions of all algorithms will be used to install pressure measurement devices at hydrants in the real-world conducting a field-study to evaluate the performance of the algorithms under real-world conditions. The results of this study will be presented in the next chapter on model-based leak localization. Some results of this study are also already published in a conference publication (Fuchs-Hanusch and Steffelbauer 2017).

4.3.1. Sensor placement under demand uncertainties

Following the methodology described in Section 4.2.3, first the uncertainties are computed, followed by sensitivity matrix calculations which are both then used for retrieving OSP locations from SPUDU algorithm for different numbers of sensors N and different strengths of uncertainty σ. Subsequently, the results will be used to conduct a cost-benefit analysis which will compare sensor number against leak localization quality.

4.3.1.1. Model output uncertainty computation

The SPUDU approach will be shown on the example of incorporating demand uncertainties on pressure measurement points in the OSP algorithm. In general, every combination of model input–model output uncertainty can be applied in this OSP methodology.

mcsteps parameter sets are realized for MCS, thus, each demand in the WDS is randomly drawn from a normal distribution. The mean μ_q is chosen to be the actual nodal demand during MNF at the junction, standard deviation σ_q is chosen to be 10% of the actual demand. Hydraulic simulations are performed with EPANET using the OOPNET interface (Steffelbauer and Fuchs-Hanusch 2015) to evaluate each parameter set. Simulations are conducted in parallel. The resulting pressures p_j at every node j in the network are consolidated to form the probability distribution functions $P(p_j)$ of the model output uncertainties for every possible measurement position. The standard deviation σ_{p_j} is taken from this distribution serving as a measure of the pressure uncertainties at the corresponding measurement location j. High σ_{p_j}—hence high noise in pressure signals at these point—indicate less ideal pressure measurements points. Simulations with a sample size of $mcepts = 10000$ are conducted and the resulting σ_{p_i} are depicted in Figure 4.2b.

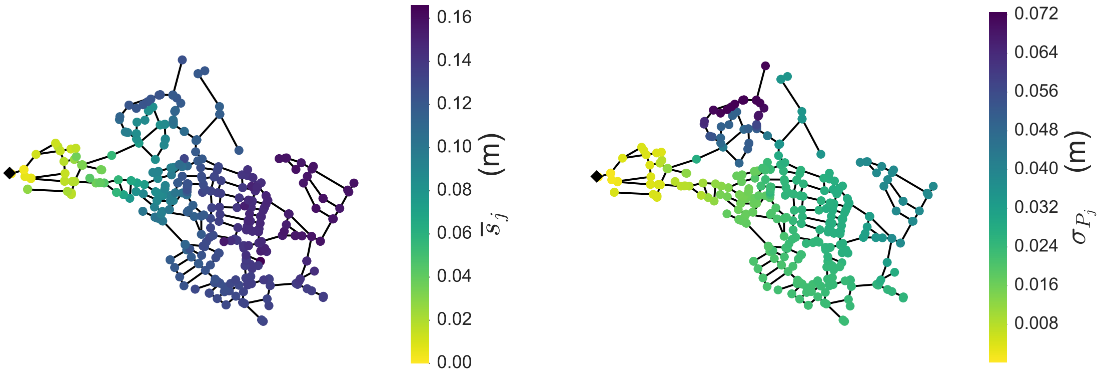
It can clearly be seen that a region in the upper part of the network exists with very high demand uncertainties. This big uncertainties originate from a big customer in this part of the network. Consequently, these points should be punished by the OSP algorithm, if the method works properly.

4.3.1.2. Sensitivity matrix computation

After the computation of demand uncertainties, the sensitivity matrix S is calculated (see Section 4.2.2.1). Every node in the WDS serves as both, a possible leak location as well as a possible measurement point. For generating the leak scenarios, the same emitter coefficient of $c_e = 0.5$ ($e_e = 0.5$) was used at every node in the system. Only one leak size was chosen to build the sensitivity matrices, because according to the work of Blesa et al. 2014 the sensor positions should not be sensitive to leak magnitudes, they are more sensitive to the working point of the WDS. The overall sensitivity for every node according to equation (4.10) is depicted in Figure 4.2a. The WDS system appears to be sensitive to pressures at the right hand side. That are the points that are as far as possible away from the DMA inflow point at the left side (diamond symbol). The residuals R—which are also needed for the OSP algorithm—are calculated with the same emitter coefficients as S.

4.3.1.3. Optimal sensor placement results

Consequently, the previous calculated sensitivities S, residuals R (both in Section 4.3.1.2) and model output uncertainties σ_{p_i} (see Section 4.3.1.1) are used for computing the optimal measurement loca-



(a) Leak Sensitivity

(b) Model Output Uncertainty

Figure 4.2.: Overall sensitivity of nodal pressures to leaks (left) and effects of the demand uncertainties on nodal pressure measurements in the Linz network (right) (taken from Steffelbauer and Fuchs-Hanusch 2016a)

tions using the developed SPUDU algorithm described in Section 4.2.2.4. The OSP problem is solved for a different $N \in \mathbb{N} \cap [2,\dots,10]$ and different $\omega \in \{0.0, 0.125, 0.25, 0.5, 1.0\}$.

ω was chosen between the case with no uncertainties ($\omega = 0.0$) and the case considering uncertainties with full strength ($\omega = 1.0$). Simulations using $\omega = 1.0$ result in a clustering of sensors in low-uncertainty regions. Thus, the sensors are not evenly spread over WDS (see for example Figure 4.4d). For that reason, ω was divided by two until sensors emerged again at regions with high uncertainties.

The N values were chosen within the range $N \in \mathbb{N} \cap [2,\dots,10]$, because sensor numbers greater than 10 were presumed as uneconomical for the studied WDN. Furthermore, placing one sensor is a trivial task—the point with the highest overall sensitivity would be the ideal measurement point[28]. Thus, no optimization algorithm would be necessary.

During all simulations, the network was expected to be a DMA with additional sensors at the inflow point L_I respectively at the outflow L_O. These positions were considered in $f(\mathbf{x})$ evaluations by joining the parameter vector $\mathbf{x}$ of length N with this additional sensor positions

$$\mathbf{x} \leftarrow \mathbf{x} \cap [L_I, L_O] \quad . \tag{4.27}$$

The resulting optimization problem is solved with the GA described in Section 2.1.2.2.1 with following settings: $\mu = 100$, $N_I = 100$, single point crossover operator with $p_r = 0.8$, uniform integer mutation with $p_m = 0.2$, tournament selection of size $k = 3$. For each possible N-ω combination, the optimization problem is solved 10 times to additionally analyze the stochastic behavior of the GA's solutions. As already discussed in Section 2.1.2.2, heuristic algorithms cannot guarantee a global optimal solution and may result in different $f(\mathbf{x}^*)$ for same N and ω parameters.

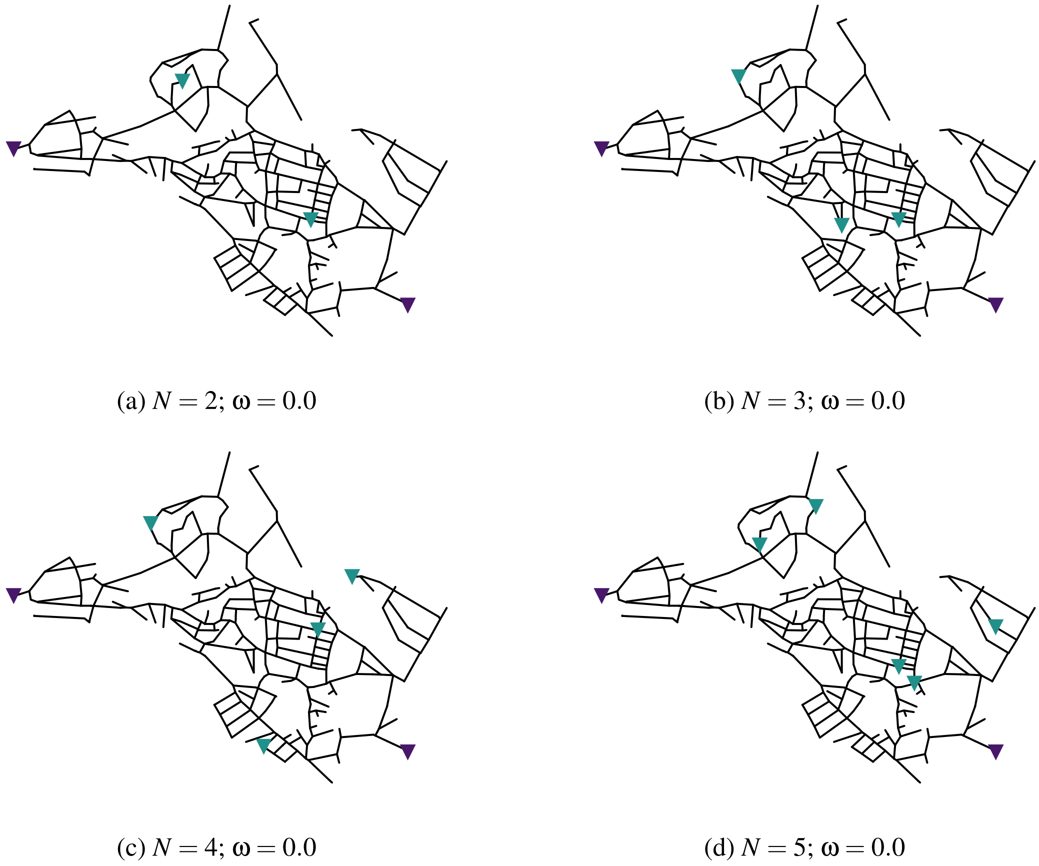
The results for all optimization runs are summarized in Table 4.2 containing the average (mean), minimum (min) and standard deviation (std) of $f(\mathbf{x}^*)$. As expected, $f(\mathbf{x}^*)$ decreases with higher N—$f(\mathbf{x}^*)$ increases with higher ω.

[28] ...unless uncertainty effects are too high there.

Table 4.2.: Results for SPUDU for different numbers of sensors N and different ω parameters. The statistics (minimum (min), mean and standard deviation (std)) of the $f(\mathbf{x})$ values are obtained from 10 GA optimization runs for each N-ω combination (taken from Steffelbauer and Fuchs-Hanusch 2016a).

	$\omega = 0.0$			$\omega = 0.125$			$\omega = 0.25$			$\omega = 0.5$			$\omega = 1.0$		
N	mean	min	std	mean	min	std	mean	min	std	mean	min	std	mean	min	std
2	0.327	0.316	0.007	0.345	0.334	0.007	0.399	0.378	0.009	0.467	0.459	0.005	0.559	0.546	0.008
3	0.265	0.247	0.010	0.284	0.270	0.009	0.312	0.301	0.009	0.372	0.355	0.010	0.444	0.423	0.014
4	0.227	0.207	0.011	0.233	0.222	0.008	0.271	0.260	0.009	0.316	0.304	0.008	0.385	0.367	0.016
5	0.203	0.189	0.011	0.211	0.196	0.008	0.237	0.219	0.011	0.275	0.263	0.011	0.338	0.314	0.012
6	0.185	0.171	0.006	0.191	0.173	0.011	0.212	0.202	0.008	0.249	0.230	0.010	0.312	0.291	0.012
7	0.175	0.158	0.008	0.172	0.156	0.013	0.188	0.176	0.009	0.228	0.202	0.014	0.291	0.268	0.013
8	0.162	0.145	0.008	0.160	0.151	0.009	0.173	0.161	0.009	0.207	0.191	0.011	0.261	0.230	0.019
9	0.144	0.130	0.007	0.150	0.140	0.005	0.161	0.133	0.011	0.189	0.179	0.007	0.249	0.222	0.016
10	0.143	0.135	0.006	0.140	0.128	0.010	0.156	0.140	0.012	0.181	0.156	0.015	0.241	0.207	0.025

Figures 4.3 and 4.4 show the ideal measurement locations with minimum found $f(\mathbf{x})$ values for selected N and ω. Results of all optimal placements of the N-ω pairs can be found in the supplementary material of Steffelbauer and Fuchs-Hanusch 2016a. Figure 4.3 shows results for $N = 2$ to $N = 5$ without uncertainty ($\omega = 0.0$). Results for $N = 5$ under incorporation of uncertainties with varying ω (between 0.125 and 1.0) can be found in Figure 4.4.



(a) $N = 2$; $\omega = 0.0$

(b) $N = 3$; $\omega = 0.0$

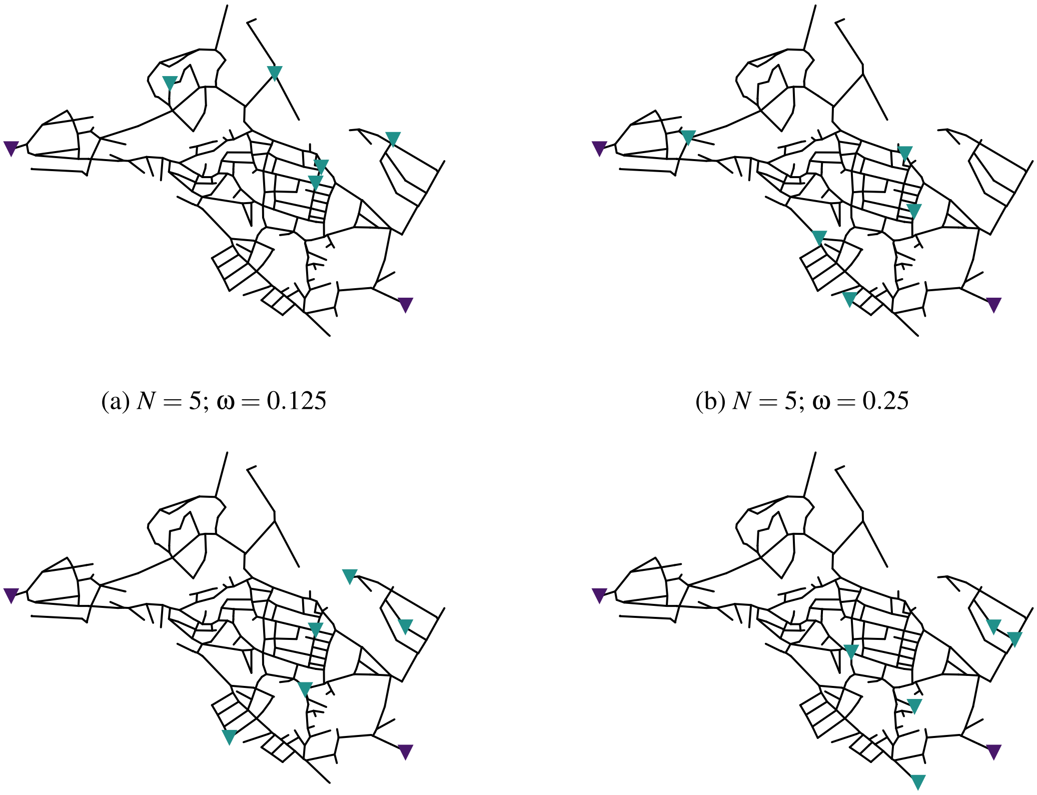
(c) $N = 4$; $\omega = 0.0$

(d) $N = 5$; $\omega = 0.0$

Figure 4.3.: Sensor positions (green triangles) resulting from the enhanced sensor placement algorithm described in section 4.2.2.4 for different number of sensors N and without uncertainties $\omega = 0.0$. Already installed sensors at the inflow and outflow point of the DMA are depicted as purple triangles. (taken from Steffelbauer and Fuchs-Hanusch 2016a)

Discussion It can be seen in Figures 4.3a to 4.3d that without uncertainties the SPUDU algorithm places sensors in regions with high uncertainties. The high uncertainty region can clearly be seen in Figure 4.2b. Incorporating uncertainties with $\omega \neq 0$ leads to different positions. More sensors are moved to low-uncertainty regions (see Figures 4.3d to 4.4d) with increasing ω. Thus, the SPUDU method results in the desired behavior—it punishes measurement locations with high uncertainties. Although, this goes hand in hand at the costs of the spreading of the sensors over the whole WDS. That is why sensors tend to cluster in low uncertainty regions.

Another interesting point is revealed in Figures 4.3a to 4.3d. The optimal positions $\mathcal{P}(N-1)$ for $N-1$ sensors is not a subset of the optimal locations $\mathcal{P}(N)$ for N sensors ($\mathcal{P}(N-1) \not\subseteq \mathcal{P}(N)$). Consequently,



(a) $N = 5$; $\omega = 0.125$

(b) $N = 5$; $\omega = 0.25$

(c) $N = 5$; $\omega = 0.5$

(d) $N = 5$; $\omega = 1.0$

Figure 4.4.: Sensor positions (green triangles) resulting from the enhanced sensor placement algorithm described in section 4.2.2.4 for different five sensors $N = 5$ and different weighting factors ω. Already installed sensors at the inflow and outflow point of the DMA are depicted as purple triangles. (taken from Steffelbauer and Fuchs-Hanusch 2016a)

greedy sensor placement algorithm will fail. Similar results have been found by Kapelan et al. 2005 for model calibration sampling design.

A small note on the size of the problem and on the computation time: Placing 10 sensors in the Linz network consisting of 392 nodes results in $\mathcal{C} \approx 2 \cdot 10^{19}$ possible sensor combinations—in words 20 quintillion possible solutions to the problem. Solving this problem with brute force by total enumeration and assuming that 50 objective function evaluations are possible each second—a reasonable assumption—the computations take exactly as long as the current age of the known universe. If the computation started right after the Big Bang 13.7 billion years ago, the city of Linz can be sure by now that they have the best possible sensor placement for ten sensors in their small subsystem in Linz-Pichling. Provided that the hydraulic model is accurate and the OSP method results in the truly optimal solutions, of course. Thus, it has to be noted that for a large N, the results in table 4.2 must be treated with caution, because of the incredible large search space. In one optimization run, $f(\mathbf{x})$ is approximately evaluated 8000 times which is certainly just a tiny fraction of the total solution space.

4.3.2. Cost-benefit analysis

This section uses the methodology described in Section 4.2.3 to find the cost-benefit behavior behind the OSP problem. The methodology is tested again in the Linz network analyzing the results obtained in Section 4.3.1.3.

The mean values of the 10 optimization runs were chosen for finding the behavior of $f(\mathbf{x})$ on N for different ω. Subsequently, the values were fitted utilizing different functions. The model fits are then tested with GoF statistics to obtain the best mathematical law describing the cost-benefit behavior.

Seven different fit-functions $f_i(N)$ describing the decreasing dependency of the sensor placement quality on N can be found in Table 4.3 in the first column. Further columns are described as follows: N_p is the number of parameters, ν the number of degrees of freedom. χ^2 is the value resulting from χ^2 statistics calculated by equation 2.87, χ^2_r is the value from the reduced χ^2-statistic (equation 2.88), AIC is calculated using equation 2.89 and BIC by equation 2.90. All values are means over the different ω. According to table 4.3, the extended power law function $f_6(N)$ leads to the best χ^2 and χ^2_r. In contrary, the simple power law equation $f_7(N)$ leads to the best AIC and BIC values. Additionally,

Table 4.3.: Goodness-of-Fit table for different fit-functions $f_i(N)$ containing the number of parameters N_P for every function, the degrees of freedom ν, the resulting values for χ^2 and reduced χ^2_r statistics, AIC and BIC (taken from Steffelbauer and Fuchs-Hanusch 2016a)

fit-functions	N_P	ν	χ^2	χ^2_r	*aic*	*bic*
$f_1(N) = a\,e^{-b\,N}$	2	7	0.00272	0.00039	-67.68	-67.28
$f_2(N) = a\,e^{-b\,N} + c$	3	6	0.01346	0.00224	-67.80	-67.21
$f_3(N) = e^{-a\,N} + b$	2	7	0.00355	0.00051	-64.37	-63.98
$f_4(N) = a\,N^2 + b\,N + c$	3	6	0.00082	0.00014	-74.96	-74.37
$f_5(N) = e^{-a\,N} + b\,N + c$	3	6	0.00031	0.00005	-83.12	-82.53
$f_6(N) = (a + b\,N)^{-c}$	3	6	**0.00006**	**0.00001**	-97.65	-97.05
$f_7(N) = a\,N^{-b}$	2	7	0.00009	0.00001	**-97.93**	**-97.53**

Table 4.4 shows the fit parameters and the errors for functions $f_6(N)$ and $f_7(N)$. Due to the high standard errors of parameters a and b resulting for $f_6(N)$, $f_7(N)$—with smaller parameter errors—is

chosen as a cost-benefit function for the values in Figure 4.5. Hence, the SPUDU placement for the OSP problem is found to follow a power law.

Furthermore, this behavior has practical consequences for WUs explained now in more detail—for $\omega = 0.0$ with the values taken from Table 4.2.: Three instead of two sensors improves $f(\mathbf{x})$ by

$$f_{N=2}^{\omega=0}(\mathbf{x}) - f_{N=3}^{\omega=0}(\mathbf{x}) = 0.327 - 0.265 \approx 0.06 \quad . \tag{4.28}$$

The same improvement for starting from three sensors doubles the number of additional sensors. Additional two sensors have to be placed—resulting in five total sensors—to obtain the same benefit

$$f_{N=3}^{\omega=0}(\mathbf{x}) - f_{N=5}^{\omega=0}(\mathbf{x}) = 0.265 - 0.203 \approx 0.06 \quad . \tag{4.29}$$

Starting from five sensors, four additional sensors, again double as many as before, have to be placed additionally for the same improvement

$$f_{N=5}^{\omega=0}(\mathbf{x}) - f_{N=9}^{\omega=0}(\mathbf{x}) = 0.203 - 0.144 \approx 0.06 \quad . \tag{4.30}$$

Therefore, the additional sensor number has to be doubled each time for linearly improving sensor placement's quality .

Remarkably, the power-law behavior stays true when incorporating uncertainties. The difference to simulations $\omega = 0.0$ is that the $f(\mathbf{x})$ for a specific N gets worse if ω increases. This is logically, since leaks are harder to find in real-world systems fraught with high uncertainties compared to theoretically perfect knowledge in simulated WDS. For achieving the same $f(\mathbf{x})$ as in simulations with $\omega = 0.0$ for $N = 2$ sensors, double as many sensors ($N = 4$) are needed for $\omega = 0.5$. Even $N = 6$ is needed if considering the full amount of uncertainties ($\omega = 1.0$). All of that can be seen in Figure 4.5.

Table 4.4.: Resulting fit parameters and error limits for fit-function $f_6(N)$ and $f_7(N)$ from table 4.3 (taken from Steffelbauer and Fuchs-Hanusch 2016a)

	$f_6(N) = (a + b \cdot N)^{-c}$			$f_7(N) = a \cdot N^{-b}$	
ω	a	b	c	a	b
0.000	0.0 ± 1.0	4.0 ± 1.0	0.52 ± 0.05	0.467 ± 0.006	0.517 ± 0.009
0.125	1.0 ± 0.5	2.2 ± 0.6	0.62 ± 0.05	0.512 ± 0.007	0.557 ± 0.010
0.250	0.7 ± 0.4	1.7 ± 0.4	0.66 ± 0.05	0.603 ± 0.008	0.591 ± 0.009
0.500	0.6 ± 0.2	1.3 ± 0.2	0.66 ± 0.03	0.705 ± 0.007	0.586 ± 0.007
1.000	-0.5 ± 0.5	1.9 ± 0.3	0.49 ± 0.03	0.804 ± 0.009	0.531 ± 0.007

4.3.3. Comparison of different sensor placement algorithms

For computing the OSPs, the hydraulic model Ragnitz-Detailed was used (see Section 2.4.3). At the moment when the computations were performed, two hydraulic models existed of the Ragnitz network: (i) the Ragnitz-Simple and (ii) the Ragnitz-Detailed network. The Ragnitz-Complicated network was generated later on when sensors were already installed in the WDS during the real-world case study. The Ragnitz-Simple model was adjudged to be too basic for comparing OSP approaches, thus, the Ragnitz-Detailed model was chosen for obtaining the ideal positions for the different OSP

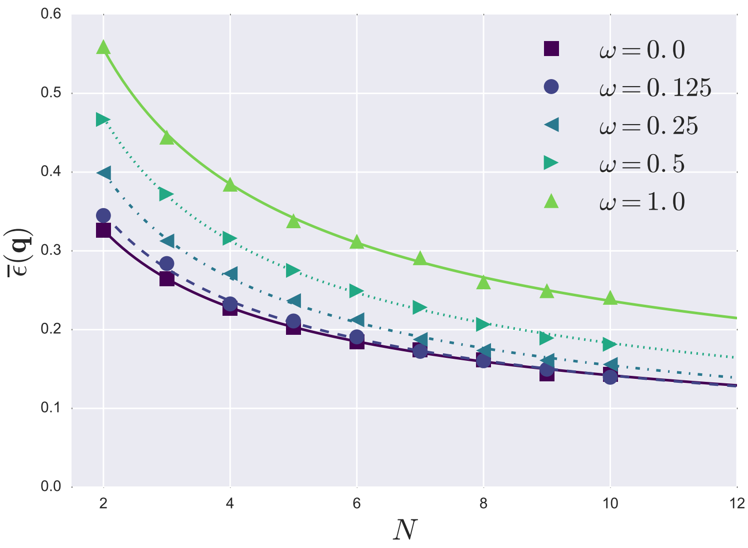


Figure 4.5.: Cost-benefit functions of OSP for different ω (taken from Steffelbauer and Fuchs-Hanusch 2016a)

algorithms. The roughness values of the Ragnitz-Detailed model were calibrated manually a year before the field study, hence realistic hydraulic simulation values were expected by the model.

Since the pressure sensors were planned to be installed at hydrants, only the 35 hydrants in the Ragnitz-Detailed network ($M = 35$) were chosen as possible measurement positions (see Figure 2.21). Five sensors were planned to be installed for each OSP method ($N = 5$). This decreased the combinations in the solution space significantly to

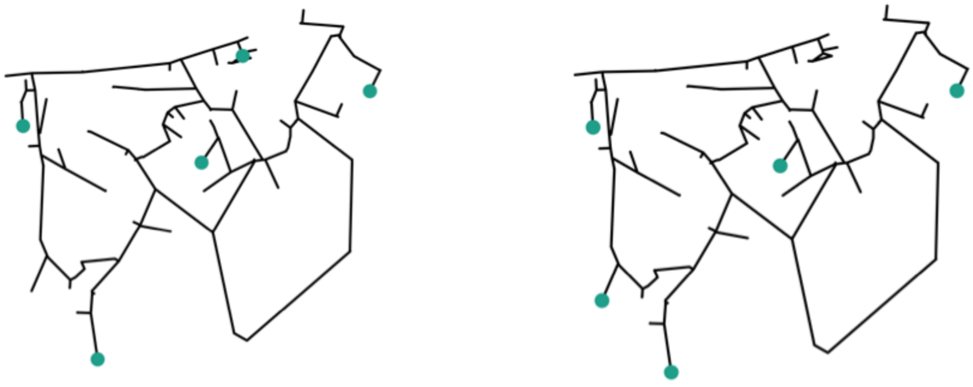
$$C = \binom{M}{N} = \binom{35}{5} = 324632 \quad . \tag{4.31}$$

Computations with the graph-based sensor placement algorithms can be found in Figure 4.6. Figure 4.6a shows the results computed with the Shortest Path 1 algorithm, while Figure 4.6b shows results for Shortest Path 2. Note that the Ragnitz-Detailed network was used for calculating the ideal measurement locations —the Ragnitz-Simple network is only used for a simpler representation of the sensor positions in the figures. Additionally, the rankings introduced by the shortest path algorithms can be found in Table 4.5. Both algorithms identify HG4339b as the most important node since both algorithms start with the same shortest path calculations. Then the ranks of the measurement positions differ according to the procedures described in Section 4.2.2.2.1 and 4.2.2.2.2. Noticeably, although the rankings of the measurement locations are quite different, the total measurement positions between the two algorithms just differ in one location. Shortest Path 1 chooses HG3933, whereas Shortest Path 2 selects hydrant HG4576.

The sensitivity matrices S for the sensitivity-based OSP methods (see Section 4.2.2.3) were computed with a leak introduced with an emitter coefficient of $c_e = 0.1$ and an emitter exponent $e_e = 0.5$ using the leak power law (in equation (2.63)). All nodes in the system served as possible leak positions resulting in $m = 1300$ leak scenarios. The leak magnitudes varied between $Q_L = 0.55$ and $Q_L = 1.0$ L/s depending on the leak node's pressure. The residual matrix R used in the Casillas (Section 4.2.2.3.3) and the SPUDU (Section 4.2.2.4) algorithm has been calculated with the same c_e values.

Table 4.5.: This table shows the rankings of the measurement positions introduced by both shortest path algorithms—Shortest Path 1 (SP1) respectively Shortest Path 2 (SP2)

Rank	**SP1**	**SP2**
1	HG4339b	HG4339b
2	HG4150	HG4744
3	HG4744	HG4150
4	HG3445	HG4576
5	HG3933	HG3445



(a) Shortest Path 1

(b) Shortest Path 2

Figure 4.6.: Results for graph-based OSP for $N = 5$

The uncertainties for the SPUDU method are again simulated with MCS. The demands are perturbed with a Gaussian-distribution function with the mean base demand and a standard deviation $\sigma = 10\%$ of its value $\mathcal{N}(\mu_q, \sigma_q = 0.1\mu_q)$. Due to the small nodal demands in the system resulting from the very detailed model, an uncertainty weight of $\omega = 1.0$ is chosen taking the full strength of the small uncertainties into account. All results for the sensitivity-based OSP methodologies are depicted in Figure 4.7.

The results for Shannon entropy based OSP method as described in Section 4.2.2.3.1 can be found in Figure 4.7a. Interestingly, this method also places a sensor in the low-pressure part of the WDS in the South, where the network is not very sensitivity to pressure changes due to leaks. Obviously, this sensor position is favored because it maximizes the entropy in the system—the second objective F_2 in the algorithm—forcing a spreading of the sensors over the whole network.

For the Pérez OSP approach, S has to be binarized. An ideal binarization threshold of $\tau = 0.279$ is found with the method described in Section 4.2.2.3.2. This algorithm defines the optimal measurement positions as depicted in Figure 4.7b. The sensors tend to cluster in a region to the South-West that is also one of the most sensitive regions of the system.

Casillas OSP method—described in Section 4.2.2.3.3—results in the positions shown in Figure 4.7c and differs from the SPUDU placement shown in Figure 4.7d just in one measurement location—HG3835 instead of HG4162. Noticeably, both locations are only chosen by these two algorithms.

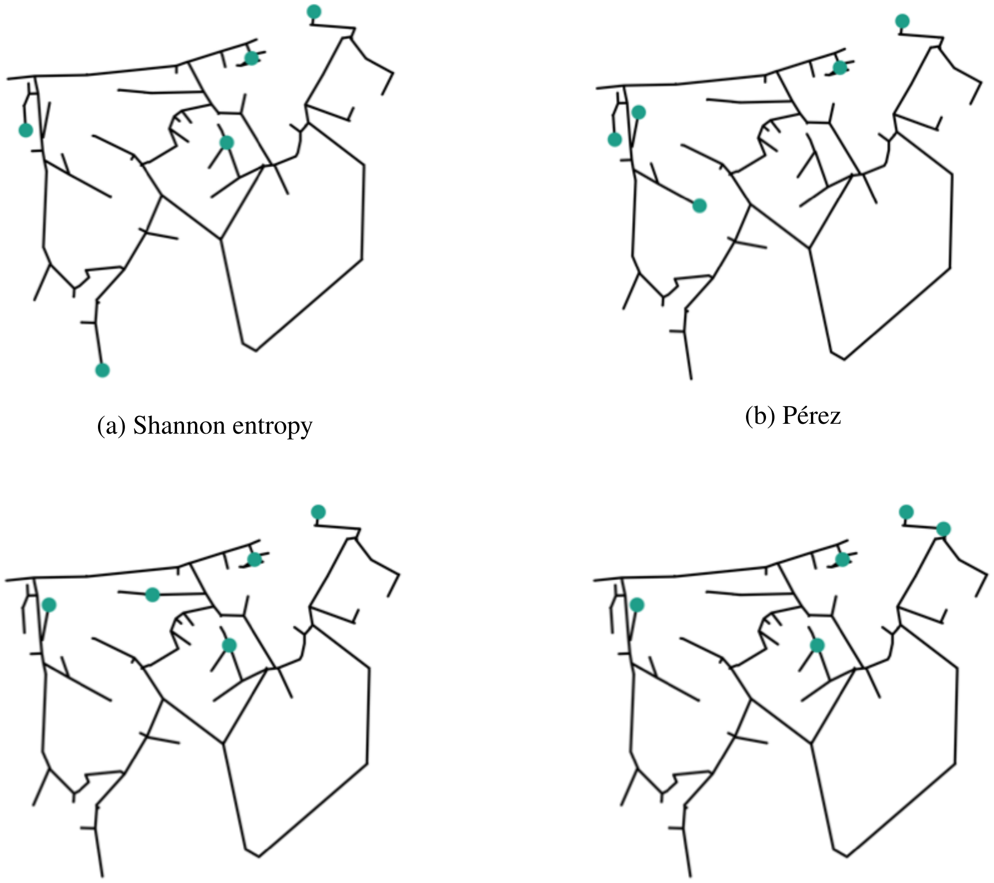
Table 4.6 summarizes the sensor positions for the different OSP methods. Additionally, the membership of each of the twelve sensor positions to the OSP algorithms is depicted in the last column. This membership is also shown in Figure 4.8. It seems that the most important hydrants in the system are HG3933 followed by HG4215 and HG4339b since they were chosen by most of the OSP algorithms to be ideal—five respectively four out of six.

Comparing all OSP results in Figure 4.6 and Figure 4.7, graph-based OSP tend to put sensors all over the network favoring a spreading of the sensors, whereas sensitivity-based approaches tend to place sensors in regions where high pressure responses due to leaks are expected—at the high pressure regions in the North of the network (for example, see Figure 2.20). Only the Shannon entropy approach placed a sensor at the South, probably for maximizing the entropy of the placement.

Additionally, the different OSP approaches are compared using different analysis criteria—most of them can be found in Schaetzen et al. 2000. The first three criteria are linked to the coverage in terms of total covered water paths of the network's pipes. This path is assumed to be the unification of all shortest paths from the source to the sensors. The first criterion—pipe-coverage (PC)—is the number of pipes covered by a certain OSP N_{PC}. The second criterion is the sum of the length of these pipes L_{PC} in meter. The third criterion is the relative percentage of this length to the total length of the network R_{PC}. These three criteria can be found in Schaetzen et al. 2000. The higher the coverage, the better a sensor placement is spread over a whole network. The later four criteria are based on the sensitivity of the OSP. The fourth criterion measures the average

$$S_{MO} = \frac{1}{N} \sum_{j \in \mathbf{x}} (\overline{s})_j \tag{4.32}$$

over all sensors of the overall sensitivity of the measurement point (see equation (4.10) for the definition of $\overline{s_j}$). A high S_{MO} is preferred. The last three criteria are the objective functions of the Shannon entropy OSP. F_1 states the maximal sensitivity again to be preferred high (see equation (4.13)). F_2



(a) Shannon entropy

(b) Pérez

(c) Casillas

(d) SPUDU

Figure 4.7.: Results for sensitivity-based OSP for $N = 5$

Table 4.6.: This table summarizes the sensor positions at the hydrants in the Ragnitz network for the different OSP methods placing five sensors for each algorithm (see the sum in the last row): Shortest Path 1 (SP1), Shortest Path 2 (SP2), Shannon entropy (SHE), binarization method of Pérez (PER), projection-based method of Casillas (CAS) and the SPUDU (SPU) method. Additionally, the membership of each of the twelve measurement position to the OSP algorithms is depicted in the last column.

Loc. / OSP	**SP1**	**SP2**	**SHE**	**PER**	**CAS**	**SPU**	$\sum$
HG3420			X		X	X	3
HG3445	X	X					2
HG3835					X		1
HG3933	X		X	X	X	X	5
HG4150	X	X					2
HG4162						X	1
HG4215			X	X	X	X	4
HG4339b	X	X	X	X			4
HG4383				X	X	X	3
HG4540				X			1
HG4576		X					1
HG4744	X	X	X				3
$\sum$	5	5	5	5	5	5	30

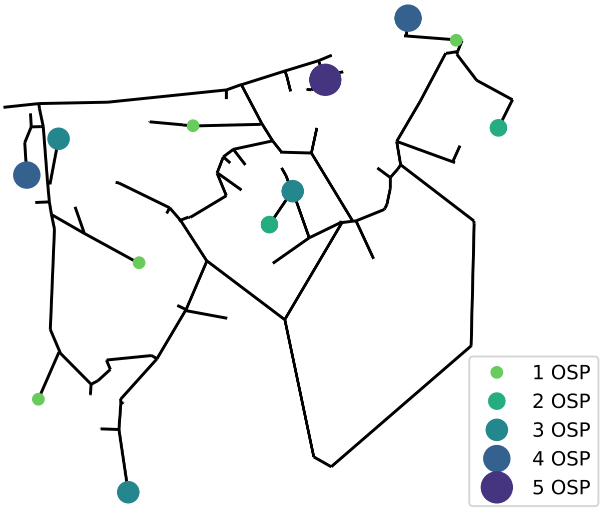


Figure 4.8.: This figure shows the sensor positions calculated for the Graz-Ragnitz network together with the membership—the number of how many OSP algorithms have chosen the measurement positions

Table 4.7.: This table shows different analysis criteria for the different OSP algorithms. The criteria are described in the text. The best performing OSP for each criterion are marked with boldface. Shortest Path 1 (SP1), Shortest Path 2 (SP2), Shannon entropy (SHE), binarization method of Pérez (PER), projection-based method of Casillas (CAS) and the SPUDU (SPU) method

OSP	N_{PC}	L_{PC} [m]	R_{PC} [%]	S_{MO}	F_1	F_2	F
SP1	240	4323	26.7	0.18	**362.07**	7.079	**0.6007**
SP2	253	4553	28.1	0.17	329.12	**7.107**	0.6104
SHE	234	4159	25.7	0.18	361.06	7.080	0.6010
PER	**280**	**4761**	**29.4**	**0.20**	354.03	7.061	0.6031
CAS	247	4206	25.9	0.18	353.25	7.078	0.6033
SPU	234	4044	24.9	0.18	347.65	7.078	0.6049

is the maximum entropy criterion (see equation (4.14)). And the final criterion is the trade-off between F_1 and F_2 which has to be minimized (see equation (4.15)). All resulting values for the before mentioned criteria can be found in Table 4.7.

The highest coverage of the network—N_P, L_{PC} as well as R_{PC}—is accomplished by the algorithm of Pérez. Additionally, this algorithm resulted in the most sensitive measurement points in the system. This maybe arises from the normalization prior to binarization while using S. Although, the coverage is the highest, the entropy is the lowest—possibly resulting from the clustering in the North-West part of the network. Additionally, the high mean sensitivity value cannot hide that the maximum leak sensitivity given by F_1 is mediocre. The worst F_1 value is given by Shortest Path 2 algorithm, although the coverage is the second highest.

Obviously, the optimization algorithm has fallen into a local optimum at finding the ideal positions for the Shannon entropy algorithm, because there exist a solution with a better fitness value F—namely Shortest Path 1. Although, the objective function values $f(\mathbf{x})$ are very close to each other ($F_{SP1} = 0.6007$ compared to $F_{SHE} = 0.6010$). In fact, the Shannon entropy OSP never scored best in any categories, besides the fact that optimizing F_1, F_2 and F are objectives of the algorithms optimization problem. Shortest Path 1 lead to better F_1 values, while Shortest Path 2 resulted in better F_2. Nevertheless, both objective values are very imbalanced for these two OSPs. High F_1 lead to low F_2 and vice versa. The result of the Shannon entropy function—scoring second in both criteria—is more balanced, favoring a sensitive as well as a widespread placement.

SPUDU favors places with low uncertainties. Therefore, this algorithm led to the worst coverage. This can be already expected while looking at Figure 4.7d since the placement resulted in two sensors very close to each other in the North-East region.

Looking at all values in Table 4.7, the differences of the different OSP approaches are not significant and if a OSP performs worse in one criterion, it makes up for it in other criteria. Hence, for real-world tests of the OSP algorithms in the next chapter, no essential performance distinctions are expected between them. This may result from the constraint on placing sensors only at hydrants permitting all other possible measurement positions in the system. Another reason might be that the network is highly branched—only four loops exist. It is possible that the OSP lead to more differing OSP analysis criteria values and measurement locations in networks containing more loops.

After the theoretical computation of the sensor positions, pressure sensors were installed in the real-world networks at the resulting twelve hydrants to test the effect of OSP on model-based leak localization performance. This is described in the next chapter.

4.4. Conclusion

This chapter tried to cover OSP techniques for model-based leak localization in WDS, a scientific topic which is not yet researched exhaustively. Three research questions also stated in the introduction of this thesis were tried to be answered throughout this chapter:

Q.2.1 Which optimal sensor placement algorithm finds the best pressure sensor positions for model-based leakage localization?

Q.2.2 How to incorporate different sources of uncertainties in sensor placement algorithms to obtain more robust optimal measurement locations?

Q.2.3 How many sensors are needed for model-based leak localization?

This questions and belonging findings are now tried to be answered in reverse order in this conclusion section.

How many sensors are needed for model-based leak localization? This question is answered by an elaborate cost benefit analysis of the OSP problem. A methodology for finding the cost-benefit behavior of OSP algorithms was shown in Section 4.2.3 on a specific OSP algorithm. In principal, every OSP algorithm can be analyzed with this methodology. The OSP chosen in this thesis—besides being developed intentionally during the work for this thesis—it is practical since the objective function is directly related to the number of leaks located directly. But it is also assumed that other OSP algorithms and their objective function values—or analysis criteria as defined in Section 4.3.3—will show qualitatively the same behavior. Logically, every OSP objective function has to improve when adding more sensors. Otherwise it won't be a good representation of the real-world.

It was shown in Section 4.2.3 that the sensor quality on finding leaks follows a power law behavior on the sensor number—at least in the examined network and for the suggested OSP method. Studies from other research groups applying other OSP methodologies on different networks and depicting the fitness function in dependency of N showed similar behavior (see for example Casillas, Puig, et al. 2013; Pérez, Puig, Pascual, Peralta, et al. 2009). Nevertheless, the work in this thesis is the first time that a mathematical law behind this behavior is tried to be revealed. Furthermore, applying the methodology on two slightly bigger networks in the city of Linz during projects showed that the sensor placement also followed a power law, but is not shown in this thesis for the sake of brevity and confidentiality. The power also comes with practical considerations for WU as doubling the number of additional sensors for a linear improvement on the quality.

The question of how many sensors are needed is answered by applying the proposed cost-benefit methodology. The number of sensors can be calculated with the help of the inverse function $f^{-1}(N)$ if a WU knows the desired localization quality they want to reach. For the power law equation, the sensor number is given through the inverse function $f^{-1}(N)$

$$N = f^{-1} \circ f(N) \quad \rightarrow \quad N = \left(\frac{f(N)}{a} \right)^{b} \quad , \tag{4.33}$$

where $f(N)$ is the desired sensor placement quality.

The cost-benefit analysis provides also shortcuts on the computational burden of the OSP problem. For large systems, for example, the OSP problem can be solved for $N = 1$ to $N = 3$ or $N = 4$, where the OSP problem is still resolvable in reasonable time. Then the resulting $f(\mathbf{x})$ values can be used to fit the cost-benefit function parameters and the performance of a OSP on the sensor number can be extrapolated to give a WU a basis for decision how many sensors lead to what performance. If the WU decided on a certain sensor number, the problem has solely be optimized for that N without the need to simulate every solution in between and subsequently giving the WU the whole cost-benefit curve.

On the way to find the cost-benefit behavior of the OSP problem, another interesting finding emerged. The simulations confirmed that the optimal sensor positions for $N-1$ sensors is not a subset of the optimal locations of N sensors. This was already shown for hydraulic calibration by Kapelan et al. 2003, but to the best of my knowledge, it is the first time that this behavior was actively observed in a OSP study on model-based leak localization. As a consequence, with greedy sensor placement algorithms it is not possible to find optimal measurement locations. Furthermore, solving the sampling design problem with GAs showed good results due to the high efficiency of these algorithms in high dimensional search spaces.

How to incorporate uncertainties in OSP algorithms? A novel OSP method—the SPUDU algorithm—was developed, capable of incorporating uncertainties of any kind. The method is generally applicable for all kind uncertainties (roughness, diameter) and all kind of sensors (flow, quality)—by using the right sensitivity matrix and choosing MCS for diverse model input model output pairs. In this thesis, the methodology was shown on the example of uncertain demands influencing pressure measurement locations for leak localization. Indeed, simulations in this chapter have shown that pressure measurement points which are sensitive on leaks are also points which may be sensitive to uncertain demands. Thus, they are less ideal positions for a robust OSP. The SPUDU algorithm showed that it avoids regions of high uncertainty if they are regarded in the optimization problem, as desired. Without uncertainties, these now omitted points were always chosen by the OSP algorithm.

Incorporating uncertainties in OSP does not change the form of the cost-benefit curve, but lead to higher costs in terms of sensor number for reaching a certain quality. The power law behavior stays still true under incorporation of uncertainties. Hence, the form of the function is robust against uncertainties speaking for its general validity.

However, it is still an open question of what value has to be chosen for ω in the SPUDU algorithm and this thesis lacks in answering it. Potentially, this problem can be solved with TSA on pressure measurement data in the system and by carefully analyzing the statistics of the residuals (see Chapter 3).

Which algorithm finds the best measurement positions for leak localization? Six different OSP algorithms have been applied to the same hydraulic network. Despite every algorithms stating to find the optimal measurement locations, the locations differed for all methods. Logically, the different OSP algorithms result in positions which are favored by their objective. All OSP algorithms have to make compromises between (i) detectability and (ii) isolability. (i) favors sensor placements at leak sensitive points clustered in sensitive areas, (ii) ideally favors placements spread over the system. Furthermore, this two objectives are not clearly defined as there is no unique definition for both criteria, only surrogate measures (similar to e.g. resilience). This can be seen in the application of the analysis criteria at the end of Section 4.3.3. The criteria do not favor a certain OSP in particular. If a method leads to extraordinary results in one criterion, it is potentially bad in another. Hence, the answer to this question has to be postponed to the next chapter when the different OSP algorithms are compared on real-world leak localization scenarios.

Nevertheless, some remarks on the applicability of the OSP algorithms can be given here already.

The limitations of sensitivity-based OSP algorithms are certaintly that they can only be applied to WDSs where a hydraulic model already exists since it is necessary to compute the sensitivity matrix. Additionally, the OSP algorithm is only as good as the model. For example, an unknown closed valve can have strong influence on the hydraulics and therefore the OSP result might not be optimal in reality. Thus, additional effort has to be put into hydraulic model calibration. Additionally, also later on if a leak localization method is applied in reality, effort has to be put in bookkeeping of the boundary conditions of the system (which have a high influence on the OSP) to not loose the optimality of the placement if the OSP method is not robust enough.

Furthermore, for bigger systems the computation time increases for calculating S as well as solving the OSP problem itself—making the method not directly applicable for big real world systems with thousands of nodes. For the SPUDU method additional computational costs arise for computing the uncertainties.

Graph-based OSP methods do not have these disadvantages—at least not in this magnitude considering the computational burden. However, the goodness of a OSP without considering the hydraulics is questionable.

Another limitation of the OSP problem in general is the high computational complexity considering the huge search space size even for small WDNs. Skeletonization or simplification algorithms, surrogate modeling or graph clustering as in Sarrate, Blesa, and Nejjari 2014 are promising remedy by reducing the problem's complexity. Certainly, this is an interesting and rewarding research direction in future.

Chapter 5

Model-Based Leak Localization

“It is a mistake to think you can solve any major problems just with potatoes."

— Douglas Adams, *Life, the Universe and Everything*

5.1. Literature review

Methods for finding the approximate location of a leak by utilizing information obtained from hydraulic measurement devices (flow meters and pressure sensors) alongside hydraulic models of WDS—model-based leak localization methods—have been around for more than a quarter-century. Despite their age, these methods are exclusively found in academic discourse and have not arrived in practice until today.

Generally, two different approaches are used that are distinguished by the hydraulic model complexity—(i) steady-state and (ii) transient techniques (see Section 2.2). Since the focus of this thesis is on steady-state models for model-based leak localization, the literature listed below only covers scientific work related to this technique. For the interested reader, a selective literature review on transient-based techniques can be found in Colombo, Lee, et al. 2009. Furthermore, literature dealing with real-network problems—although covering both simulation and real-world field test— is listed below. For the sake of brevity, publications are not listed that propose methods which are exclusively applied on toy examples which consist of few nodes only.

Model-based leak localization with steady-state hydraulic models was first introduced by Pudar and Liggett 1992. This scientific article was a milestone for leak detection and localization research mainly for two reasons (Colombo, Lee, et al. 2009): First, it was the earliest paper proposing leak detection and localization with pressure sensors. Second, for the very first time, the problem was explicitly formulated as an inverse problem. The objective of Pudar and Liggett 1992 was to minimize the sum of squared differences between simulated and measured pressures with the Levenberg-Marquardt algorithm (see Section 2.1.2.1.4). Additionally, Pudar and Liggett 1992 distinguished between over-, even- and under-determined problems. While the solutions were exact for over- and even-determined problems, exact answers were not possible for the under-determined case, even though useful information could be retrieved. Since the pioneering work of Pudar and Liggett 1992, leak localization by solving inverse model-based approaches has been extensively investigated.

Pérez, Puig, Pascual, Quevedo, et al. 2011, for example, solved the problem of finding leaks by analyzing pressure (and flow) sensitivities. The approach consisted of binarizing the leak sensitivity matrix with a certain threshold resulting in a signature matrix for leaks. Subsequently, the leak's position is found by comparing the binarized residuals—differences between measurements of leak and leak-free scenarios—with the columns of this matrix. The column most similar to the residuals implicates the leak's location in the WDS. The method has been applied to DMAs in simulation and on a real-world case study. The simulations were applied to the Place del Diamant DMA in Barcelona (Spain). The network consisted of 1600 nodes and approximately 41 km of total pipe length. Leaks were simulated as additional demands of 1 L/s, which correspondent to approximately 3 % of the MNF. In total, eight pressure sensor were used to find the leak. Additional simulations have been performed with uncertainties in demand to test the robustness of the method. Good results were obtained without consideration of uncertainties. Increasing the uncertainty lead to poorer results. Furthermore, the results showed that sensors of high accuracy are necessary to locate leaks properly. The real-world experiments have been applied on a real-network with 260 nodes. Three pressure sensors were used and leaks were forced between 1.7 and 5 L/s for 15 minutes in the system. With the proposed method, 31 out of 42 leaks were detected in the right zone, although this zone contained approximately one third of all nodes. The results were astonishing, despite the fact that the few sensors were not accurate nor optimally placed. Further improvements were expected by advancing demand estimations.

The method was improved by Pérez, Quevedo, et al. 2011. Two methodologies were compared—the binarised residuals (from Pérez, Puig, Pascual, Quevedo, et al. 2011) and a correlation method making use of the full information contained in the sensitivity matrix. The methods were tested under different boundary conditions and under uncertainties in the Nova Icaria DMA in Barcelona (Spain) consisting of more than 3000 nodes. Simulations with noise of 0.25 % in the measurements and without noise were compared. The correlation method proved to be more robust under uncertainties. For smaller leak sizes of 1.7 L/s, both methods had great difficulties finding leaks under uncertainties. For different boundary conditions (e.g. demand loads in the network), again, the correlation method was more robust.

Casillas, Garza-Castanon, et al. 2013 extended the proposed method of Pérez, Quevedo, et al. 2011 by making use of EPS. Three different methods where tested, the (i) angle method, (ii) least squares optimization and (iii) correlation method. Simulations where performed using the Nova Icaria DMA in Barcelona (Spain) equipped with six pressure sensors. Leaks were simulated using the pressure dependent emitter equation (see equation (2.63)), the emitter coefficient was chosen to produce an average leak outflow of 1.67 L/s with values ranging from 0.7 to 3 L/s. Additional noise on measurements and demand between 1 % and 5 % was introduced. Simulations showed that the angle method performed best in the presence of noise reaching a localization precision of under 200 m.

Furthermore, Casillas, Garza Castañón, et al. 2014 studied the EPS leak localization in more detail by comparing five different ways of using the leak sensitivity matrix: (i) the binarization approach of Pérez, Puig, Pascual, Quevedo, et al. 2011, (ii) the angle method, (iii) the correlation method, (iv) the Euclidean method and through (v) least square optimization. The methods were tested on two academic networks (Hanoi and Quebra) and on a real-world DMA (Nova Icaria in Barcelona) using real leak data. In the academic network simulations, the leaks were introduced as extra demand with unitary patterns along the time horizon. For the real-world case study, leaks were generated through the leak equation (2.63). The efficiency of the methods was tested with different leak magnitudes, uncertainties in measurements (Gaussian white noise of 2 %) and uncertainties in nodal demands (2 - 4 %). The sensitivity matrix in the Hanoi network was simulated with a large burst of 50 L/s, leak tests were performed with leak sizes ranging between 10 and 80 L/s. The Quebra network simulations were performed with leak size of 0.01 L/s ranging between 0.01 and 0.2 L/s. The simulations in Barcelona were performed with leak size of 3 L/s (range of 0.7 to 6.3 L/s), with 6 and 15 pressure sensors. Data for a real leak—5.6 L/s running for 30 hours—was also used and the real leak was located with five pressure sensors—one was not working—within 100 meter. Overall, the study showed that the optimization method (v) and the angle method (ii) performed best through all tests.

Calibration approaches can also be used to find leaks (e.g. Sanz et al. 2015; Z. Wu et al. 2009). Z. Wu et al. 2009 developed a model-based optimization method using a competent genetic algorithm to calibrate roughness and leakages. Leakages were simulated as pressure-dependent demand utilizing the emitter equation. The study investigated an example from literature (Pudar and Liggett 1992) (5.5 L/S at three nodes) and a real-world system of 15 square kilometers consisting of 122 pipes and 841 nodes with 3000 properties. In the real-world example, 28 pressure loggers were used and leaks were calibrated for a fixed number of 25 locations. The solutions of the optimization problem found leaks close to historical leak data of 22 reported leaks. Remarkably, using this method the authors could find that a leak of 10 L/s—a leak size hardly resolvable with previous methods.

Sanz et al. 2015 developed a novel method utilizing online demand component calibration of geographically distributed demands. Sudden changes in demand components during calibration can be detected by comparing new with previous calibrated parameters. Due to geographic information, a leak can be localized associating an unusual high demand to a certain pattern. The method was tested with real-world data on the Nova Icaria network in Barcelona. The MNF of the measurement zone is 20 L/s with a peak hour demand of 50 L/s. Five pressure sensors were used and three leaks with three different leak sizes (1, 3 and 5 L/s) were artificially generated in the system. The developed leak membership method lead to a localization accuracy of 180 m in all scenarios except one. Due to the small leak size of 1 L/s in a zone with a predominant demand for the system, the algorithm failed to locate this leak, although, it was detected by the algorithm.

A different method was introduced by Poulakis et al. 2003 using a Bayesian identification methodology coupled with exhaustive search algorithm to provide estimates of leak magnitudes and locations. This method was also able to handle unavoidable uncertainties in measurement as well as modeling errors. The method was applied on the WDS described in Section 2.4.1 with leak sizes ranging between 22.8 and 57 L/s and pressure sensors either at all nodes or with seven sensors only. Additionally, the study investigated uncertainties as well as a multiple leak case with two leaks of sizes 114 respectively 44.7 L/s. Without uncertainties, the method was able to exactly find the leak magnitudes and locations. Though, with noise the method was ineffective. If errors were under a certain threshold, the leak's location and the outflow was defined correctly. The threshold values depended on the characteristics of the inverstigated WDS, the leak's location and magnitude, as well as the number, location and type (flow, pressure) of the sensors.

Shinozuka et al. 2005 experimented with the same network as Poulakis et al. 2003. Yet, they used pressure sensors combined with ANN to locate large bursts after earthquakes in pipe systems. Bursts were defined through the affected cross section area of the pipe ranging from 1 to 100 %. The ANN was trained through EPANET simulations (Rossman 2000) on just one burst location. Three pressure sensors were used and the simulations were validated by using data sets that had never been used before.

Model falsification diagnosis for leak localization with UFMs is used by Goulet et al. 2013. Simulations were performed in a case study area in a DMA in Lausanne with approximately 300 pipes and a MNF of 14 L/s. Leaks were monitored during the MNF, because the customer demand uncertainties were minimal this time. Uncertainties in pipe elevation, diameter, minor loss, roughness, demand

and sensors resolution of the 14 flow meters were investigated. Leak scenarios were generated with EPANET (Rossman 2000) with outflows between 0.4 and 1.7 L/s. Identification of the right leak scenarios with high certainty levels was only possible for leaks greater than 1.25 L/s; the largest leaks were localized within a radius of 500 meters.

Izquierdo et al. 2007 used a combination of (i) deterministic and (ii) machine learning model components. (i) Hydraulic EPANET models were used to retrieve fuzzy state estimates together with a description of the anomalies responsible for the network faults. (ii) The machine learning component consisted of neuro-fuzzy ANNs. The study used small-scale simulations on a small system with 20 nodes and 32 pipes. The ANN approach correctly assigned leaks greater than 10 L/s within two nodes and leaks greater than 6 L/s within three nodes. Leaks under one liter per second were not classified correctly.

In the study of Mashford et al. 2012, information on the leak's size and location was obtained with the help of SVMs by analyzing data from pressure and flow measurement devices. The study consisted of simulations exclusively using EPANET. Leaks between 0 and 3 L/s were produced and an accuracy of locating leaks within 100 meters of 77 % was achieved with six pressure sensors in a system of a total pipe length of approximately three kilometers.

All studies above—also the methods using data driven approaches—utilized hydraulic models, be it only for the generation of training scenarios. Romano et al. 2013 was the first to the best of my knowledge who introduced a localization technique that operates without using any hydraulic models at all. Herein, the approximate location of pipe burst events is retrieved by making use of a multivariate Gaussian mixture-based graphical model and geostatistical techniques. It was found that ordinary co-kriging technique was best for this task. The main advantage of this method over model-based methods is that the introduced artificial intelligence techniques require much less frequent measurements of flow and pressure than transient methods. They also rely on the empirical observations only without demanding precise knowledge on network parameters. The methodology was tested by opening hydrants in a real-life DMA with 17.8 km of pipes and 925 customer connections. Only pressure sensors (altogether 13 measurement devices) were used to locate leaks. The bursts were generated at five different locations with outflows ranging between 5 and 18.3 L/s corresponding to 51 % to 186 % of the average DMA inflow. The novel data-driven methodology was able to successfully locate the simulated pipe burst events at the different locations within the studied DMA. Localization accuracy depended on the number of pressure sensors—the more the better—as well as their spatial layout within the studied DMA.

As outlined in this section and in Table 5.1, many distinct model-based leak localization methods exist in literature based on various different mathematical techniques and principles. However, none crystallizes to be ideal in finding leaks over others. Furthermore, their practical application on real-world systems by water companies is still questionable (Romano et al. 2013). Either, the techniques demand a high number of sensors (>20) at doubtful locations for quite small DMAs, or the leaks, that are potentially located, are in fact huge pipe bursts (80 L/s!). Such incidents would immediately flood huge areas in cities and, hence, would obviously not have to be located anymore.

Table 5.1.: Literature review on model-based leak localization methods. Abbreviations for methods: Binarized Residuals (BIN), angle method (ANG), corrlation method (CORR), Euclidean method (EUC), least squares optimization (LSO), Genetic Algorithm (GA), Extended Period Simulation (EPS), Artifical Neural Networks (ANN), Support Vector Machines (SVM). In the networks columns, S stands for simulation, R stands for real-world field test and n_n is the node number. Leak column: c_e are leak outflow simulated with emitter equation, q_D stands for additional demand. In the sensor column the number of sensors is listed, (p) stands for pressure and (Q) stands for flow sensor.

Literature	**Method**	**Networks**	**Leak (L/s)**	**Sensors**
Pudar and Liggett 1992	sum of squared differences	S: Small Network $n_n = 7$ S: Larger Network $n_n = 20$	c_e : >50 (1 to 3 leaks) c_e : > 7 (1 to 9 leaks)	3 & 7 × (p) 10 & 19 × (p)
Pérez, Puig, Pascual, Quevedo, et al. 2011	BIN	S: Placa del Diamant $n_n = 1600$ R: $n_n = 260$	q_D: 1.0 c_e : 1.7 - 5.0	8 × (p) 3 × (p)
Pérez, Quevedo, et al. 2011	(i) BIN, (ii) CORR	S: Nova Icaria $n_n = 3320$	q_D: 1.7 - 6.3	15 × (p)
Casillas, Garza-Castanon, et al. 2013	EPS with (i) ANG, (ii) LSO, (iii) CORR	S: Nova Icaria $n_n = 3320$	c_e : 1.67 [0.7-3.0]	6 × (p)
Casillas, Garza Castañón, et al. 2014	EPS with (i) BIN, (ii) ANG, (iii) CORR, (iv) EUC, (v) LSO	S: Hanoi $n_n = 31$ S: Quebra $n_n = 55$ S: Nova Icaria $n_n = 3320$ R: Nove Icaria $n_n = 3320$	q_D: 50 [10.80] q_D: 0.01[0.01-0.2] c_e : 3.0 [0.7-6.3] c_e : 5.6	31 × (p) 55 × (p) 6 & 15 × (p) 5 × (p)
Z. Wu et al. 2009	Competent GA	S: Small Network $n_n = 7$ R: $n_n = 841$	c_e : 5.5 (3 leaks) c_e : 10 (25 leaks)	2 & 3 × (p) 28 × (p)
Sanz Estapé 2016	demand component calibration	R: Nova Icaria	q_D: 1,3,5 (3 positions)	5 × (p)
Poulakis et al. 2003	Bayesian identification	S: Poulakis	q_D: 22.8-114 (1-2 leaks)	7 & 31 × (p)
Shinozuka et al. 2005	ANN	S: Poulakis	c_e : very large bursts	3 × (p)
Goulet et al. 2013	model falsification diagnosis	S: Lausanne	q_D: 0.4-1.7	14 × (Q)
Izquierdo et al. 2007	neuro-fuzzy ANN	S: small system $n_n = 20$	q_D: 1-10	8 × (p) & 7 × (Q)
Mashford et al. 2012	SVM	S: ≈3 km	c_e : 0-3	3 × (p)
Romano et al. 2013	Gaussian mixture-based graphical models with geospatial techniques	R: 17.8 km and 925 customer	5.0-18.3 (5 positions)	6 & 13 × (p)

The aim of this chapter is to develop an inverse method which is both: based on static hydraulic simulations and able to locate small leaks compared to the examples above with fewer pressure sensors. Of course, in the case of leak localization by static methods concerning all the uncertainties in hydraulic network models, an inverse program is unlikely to provide exact solutions to find the leak location supplanting more conventional leak pinpointing methods. However, it may contribute to leak surveys (Colombo, Lee, et al. 2009) possibly providing important new insights in water management itself.

5.2. Methodology

The methodology described here summarizes the methods in the related scientific publications (Steffelbauer and Fuchs-Hanusch 2016b; Steffelbauer, Günther, and Fuchs-Hanusch 2017; Steffelbauer et al. 2014a,b; Steffelbauer, Neumayer, et al. 2014) derived during the development of this thesis.

5.2.1. Leak localization as an inverse problem

In general, WDS analysis deals with forward problems (Colombo, Lee, et al. 2009). All hydraulic models described in Section 2.2 solve forward problems in which system parameters (e.g. demands, pipe roughnesses) are known and the network equations are solved to obtain hydraulic variables like flow and pressure. For the inverse problem as described by Pudar and Liggett 1992, the system states are known through measurements (demands, pressures, flows), for example, but some parameters are considered to be unknown (e.g. other demands, leaks, roughnesses). For the leak localization problem we presume that the network characteristics are known (e.g. demands and roughnesses) and, additionally, some flows and pressures are measured. The only unknown is the leak in the system which one aims to find. Subsequently, the model parameters—for the leak localization problem formulated as parameters describing the leak (e.g. leak position and magnitude)—are adjusted until the simulated values obtained from the hydraulic simulations match as perfectly as possible the measured pressure and flow values. This is an optimization problem and can be solved by the methods described in Section 2.1. The solution of this optimization problem, in terms of the parameter set describing the measurements best, provides the leak's position and magnitude. Of course, numerous ways may define the problem (Pudar and Liggett 1992) as can be seen in the various different techniques described in Section 5.1.

In this thesis model-based leak localization is mathematically formulated through minimizing the disagreement between a vector containing measured values $\mathbf{m}$ and a corresponding vector containing values from hydraulic simulations $\hat{\mathbf{m}}(\mathbf{x})$ with respect to the parameter vector $\mathbf{x}$. The difference $d(\mathbf{m}, \hat{\mathbf{m}}(\mathbf{x}))$ between the two vectors can be described using an arbitrary metric. In mathematical terms, the problem is formulated in following way

$$f(\mathbf{x}) = d(\mathbf{m}, \hat{\mathbf{m}}(\mathbf{x})) \quad \rightarrow \quad \min_{\mathbf{x}} f(\mathbf{x}) \qquad . \tag{5.1}$$

$f(\mathbf{x})$ represents a scalar (one-dimensional) function—the fitness or objective function as already described in Section 2.1. This function also spans the fitness landscape in the objective space over the parameter space (see Section 5.2.3).

The parameter vector $\mathbf{x}$ consists of the (i) leak magnitude and (ii) the leak's position in the WDS:

(i) Leaks are simulated using the power leak law (Ferrante et al. 2014) outflow equation as defined in Section 2.2 in equation (2.63). While making use of steady state simulations, emitter exponent e_e can be set to a fixed value of $e_e = 0.5$. Then the leak outflow Q is just dependent on the emitter coefficient c_e and the pressure p at the leak's position. p is calculated by the hydraulic solver, but c_e can be chosen to obtain the desired Q. Therefore, the single parameter c_e can be chosen to describe leak's magnitude.

(ii) The leak's position is given by the node ID in the WDS. In this thesis it is assumed—if not stated otherwise—that leaks occur at nodes.This is a small imprecision since real leaks occur at pipes or pipe fittings, but the imperfection of the localization methodology finding just the approximate area of a leak that contains many pipes and nodes makes this approach sufficient for this problem (similar to Pérez, Puig, Pascual, Quevedo, et al. 2011).

Thus, the parameter space is two-dimensional and adequately represented through a vector $\mathbf{x}$ of following form

$$\mathbf{x} = \begin{pmatrix} c_e \\ L_P \end{pmatrix} \quad . \tag{5.2}$$

Since the leak magnitude c_e is a continuous variable and the leaks position L_P is discrete, the problem (as formulated in equation (5.1)) is a mixed-integer programming problem (see Section 2.1.1).

Finding the right location and size of a leak equals finding the location of the minimum in the fitness landscape in the $c_e - L_P$ parameter space that is spanned through an arbitrary distance metric d.

In general, model-based leak localization is formulated as an optimization problem like in equation (5.1) which can be solved using any method presented in Section 2.1.

5.2.2. Model-based leak localization algorithm

Finding the right location and size of a leak corresponds finding the location of the minimum in the fitness landscape in the $c_e - L_P$ space in an arbitrary distance metric. The assumption that the problem as defined in equation (5.1) is multi-modal will be approved in the results section of this chapter (see Section 5.3). Hence, all deterministic algorithms (see Section 2.1.2.1) will generally fail to find the global optima respectively the most likely location of the leak. Thus, stochastic algorithms (see Section 2.1.2.2) will be used to localize leaks. Specifically, DE algorithm (see Section 2.1.2.2.2) will be used due to its good convergence properties already discussed in Section 2.1.

Note that the problem defined in equation (5.1) is a mixed-integer problem, but the DE algorithm is only capable of handling continuous problems. For that reason, the discrete L_P parameter has to be transformed. This problem is circumvented as described at the end of Section 2.1.2.2.2.

A schematic overview of the algorithm can be found in Figure 5.1. The inputs are organized in three distinct parts. First, the input parameters for the DE algorithm itself (see Section 2.1.2.2.2), including, for example, the parameters for the mutation operator Θ_M, the recombination operator Θ_R and selection operator Θ_S, settings of stopping criteria, population size or DE strategy. Second, the hydraulic network model that is used for the simulations loaded, for example, from a GIT repository serving as a input file database. Third, the measurement data $\mathbf{m}$ that is obtained by querying a TSDB (e.g. InfluxDB).

The DE algorithm is started suggesting a bunch of candidate solutions $\mathbf{x}$. This candidate solutions together with the hydraulic network model are simulated using OOPNET (Steffelbauer and Fuchs-Hanusch 2015) to obtain the simulation vectors $\hat{\mathbf{m}}(\mathbf{x})$.

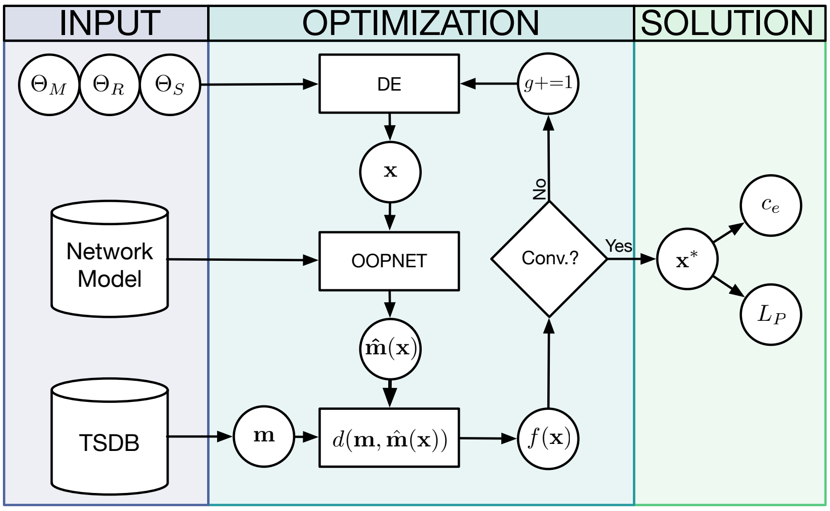


Figure 5.1.: Schematic representation of model-based leak localization algorithm

Subsequently, the fitness $f(\mathbf{x})$ is evaluated by calculating the distance function d between the measurements and the simulated values according to a specific chosen metric (see Section 5.2.3.1).

Finally, it is checked if the algorithm converged by validating the previously defined stopping criteria. If the algorithm converged, the best solution $\mathbf{x}^*$ of the population of candidate solutions yields the leak's position L_P and it's magnitude c_e. If the stopping criteria are not fulfilled, the number of the current generation g is increased and the candidate solutions are altered as well as evaluated once more according to the DE algorithm.

Nevertheless, the results may vary from leak localization run to run because of the stochastic nature of the DE algorithm. As mentioned in Section 2.1.2.2, heuristic algorithms cannot guarantee a global optimal solution, but at least have the possibility to converge into the global optimum in contrast to deterministic algorithms. That is the reason why, in general throughout this thesis, a couple of localization runs is repeatedly performed (e.g. 100 repetitions) using the same input parameters to increase the confidence in a specific found leak position and magnitude.

5.2.3. Fitness landscape analysis

Advances in understanding characteristic properties of an optimization problem and their effects on algorithm performance are critical (H. R. Maier et al. 2014). These characteristics are represented by the fitness landscape which was first introduced by Wright 1932. A multidimensional fitness landscape describes the search space of an optimisation problem spanned by evaluating the fitness function for all possible solutions. Optimization algorithms have to move through this landscape to find the optimal solution. Moreover, the fitness landscape is not only dependent upon the problem itself, but also on the choice of the algorithm and its parameters. (H. R. Maier et al. 2014) Completely different types of fitness landscapes are likely obtained for the same class of problems (e.g. demand calibration, leak localization). Likewise, completely different problems may possess similar landscapes. That is why specific algorithms might be ideal for finding optimal solutions to a whole problem class. Consequently, the applicability of different algorithms needs to be related to the properties of the fitness landscape of a particular problem (H. R. Maier et al. 2014). Hence, the most important motivation for fitness landscape analysis is retrieving the problem's characteristics enabling a better understanding of optimization algorithm performance (H. R. Maier et al. 2014)

Especially for real-world problems, computational efficiency is a crucial issue. Long evaluation times of simulation models combined with complex fitness landscapes can lead to high computational burden, which prevents finding global optimal solutions within reasonable time (H. R. Maier et al. 2014). Herein, fitness landscape analysis provides remedy by supporting in choosing the right algorithm. Its parametrization increases computational efficiency/convergence and decreases the timeframe to find global optima. Indeed, fitness landscape analysis might seem as the holy grail of optimization (Pitzer, Affenzeller, et al. 2011), however, fitness landscape analysis is usually much more resource intensive than just solving a given problem. It is therefore not the most economical option to solve single applied studies. Nevertheless, the resulting insights can be valuable for increasing problem understanding (Pitzer and Affenzeller 2012). That is why a closer look on the fitness landscape of, e.g., the leak localization problem may provide new insights on which class of optimization algorithms has to be preferred over others. For example, stochastic algorithms clearly outperform deterministic ones for multi-modal problems (as already discussed in Section 2.1). As the No-Free-Lunch theorem states, all optimization algorithms averaged over all possible problems perform equally well (Wolpert and Macready 1997). Nevertheless, Pitzer and Affenzeller 2012 concluded that fitness landscape analysis might provide at least some free appetizers (Droste et al. 1999).

The parameter space for model-based leak localization is a two-dimensional space consisting of the location L_P of the leak and its magnitude defined by the emitter coefficient c_e. The fitness function is clearly defined as the distance between the measurement vector and the vector obtained by hydraulic simulations with a specific $c_e - L_P$ parameter set. Consequently, the fitness landscape is obtained by calculating the fitness of all possible $c_e - L_P$ combinations. L_P is defined through every node in the system, the leak magnitudes have to be discretized to obtain numerical solutions. Every possible combination of c_e and L_P is evaluated, stored in a matrix

$$F = \begin{pmatrix} f_{11} & \cdots & f_{1n} \\ \vdots & \ddots & \vdots \\ f_{m1} & \cdots & f_{mn} \end{pmatrix} \tag{5.3}$$

where the single elements are built in the following way

$$f_{ij} = d\left(\mathbf{m}, \hat{\mathbf{m}}(\mathbf{x})\right) \quad . \tag{5.4}$$

$\mathbf{x}$ is the parameter vector defined in equation (5.2). The values of matrix F represent the fitness landscape for model-based leak localization. The matrix can be visualized as contour plot or heat map by allocating the fitness values to individual colors. Already first visualizations of the fitness landscape will provide deep insights on the leak localization problem itself on a basis never experienced before.

5.2.3.1. Metrics for objective functions

The fitness landscape depends heavily on the metric that is used to calculate the objective function. Crucial improvements are expected by finding the best metric for model-based leak localization, increasing the precision of the solutions as well as decreasing the computation time. There are different ways to formulate the objective function for leak localization since various metrics can be applied to the problem. All describe the discrepancy between measurements $\mathbf{m}$ and simulation results $\hat{\mathbf{m}}(\mathbf{x})$.

One family of metrics is the family of Minkowski distance metrics. Its general form is

$$d(\mathbf{m},\hat{\mathbf{m}}(\mathbf{x})) = \left(\sum_{i=1}^{n} |m_i - \hat{m}_i(\mathbf{x})|^p\right)^{1/p} \quad . \tag{5.5}$$

Popular forms of this distance metric are Manhattan distance or Taxi metric (Minkowski with $p = 1$)

$$d(\mathbf{m},\hat{\mathbf{m}}(\mathbf{x})) = \sum_{i=1}^{n} |m_i - \hat{m}_i(\mathbf{x})| \quad , \tag{5.6}$$

the Euclidean distance (Minkowski with $p = 2$)

$$d(\mathbf{m},\hat{\mathbf{m}}(\mathbf{x})) = \sqrt{\sum_{i=1}^{n} (m_i - \hat{m}_i(\mathbf{x}))^2} \tag{5.7}$$

or Chebyshev (or maximum) distance for the limit $p \to \infty$ of the Minkowski metric

$$d(\mathbf{m},\hat{\mathbf{m}}(\mathbf{x})) = \max_i |m_i - \hat{m}_i(\mathbf{x})| \quad . \tag{5.8}$$

The distance can also be expressed as an angle between the measurement vector and the vector resulting from the hydraulic simulations. This is called the cosine distance

$$d(\mathbf{m},\hat{\mathbf{m}}(\mathbf{x})) = 1 - \frac{\mathbf{m}\cdot\hat{\mathbf{m}}(\mathbf{x})}{\|\mathbf{m}\|_2\|\hat{\mathbf{m}}(\mathbf{x})\|_2} \quad . \tag{5.9}$$

It has to be noted that the cosine distance is not a proper distance metric since it violates the triangle inequality property. Subtracting the mean $\overline{m}$ or $\overline{\hat{m}}$ of the respective vectors leads to the correlation distance

$$d(\mathbf{m},\hat{\mathbf{m}}(\mathbf{x})) = 1 - \frac{(\mathbf{m}-\overline{m})\cdot\left(\hat{\mathbf{m}}(\mathbf{x}) - \overline{\hat{m}}(\mathbf{x})\right)}{\|(\mathbf{m}-\overline{m})\|_2\left\|\left(\hat{\mathbf{m}}(\mathbf{x}) - \overline{\hat{m}}(\mathbf{x})\right)\right\|_2} \tag{5.10}$$

Correlation and cosine distance are subtracted from 1 to guarantee that the distance stays positive. Other distance metrics that are used in this thesis are the Sørensen distance

$$d(\mathbf{m},\hat{\mathbf{m}}(\mathbf{x})) = \frac{\sum_{i=1}^{n}|m_i - \hat{m}_i(\mathbf{x})|}{\sum_{i=1}^{n}|m_i + \hat{m}_i(\mathbf{x})|} \tag{5.11}$$

and the Canberra distance

$$d(\mathbf{m},\hat{\mathbf{m}}(\mathbf{x})) = \sum_{i=1}^{n}\frac{|m_i - \hat{m}_i(\mathbf{x})|}{|m_i| + |\hat{m}_i(\mathbf{x})|} \quad , \tag{5.12}$$

very sensitive to small changes near zero.

5.2.3.2. Reordering of the parameter space

The fitness landscape depends not only on the distance metric, but also on the parameter space order which further governs its shape. For that reason, rearranging the coordinates of the fitness landscape can have an influence on the performance of the leak localization algorithm. The parameter space is two-dimensional with one dimension representing the leak size and the other dimension is representing the leak's location in the system.

The magnitude of the leak c_e is a metric variable, thus, it has a natural ordering from the smallest to the biggest value. This work focuses therefore on sorting the L_P parameter which misses a natural order. The underlying idea is that leak locations in close proximity will lead to similar pressure measurements. As a consequence, these alike measurements lead to analogous results by means of distance metrics. Thus, bringing nearby leak locations together in parameter space through rearranging will have a smoothing effect on the fitness landscape since local optima may unify (see Figure 5.2 for a schematic representation). This makes finding the minimum easier for optimization algorithms.

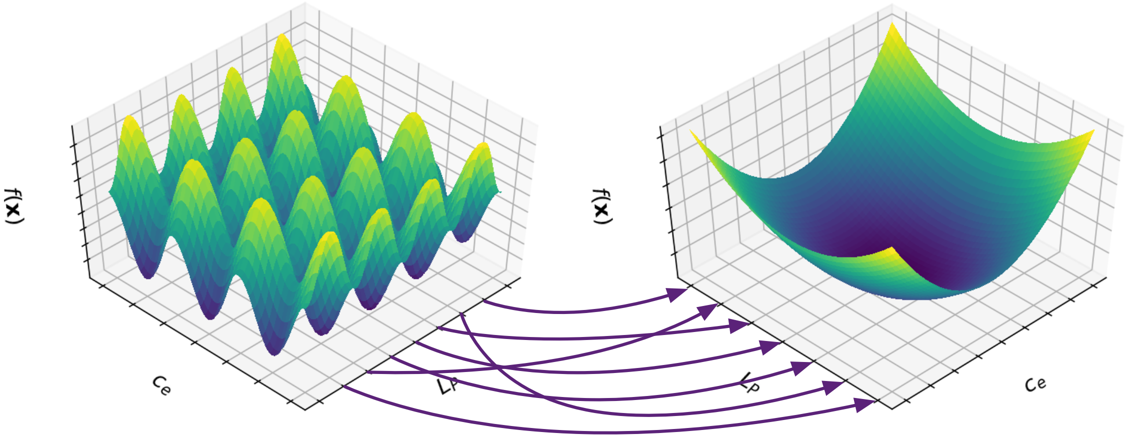


Figure 5.2.: Schematic representation of reordering the L_P-axis and its effect on the fitness landscape

Since WDN can be represented as graphs (see Section 2.2.4) and these graphs contain one or more cycles, ideal sorting does not exist for the nodes and hence the L_P axis. Thus, four different sorting algorithms are presented and applied on a small toy example with ten nodes and ten links (see Figure 5.3). These different sorting algorithms will be benchmarked later on in this thesis.

The results of the different sortings will be graphically represented as described next (see Figure 5.3). The naming convention of the nodes in the toy network is following: The ten nodes are named with capital letters from A to J, starting at the top with the letter A. Then the nodes are named from top to bottom and from left to right—TBLR-sorting (as can be seen in Figure 5.3). Note that this relates to the node names only. The order is represented in colors—going from dark to lighter colors which represent numbers. Imagine a sorting similar to the naming convention from top to bottom, but instead of left to right, the sorting is reversed going from right to left—TBRL (instead of TBLR). The ordering will still start at node A, but the second node in the TBRL ordering scheme is node D, since it is more to the right than the other nodes in this row, followed by C and B. Instead of the naming convention TBLR (*ABCDEFGHJ*) the order TBRL will be *ADCBFEJIHG*. For a real network consisting of hundreds of nodes, a numbering of the nodes is not reasonable, hence, the ordering of the nodes will be represented in a coloring scheme going from dark to light colors. This can be seen in Figure 5.3, where the network with colored nodes in the TBRL-order is represented as colors. Additionally, on the right side, the ordering of the L_P axis is represented in which the fitness landscape will be shown as well as the optimization algorithm will operate.

The four different orderings of the L_P parameter in the parameter space are represented in the following paragraphs:

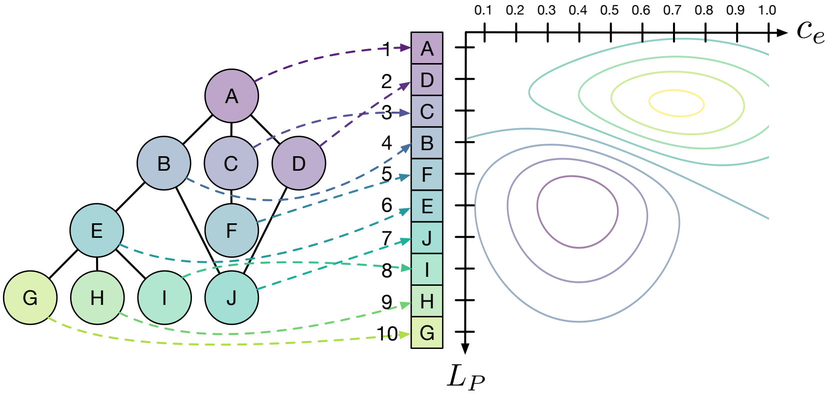


Figure 5.3.: Naming convention of toy problem (letters A to J) and color representation of sorting scheme TBRL—First Top before Bottom then Right before Left—of the L_P-axis

Table 5.2.: Results of sorting algorithms for the example network in Figure 5.4

Sorting	A	B	C	D	E	F	G	H	I	J
Alphabetical	1	2	3	4	5	6	7	8	9	10
Cuthill-McKee	7	5	9	8	2	10	1	3	4	6
Depth-First Search	1	2	9	8	3	10	4	5	6	7
Random Sorting	3	10	5	4	2	7	6	9	8	1

5.2.3.2.1. (i) Alphabetical

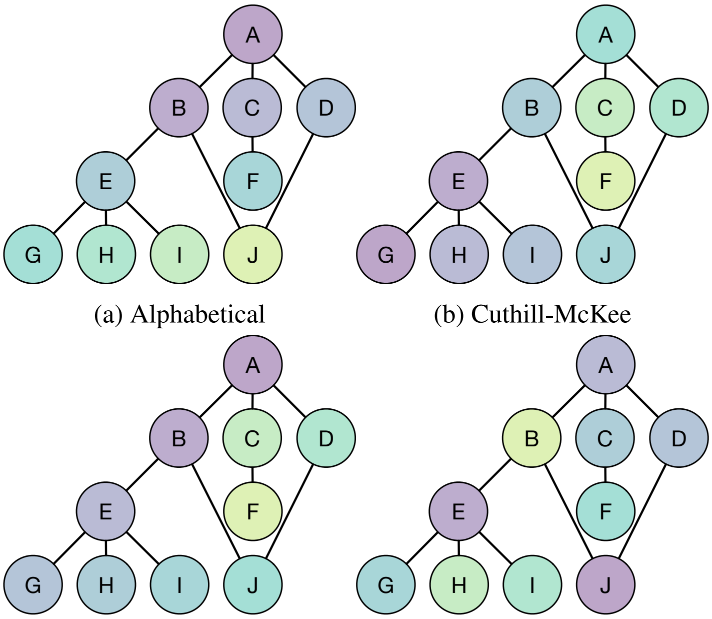
The first ordering is an alphabetical sorting of the names of the junctions. Hydraulic models are either directly constructed by humans or exported from GIS files which again are generated by humans. The naming of nodes in these files complies with naming conventions made by the WU, for example. OOPNET (Steffelbauer and Fuchs-Hanusch 2015) sorts the nodes routinely alphabetically. Hence, network models appear automatically in this alphabetical order in compliance with their naming conventions. In general, this represents a good sorting as naming conventions usually comply with, for instance, street names, property numbers, parts of the network or start from top to bottom and left before right (see Figure 5.4a). For that reason, alphabetical sorting is not random, but may not be the best order of nodes for leak localization. For instance, if trunk mains start with the letter m and household connections start with h, household connections—although geographically at the same location as a leaky trunk main, may be in the L_P space at complete different positions. Since the nodes are in alphabetical order, the sorting of the L_P axis is $L_P = \{ABCDEFGHIJ\}$.

5.2.3.2.2. (ii) Cuthill-McKee

The second ordering—the Cuthill-McKee algorithm (Cuthill and McKee 1969)—defines a nodal numbering scheme. It reduces the bandwidth of the incidence/adjacency matrix of the network graph respectively WDN (see Section 2.2 equation (2.52)). The bandwidth k of a matrix A is the maximum number such that

$$A_{ij} = 0 \quad \text{if} \quad |i-j| > k \quad \forall i,j \quad . \tag{5.13}$$

Reducing the bandwidth of the adjacency matrix results in an order where connected nodes and node clusters tend to have minimal discriminate node labels. The adjacency matrices of the toy example before and after applying the Cuthill-McKee algorithm can be found in Figure 5.5. The adjacency



(a) Alphabetical

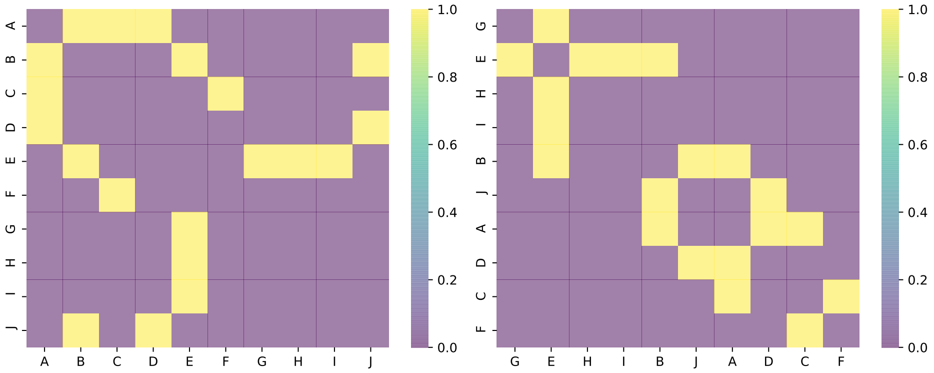
(b) Cuthill-McKee

(c) Depth-First Search (DFS)

(d) Random Sorting

Figure 5.4.: Comparison of different sorting algorithms on a small toy example

matrix for the alphabetical order is found in Figure 5.5a, the adjacency matrix for the Cuthill-McKee labeling scheme is found in Figure 5.5b. Obviously, the bandwidth for the Cuthill-McKee algorithm is smaller. The ordering of the L_P-axis after applying the algorithm results in $L_P = \{GEHIBJADCF\}$. The bandwidth before applying the algorithm is $k = 7$ and $k = 3$ after application of the Cuthill-McKee ordering scheme. Additionally, the Cuthill-McKee sorting of the toy network with corresponding coloring scheme as described before is depicted in Figure 5.4b.



(a) Alphabetical, $k = 7$

(b) Cuthill-McKee, $k = 3$

Figure 5.5.: Graphical representation of the adjacency matrix of the toy example before and after applying the Cuthill-McKee algorithm

5.2.3.2.3. (iii) Depth-First Search (DFS) Third, a DFS algorithm is used (Diestel 2017). Starting at a root node, the DFS algorithm explores every node once by reaching as deep into branches of the graph as possible. An exemplary rule might be that if the algorithms is at a fork in the path, the algorithm always chooses the most right node to continue its search. If the algorithm aborts, either because there are no more nodes in the branch to visit (dead-end) or all nodes connected to the current node have been visited (circle), the algorithm returns to the last visited fork and searches all possible not previous visited paths from there. This guarantees that the nodes at a branch are connected in the parameter space. Figure 5.4c shows DFS for the toy example starting at root node A. Following the described search rules the algorithm results in sorting $L_P = \{ABEGHIJDCF\}$.

5.2.3.2.4. (iv) Random Sort Finally, the nodes are sorted in a completely random way for benchmarking purposes to test, if the ordering of the parameter space has an influence on the convergence for leak localization at all. A random sorting of the nodes is generated with a random number generator, resulting in the node sequence $L_P = \{JEADCGFIHB\}$ and depicted in Figure 5.4d.

All resulting sorting for the toy example as described in the previous paragraphs and depicted in Figure 5.4 are summarized in Table 5.2.

5.2.4. Extension to extended period simulations

The approach described in Section 5.2.2 is extended with a similar method as described in Casillas, Garza-Castanon, et al. 2013, since the measurement points can be virtually increased by additional consideration of the time domain. This increases the available measurements leading to increased robustness of the algorithm and increased sensitivity according to the arguments brought up by Pudar and Liggett 1992 on their study on over- and under-determined systems for model-based leak localization. In general, the pressure and flow measurements for leak localization are available not only for single time-steps, but also over a certain time-span in the time domain. Hence, a time-dependent[29] measurement vector $\mathbf{m}_t$ can be formed. Additional simulations can be made dependent on time by introducing patterns describing the demand. This leads as well to a time-dependency of the modeling results $\hat{\mathbf{m}}_t$. Note that this are just EPS, or in other words, a series of steady-state simulations over time-dependent demand loads and changing tank water levels. In general, leak parameters are not time dependent[30] $\mathbf{x} \neq f(t)$ since the leaks position found by the algorithm stays constant over time as, logically, leaks do not travel through the system. The time dependency by the means of pressure dependent demand is accounted for through the emitter coefficient c_e. A possible expansion of the approach improving the pressure-dependency conditions of leaks is by including the emitter exponent e_e in the genome of the leak localization problem.

Since $\mathbf{m}_t$ and $\hat{\mathbf{m}}_t(\mathbf{x})$ are time-dependent, the fitness function depends on t as well

$$f_t(\mathbf{x}) = d\left(\mathbf{m}_t, \hat{\mathbf{m}}_t(\mathbf{x})\right) \quad . \tag{5.14}$$

To keep the optimization problem still single-objective, the fitness function is averaged over time.

$$f(\mathbf{x}) = \frac{1}{N_t} \sum_{t=1}^{N_t} f_t(\mathbf{x}) \quad \rightarrow \quad \min_{\mathbf{x}} f(\mathbf{x}) \qquad , \tag{5.15}$$

[29] Time dependency is expressed by a subscript t.

[30] Of course, leaks can increase their size over time, but this effect is not taken into account in this approach in the interest of simplification.

where N_t is the number of time steps respectively measurements over time that are taken into account. This approach is similar to (Casillas, Garza-Castanon, et al. 2013). The algorithm detects if the measurement vector is time dependent and reacts to it by performing EPS simulations over the same time period as contained in the measurement vector in a complete automated way.

5.2.5. Extensions to time series analysis forecast models

For pairing the leak localization with forecast models as described extensively in Chapter 3 and to additionally make the model-based leak localization more robust by minimizing the effect of modeling uncertainties, the localization problem as defined in equation (5.1) is slightly reformulated. Instead of taking the direct measurements, the changes or residuals are used. These residuals are differently produced, depending on if they are generated through (i) measurements (replacing **x** in equation (5.1)) or through (ii) simulations (replacing $\hat{\mathbf{m}}(\mathbf{x})$ in equation (5.1)).

(i) On the measurement side of the equation, the residuals are constructed as the difference between a forecast with a time series model $\hat{x}_t$ supposing leak-free conditions in the network and the actual measurements x_t of the system. Mathematically, this is formulated in following way

$$\Delta\hat{\boldsymbol{\mu}}_t \;=\; \hat{\mathbf{x}}_t - \mathbf{x}_t \quad . \tag{5.16}$$

On this side of the equation the flow and pressure differences computed with the methods described, for example, in Section 3.3.5.3 and 3.3.5.5 can be used.

(ii) On the simulation side of the equation, the residuals are generated through subtracting simulations with a leak $\hat{\mathbf{m}}_t(\mathbf{x})$ defined by the parameter vector **x** (see equation (5.2)) from leak-free simulations with $\mathbf{x} = \mathbf{0}$, or in mathematical terms

$$\Delta\hat{\mathbf{m}}_t(\mathbf{x}) \;=\; \hat{\mathbf{m}}_t(\mathbf{0}) - \hat{\mathbf{m}}_t(\mathbf{x}) \quad . \tag{5.17}$$

Consequently, this results in following objective function

$$f_t(\mathbf{x}) \;=\; d(\Delta\hat{\boldsymbol{\mu}}_t, \Delta\hat{\mathbf{m}}_t(\mathbf{x})) \quad . \tag{5.18}$$

The mean over the time can either be computed as in equation (5.15), or the time average is build over the residual estimates beforehand of the simulation.

Furthermore, if good estimates for the leak outflow exist, these estimates can be directly incorporated into the optimization problem itself. This has the advantage of simplifying the parameter vector and hence reduces the search space to a single parameter. For this reason the leak is generated as an additional demand Δq of the size of the leak outflow estimate at a specific node.

$$\mathbf{x} = \begin{pmatrix} L_p \\ c_e \end{pmatrix} \;\rightarrow\; \mathbf{x} = \begin{pmatrix} L_{P|\Delta q} \end{pmatrix} \quad . \tag{5.19}$$

Consequently, the leak localization task consists only of finding the leak position through utilizing the remaining pressure measurements without additionally finding its size. This can be done trough total enumeration of the parameter space resulting in as many hydraulic simulations as possible leak positions in the system.

5.2.6. Quality parameters for leak localization performance evaluation

In general, four different measures are adduced for evaluating the quality of the leak localization performance. These four parameters are following

(i) The **topological distance** d_T between the real leak $\mathbf{x}_R^*$ and the leak found by the algorithm $\mathbf{x}_F^*$—This distance is calculated through Dijkstra's shortest path algorithm (Dijkstra 1959) using the Graph theoretic representation of a WDS (see Section 2.2.4).

(ii) The **false positives** FP—This is the percentage of nodes with a fitness function value smaller than the fitness function value of the real leak. Mathematically this can be written by

$$N_{FP} = |\{i \in \{1,2,\ldots,N_n\} : f(\mathbf{x}_i) < f(\mathbf{x}_R)\}| \tag{5.20}$$

$$FP = \frac{N_{FP}}{N_n} \cdot 100 \quad , \tag{5.21}$$

where N_n is the number of all nodes in the network and N_{FP} is the number of all false positive nodes. This measure is equal to the false positive measure defined by Moors et al. 2018.

(iii) The **maximum span** (MS) of the false positives—This measure is defined as the maximum value of the topological distances between all false positive nodes

$$MS = \max_{i,j} (d_T(i,j)) \quad \forall i,j \in FP \quad . \tag{5.22}$$

(iv) The **leak-size** convergence—The difference between the real and the estimated leak size, either expressed in c_e values or leak outflow Q_L.

5.2.7. Differences to similar approaches in scientific literature

The technique described in this methodology section is a model-based leak localization approach solving an inverse optimization problem using steady-state hydraulic models as introduced by Pudar and Liggett 1992. This approach is capable of using both types of hydraulic sensors—pressure as well as flow measurements. The optimization problem is solved with evolutionary algorithms and formulated using parameters describing the leak, for example, the leak's location and the leak's magnitude. This approach, at first glance, is very similar to two formulations found in literature, namely the approaches described in (i) Z. Wu et al. 2009 and (ii) Casillas, Garza Castañón, et al. 2014. Here is a description of the differences between the approaches:

(i) **Distinction to Z. Wu et al. 2009:** At first glance, solving the optimization problem with genetic algorithms by calibrating leak parameters sounds very similar to the work of Z. Wu et al. 2009. Looking at more detail, these two approaches are far apart. Four main distinctions can be made: First, Wu's approach couples finding leakages with other calibration tasks as calibrating roughness values. These two calibration tasks are decoupled in the approach in this thesis. Second, the genome of Wu describing leakages looks nearly the same as the approach in this thesis. A closer look at Wu's work shows that it focuses on multiple leaks to find background leakages solving the optimization problem for 25 and more leaks together. This is a complete conceptual different scope than in this thesis, where the aim is to detect a leak as it occurs and locate it as fast as possible. Third, the high number of sensors with the comparably large leaks greater than 10 L/s in Wu's work differs from the scope of this thesis. This work focuses on smaller leaks with less sensors. And finally, Wu uses a competent genetic algorithm without testing if the algorithm is sufficient for solving the problem. This thesis tries to improve the performance by finding the best algorithm and additionally increasing the performance of this algorithm by analyzing the optimization problem itself.

(ii) **Distinction to Casillas, Garza Castañón, et al. 2014:** Although different fitness functions are benchmarked which are similar to certain metrics described in Section 5.2.3.1 the approach is very different. Casillas, Garza Castañón, et al. 2014 uses direct methods for the metrics. The simulation results are compared with pre-computed columns of the sensitivity matrix. This means that for each leak size and for all time-steps, leak scenarios have to be computed before leak localization runs can be conducted. To the best of my understanding, only the least-square optimization method uses an approach somehow similar to this thesis' approach. The methods introduced in this thesis perform leak localization with only indirect methods on the fly without the necessity of pre-computed sensitivities and, therefore, is more economical in terms of computation time. Casillas, Garza Castañón, et al. 2014 for themselves state in their work that indirect methods are allowing more information about the leak (e.g. they additionally provide correct leak sizes better fitting the pressure data). Furthermore, the work of Casillas, Garza Castañón, et al. 2014 deals with huge "leaks", e.g. of 50 L/s and more in the Hanoi network. It is questionable if the sensitivity matrix—a linear approximation of the system response to leaks—still describes the non-linear WDS behavior realistically for such enormous faults. Additionally, it is doubtful that inexplicable small leak sizes of 0.01 L/s in the Quebra network lead to measurable pressure drops. Of course, ideas like increasing the robustness of the algorithm by extending the problem in time utilizing EPS are without any doubt influenced by the work of Casillas, Garza-Castanon, et al. 2013.

The overall aim and scope of this thesis is very distinct from literature examples in Section 5.1. Comparison with Table 5.1 shows that leak localization has been applied to mostly simulations and not real-world field studies with mainly large leaks or bursts and numerous pressure sensors. This thesis aims to develop and apply leak localization algorithms on small leaks smaller than 1 L/s on real-world examples with a comparable small number of pressure sensors. Certainly, this is a great challenge.

5.3. Results and discussion

5.3.1. Simulations in a real-world network

First, model-based leak localization is performed through simulations in a real-world network. The network is the Ragnitz-Simple network already presented in Section 2.4.3. This simulations serve as a benchmark, whether the algorithm is capable of finding leaks in a perfect environment consisting of sensors with no measurement uncertainties and a flawless hydraulic model. However, the accuracy of the field-study sensors (see Section 2.4.3.1) is taken into account by taking only significant decimal digits of the simulation results.

A small leak—compared to the literature in Section 5.1—is generated in the system. The leak is simulated through the leak emitter equation (2.63) with an emitter coefficient of $c_e = 0.12$ and an $e_e = 0.5$ resulting in a leak outflow of $Q_L \approx 1.0$ L/s. The leak position in the system is depicted in Figure 5.6 as a gray cross. The position is equal to position III in Figure 2.32 where a real artificial leak was generated during the case study. The sensor positions for the five pressure sensors correspond to the positions that resulted from the sensor placement method after Casillas (Figure 4.7c in Section 4.3.3) and are depicted in Figure 5.6 as green squares. Additionally, the inflow is measured at the tank depicted as green hexagon in Figure 5.6.

The objective function (equation 5.1) is defined via the cosine metric in equation (5.9) and the parameter space defined as in equation 5.2 is in Cuthill-McKee order (see Section 5.2.3.2). The optimization problem is solved using DE (see Section 2.1.2.2.2) with the DE/rand/1 formulation. The parameters for the algorithm are $CR = 0.7$ and $F_1 = 0.5$ with a population size of $\mu = 30$ and $N_I = 100$ iterations per optimization run. Since stochastic algorithms can get stuck in a local optima, each optimization run is 200 times repeated to retrieve a statistic over the likelihood that the algorithm finds the leak. A single optimization run performs 3000 hydraulic simulations and single hydraulic simulations are evaluated in parallel to accelerate the computation. Consequently, an optimization run takes 21 seconds on a four core processor. The whole 200 optimization runs take 70 minutes for a total of 600.000 EPANET simulations. The results are presented in Figure 5.6 and in Table 5.3.

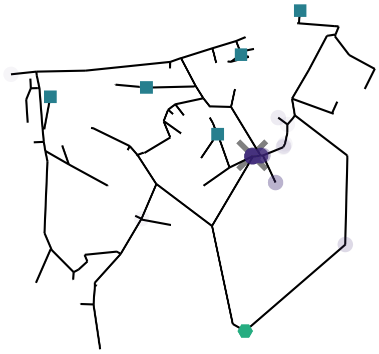


Figure 5.6.: Results of 200 leak localization runs for the Ragnitz-Simple network

In Figure 5.6 the found locations by the algorithm are depicted as purple translucent circles. The opacity of the locations corresponds to the frequency of how often the algorithm has found a leak in the vicinity. 89 % of the simulation results were within 40 meter of the simulated leak and only 4.5 % of the found solutions were located outside of a distance of 200 meter. It has to be noted that distance means always the topological distance d_T along the pipes computed with Dijkstra's shortest path algorithm. This computation is very efficient in the Graph theoretic representation of a WDS (see Section 2.2.4).

Table 5.3 shows a statistical summary of the simulations in terms of the found emitter coefficient c_e, the d_T and the final fitness value after an optimization run $f(\mathbf{x})$. The mean values and the standard deviations are high, but this resulted from simulations which final solutions are very distant to the leak. The quantiles show that most of the solutions are in fact in close proximity to the true leak resulting also in a similar leak outflow.

5.3.2. Fitness landscape analysis

The importance of the fitness landscape analysis was already introduced in Section 5.2.3. This section presents results and deeper insights from this approach on model-based leak localization. First, the

Table 5.3.: Statistics of 200 leak localization runs

	c_e	d_T	$f(\mathbf{x})$
mean	0.36	47.1	3.7e-07
std	2.02	145.7	2.2e-06
min	0.00	0.0	-2.2e-16
25%	0.12	3.8	-2.2e-16
50%	0.12	7.0	-2.2e-16
75%	0.12	32.3	-2.2e-16
max	20.00	1029.4	1.4e-05

effect of the number of sensors on the fitness landscape is discussed. Second, different metrics are tested to see their influence on the optimization problem. Both tasks are applied on a toy example from literature, because the results are demonstrative. Subsequently, the fitness landscape of the real-world Ragnitz network is examined in detail, including an investigation of different sorting algorithms as described in Section 5.2.3.2 and also different fitness metrics. Model-based leak localization will be applied on both networks for different metrics as well as different parameter space orderings—assessing the influence of landscape's shape on the optimizer. It has to be noted that the work presented here in this section is a summary of the work presented in two conference papers, namely, (i) Steffelbauer and Fuchs-Hanusch 2016b and in (ii) Steffelbauer, Günther, and Fuchs-Hanusch 2017.

5.3.2.1. Reducing the number of sensors

For the first vivid examples, the network of Poulakis (see Section 2.4.1) is taken. A leak is generated in the system at node J-16 through the emitter equation (2.63) with $c_e = 0.2$ and $e_e = 0.5$, resulting in a leak outflow of $Q_L = 1.0$ L/s. In Figure 5.7 the leak position is marked with a gray cross.

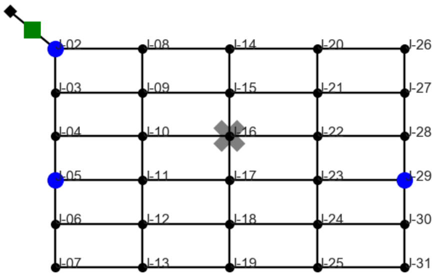


Figure 5.7.: Network of Poulakis (Poulakis et al. 2003) with flow meters (green squares) and used pressure sensors (blue circles) and leak position (black cross) for generating fitness maps

Subsequently, the fitness landscape is generated as described in Section 5.2.3. In this section, the Euclidean metric is used to compute the objective function for the fitness landscape. First, a flow sensor at every pipe and a pressure sensor at every junction is assumed. The corresponding fitness landscape can be found in Figure 5.8a. A short note on the fitness landscapes: every fitness landscape plot in this thesis is depicted in $\log_{10}$-scale to emphasize small differences around the local and global optima. Additionally, at the top, the fitness values F in $\log_{10}$-scale varying over c_e are depicted containing the minimum, mean and maximum over all leak positions L_P and the results corresponding to the actual leak position (named "at leak"). Similar curves are depicted at the right hand side of the fitness landscape with the F values varying over L_P, again with minimum, mean, maximum and at leak (values corresponding to $c_e = 0.2$) curves generated now over all possible c_e instead of the L_P values.

The fitness landscape for every known flow and pressure in the system shows a clear and distinct global optimum at the right leak point L_P =J-16 and at the right emitter coefficient value location with $c_e = 0.2$ in the parameter space. Nevertheless, already for a system where every information is known, small and shallow local optima occur at L_P = J-10 and L_P =J-22. Since it is uneconomically and unrealistic to install measurement devices at every pipe and every node in a WDS, the sensor number is reduced. Clearly, this has an influence on the shape of the fitness landscape.

Figure 5.8b shows the fitness landscape, if pressure is measured at every node, but the flow is only measured at the inflow point. Indeed, the landscape shows a different structure than before. More local optima occur, however, the global optimum is still at L_P =J-16 and $c_e = 0.2$.

Reducing the sensor number even more exaggerates this effect (see Figure 5.8). For that reason, only the inflow is measured and only three pressure sensors are placed in Poulakis net, although, the pressure sensor location for position J-05 and J-29 are computed through the optimal sensor placement algorithm of Casillas. The position at J-02 represents a pressure measurement at the inflow point of the DMA. Additionally, the inflow is supposed to be measured in the system. The exact locations of the sensors are depicted in Figure 5.7. It can be clearly seen that more and more peaks arise in the landscape. Furthermore, the global optimum moved from the real leak location to its neighboring node J-15 and to a slightly smaller c_e value. Thus, even with a perfect optimization algorithm capable of finding the global optimum with absolute certainty—although no such algorithm exists—will not be able to find the exact leak location in the WDS even in a perfect environment possessing no measurement uncertainties nor model inaccuracies. Already by simply reducing the number of pressure measurements, only approximate locations of the leak can be retrieved by a model-based leak localization anymore.

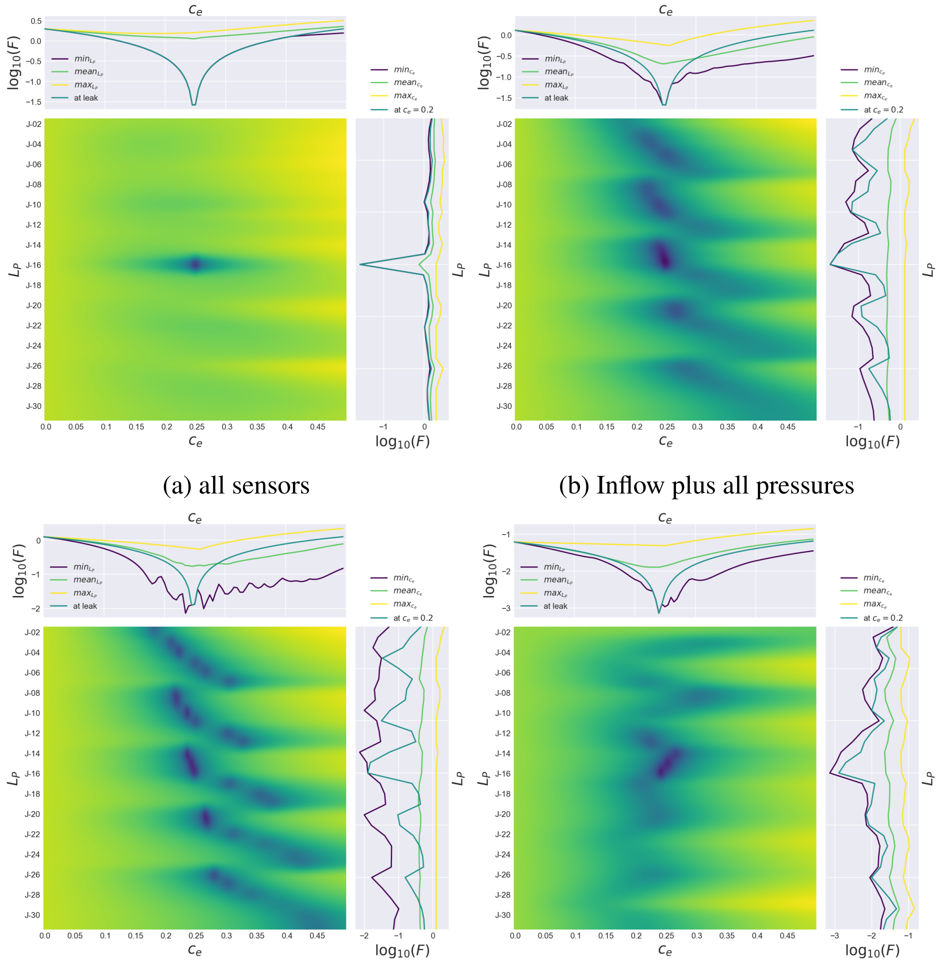
Abandoning additionally the inflow measurements in Figure 5.8d, changes the situation for the worse. The global optimal solution spreads now over the leak locations J-15 and J-14. Furthermore, the whole fitness landscape blurs and washes out and does not remotely resemble the ideal case in Figure 5.8a.

The relative large number of local optima[31] that was observed in the first attempts to depict the fitness landscape, suggests that the model-based leakage localization problem can only be solved with metaheuristic algorithms like DE.

5.3.2.2. Different metrics—different landscapes

Not only the number and position of sensors affects the form of the fitness landscape, also the metric in which the objective function is formulated influences the shape. For that reason, the different metrics

[31] Even the ideal case with the maximum number of sensors shows shallow local optima where deterministic algorithms can get stuck.



(a) all sensors

(b) Inflow plus all pressures

(c) optimal sensor placement plus inflow

(d) only three pressure sensors

Figure 5.8.: Fitness landscapes for different sensor numbers

described in Section 5.2.3.1 are applied on the Poulakis network with the three pressure and an inflow sensors and a leak at the same node as depicted in Figure 5.7. Only the leak size is set to $c_e = 0.25$ to place the leak in exactly the middle of the subsequent figures. This corresponds to a leak outflow of $Q_L = 1.27$ L/s.

The Euclidean metric (Minkowski order $p = 2$) defined for this problem in equation (5.7) was already depicted in Figure 5.8c. The fitness landscape resulting from the Minkowski metric with lower order ($p = 1$)—the Manhattan metric as defined in equation (5.6)—is depicted in Figure 5.9a. The problem using the maximum metric—the Minkowski metric with highest order of $p = \infty$—is illustrated in Figure 5.9b. Further distance metrics are the cosine metric (see equation (5.9)) in Figure 5.9c, the correlation metric (equation (5.10)) in Figure 5.9d, the Sørensen metric (equation (5.11)) in Figure 5.10a and the Canberra metric (equation (5.12)) in Figure 5.10b.

The Minkowski metrics as well as the Sørensen show the same behavior—a very peaky fitness landscape consisting of many narrow local optima with nearly the same fitness value and a global optimum which is not exactly at the same position but approximating the real leak. Due to the shape of the fitness landscape, model-based leak localization can only be solved with meta-heuristic algorithms.

On the other hand, the landscape resulting for the cosine metric (Figure 5.9c) possesses only one distinct minimum, which is at the correct L_P and c_e value as the real leak. Looking at the function closely reveals that additional local optima exist. However, they are shallower and not as distinct as the global optimum. For that reason, meta-heuristic algorithms are still necessary for finding leaks, although, convergence to the global optimum is expected to be easier for algorithms concerning model-based leak localization formulated in cosine metric.

The fitness landscape calculated with the correlation metric is shown in Figure 5.9d. Three distinct minima are generated through this metric. Again, the global minimum is not at the correct L_P nor the correct c_e value.

The fitness landscape computed with the Canberra metric in Figure 5.10b shows a behavior between the Minkowski metrics with many local optima and a global optimum similar to the simulations with no inflow sensors in Figure 5.8d. Additionally, the global optimum is not at the right leak position and does not have the right c_e value.

In the next section, the influence of the distance metric on the convergence of the DE algorithm is investigated in detail.

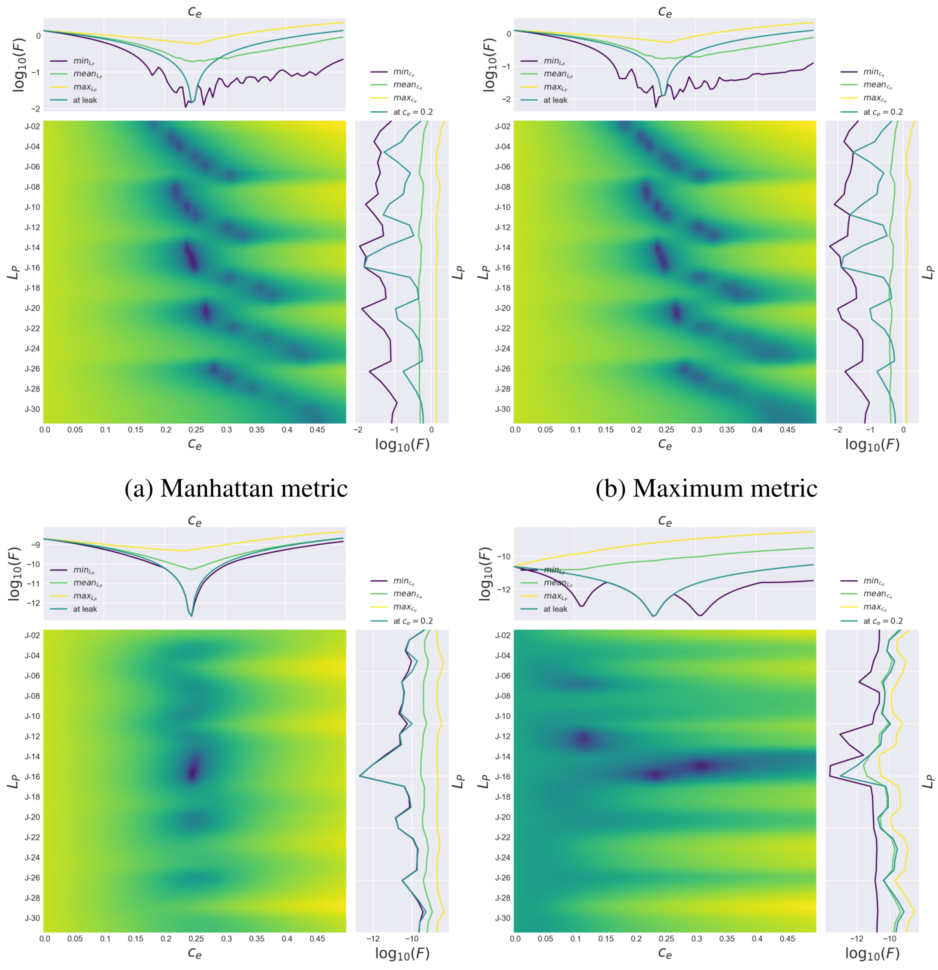
5.3.2.3. Influence of the metric on the leak localization performance

The model-based leak localization problem is solved with the DE algorithm for the different distance metrics (equations (5.5) to (5.12)). For each metric, 1000 optimization runs are undertaken. The c_e value is always the same ($c_e = 0.25$), but the leak's location is chosen at random for each optimization run. Leaks are generated in the middle of the pipe. DE/rand/1 algorithm is chosen and its parameters are $F_1 = 0.5$, $CR = 0.7$, population size $\mu = 20$ and the optimization takes place for $N_I = 100$ iterations (see Section 2.1.2.2.2). In total, this results in 2000 hydraulic simulations for each optimization run, or 2 million hydraulic simulations for each metric.

The efficiency is tested by comparing the metrics through two different performance measures. The first performance measure—connected with the L_P parameter—is the average topological distance $\overline{d}_T$ of the leak found by the algorithm to the real leak as a function of the objective function evaluations. $\overline{d}_T$ is build over all optimization runs, but for each distance metric separately. The results for $\overline{d}_T$ can be found in Figure 5.11a.

Similarly, the second performance measure is computed—the convergence of the emitter coefficient c_e in dependency of objective function evaluations averaged over all optimization runs. This curve represents the algorithms ability in estimating the real leak magnitude. Results are presented in Figure 5.11b.

In Figure 5.11 it can clearly be seen that the distance metrics of the Minkowski family (manhattan, euclidean and max) convergence the slowest. Specifically, higher Minkowski metric orders p result in slower convergence of $\overline{d}_T$. Furthermore, the Minkowski metric results in poorer solutions at the end of the optimization compared to cosine, correlation or Sørensen metric. That is a consequence of the many local optima in the fitness landscape resulting for Minkowski metrics. Overall, cosine metric performs best in converging to the right leak spot. Of course, this is due to the existence of an outsticking global optimum in the fitness landscape at the right leak location. Canberra and correlation distance perform similar to cosine metric. Canberra metric has the property to be highly sensitive to small changes of values near zero, consequently, its good results are caused by this sensitivity. Sørensen metric performs somewhere in between Minkowski and cosine metric—starting slowly but



(a) Manhattan metric

(b) Maximum metric

(c) Cosine metric

(d) Correlation metric

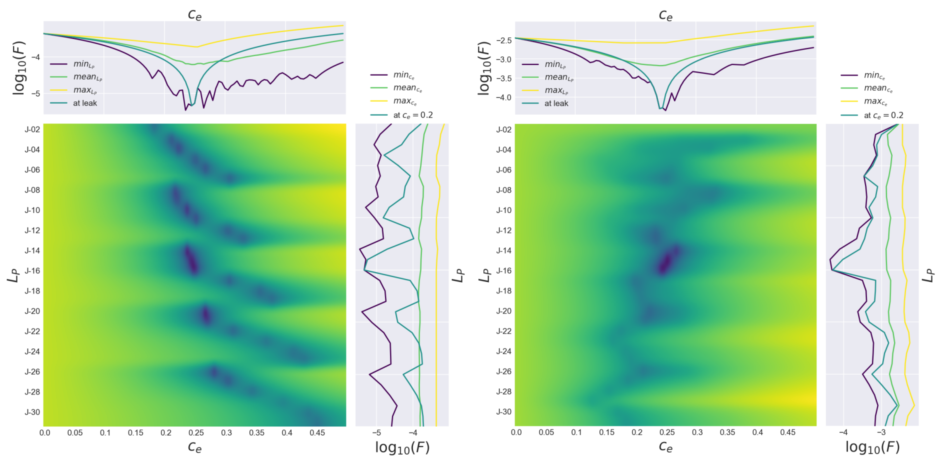
Figure 5.9.: Comparison of different metrics on fitness landscape (Part 1)

finishes with the same performance as the cosine metric after a few hundred function evaluations. Furthermore, solutions do not improve after approximately 500 function evaluations anymore for any metric.

The convergence speed in the c_e parameter is complementary to the distance convergence. Concerning c_e values, the Minkowski metrics perform best whereas cosine, Canberra and correlation metric behave poorly. Nevertheless, the clear winner is the Sørensen metric. It has to be noted that DE tends to overestimate c_e values for all distance metrics. The convergence of c_e is less important than the convergence speed of L_P. Model-based leak localization is more focused on finding the right position of the leak instead of its magnitude, because the leak outflow can be estimated through, e.g., inflow measurements.

Nevertheless, a method will be presented further on (see Section 5.3.5) that utilizes the differences in the convergence of the two parameters for finding leaks when no flow measurement is available.

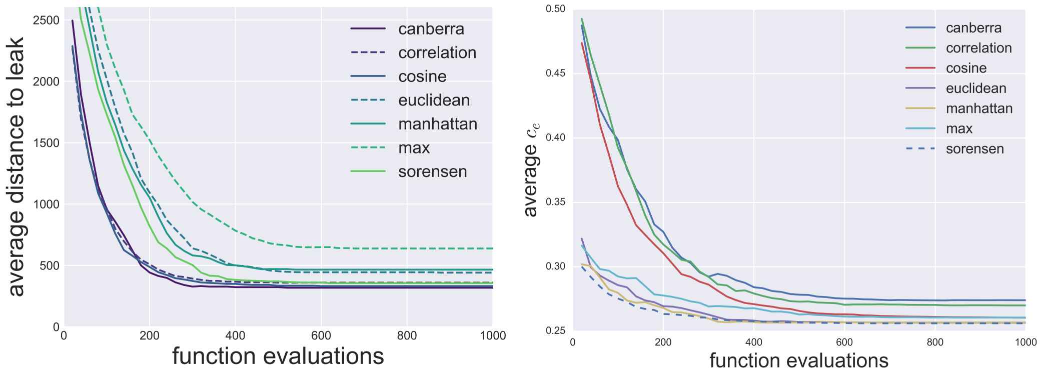
Analysis of the final leak position estimates showed that after 2000 function evaluations, Canberra and cosine metric resulted in the best leak position estimates, whereas the Minkowski family performed



(a) Sorensen metric

(b) Canberra metric

Figure 5.10.: Comparison of different metrics on fitness landscape (Part 2)



(a) d_T convergence

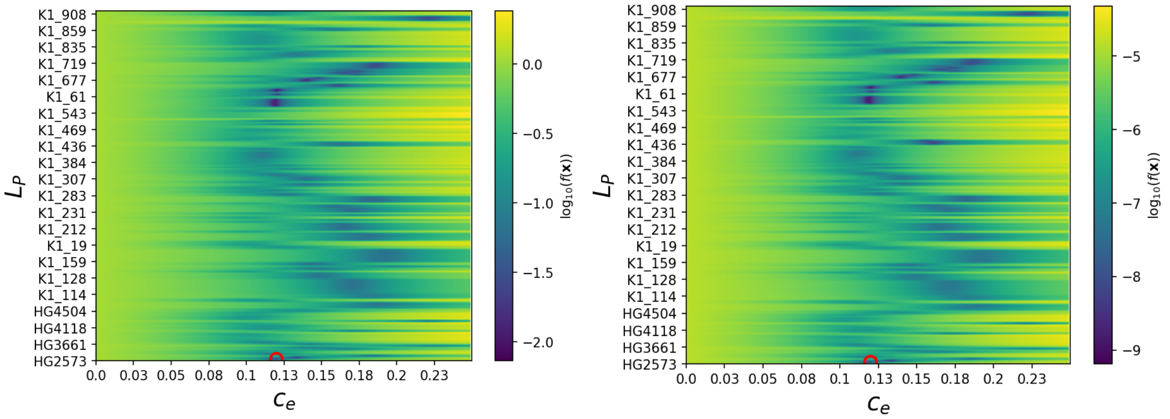
(b) c_e convergence

Figure 5.11.: Average convergence of the (a) topological distance d_T and of the (b) emitter coefficient c_e as a function of fitness function evaluations for different distance metrics

worse.

5.3.2.4. Fitness landscapes of a real-world network

The fitness landscapes of the real-world Ragnitz network, as described in Section 5.3.1, and depicted in Figure 5.6 are computed in this section. The leak is at the same position as stated in the former section and the sensor positions are also the same. The fitness landscape for the Euclidean metric can be found in Figure 5.12a, the results for the cosine metric are depicted in Figure 5.12b. Landscapes are interspersed with a huge amount of local minima. Switching to another metric influences the fitness landscapes as already discussed before, but does not reduce the problem's complexity arising from the many minima. For that reason, search space order is improved as already described in Section 5.2.3.2 which will potentially smooth the fitness landscape.



(a) Euclidean metric

(b) Cosine metric

Figure 5.12.: Fitness landscape resulting for real-network (alphabetical sorting). The leak's position in the fitness landscape is marked with a red circle.

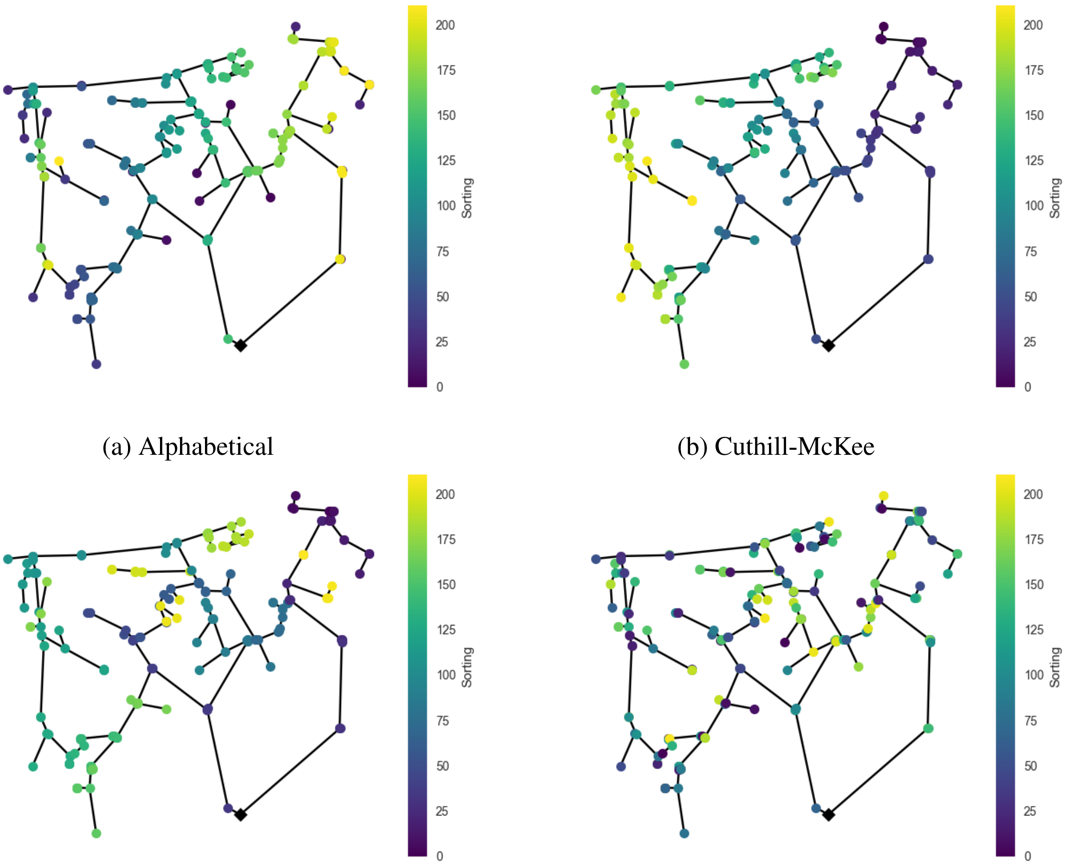
5.3.2.5. Rearranging the search space

The different sortings of the network nodes—Alphabetical, Cuthill-McKee, DFS and Random (see Section 5.2.3.2)—are depicted in Figure 5.13. The colors correspond to the rank of the nodes according to the specific order. The colors are related to the closeness of points to each other in the parameter space. Subsequently, the fitness maps for each possible combination of ordering and distance metric are calculated for the leak with leak outflow of 1 l/s with $c_e = 0.12$ at the position which is marked in Figure 5.6. The position of the leak in landscape plots is marked with a red circle in Figure 5.14.

Figure 5.14 shows that the different search space sortings illustrated in Figure 5.13 have a strong influence on fitness landscape's shape. Obviously, more random sortings of the axis result in more local minima. This occurs for alphabetical sorting in Figure 5.14a or random sort in Figure 5.14b. Optimization algorithms are expected to perform worse using this sorting algorithms by converging slower to the global optimum and have a higher likelihood to restrain in a local minimum. Sortings introduced by the Cuthill-McKee algorithm in Figure 5.14c or the DFS algorithm depicted in Figure 5.14d clearly have a smoothening effect on the fitness landscape of the model-based leak localization problem. The more distinct the global optimum in the fitness landscape, the better is the leak localization algorithm supposed to work. Hence, better convergence properties are expected for Cuthill-McKee and DFS algorithms, overruling alphabetical and random search space orders.

5.3.2.6. Performance evaluation of sorting algorithms

The performance of the algorithm is evaluated similarly as in Section 5.3.2.3. Again, DE is used in the DE/rand/1 formulation with $F_1 = 0.5$, $CR = 0.7$, a population size of $\mu = 30$ and $N_I = 100$ iterations (see Section 2.1.2.2.2). Since DE is stochastic, 200 optimization runs are performed and the results are statistically evaluated over all runs. Again, d_T and c_e convergence over fitness function evaluations are taken as performance measure. Contrary to Section 5.3.2.3, the leak is always at the same position and not randomly distributed over the whole network this time. 3000 hydraulic function evaluations are performed in each optimization run. In total, 16.8 million EPANET simulations were performed



(a) Alphabetical

(b) Cuthill-McKee

(c) DFS

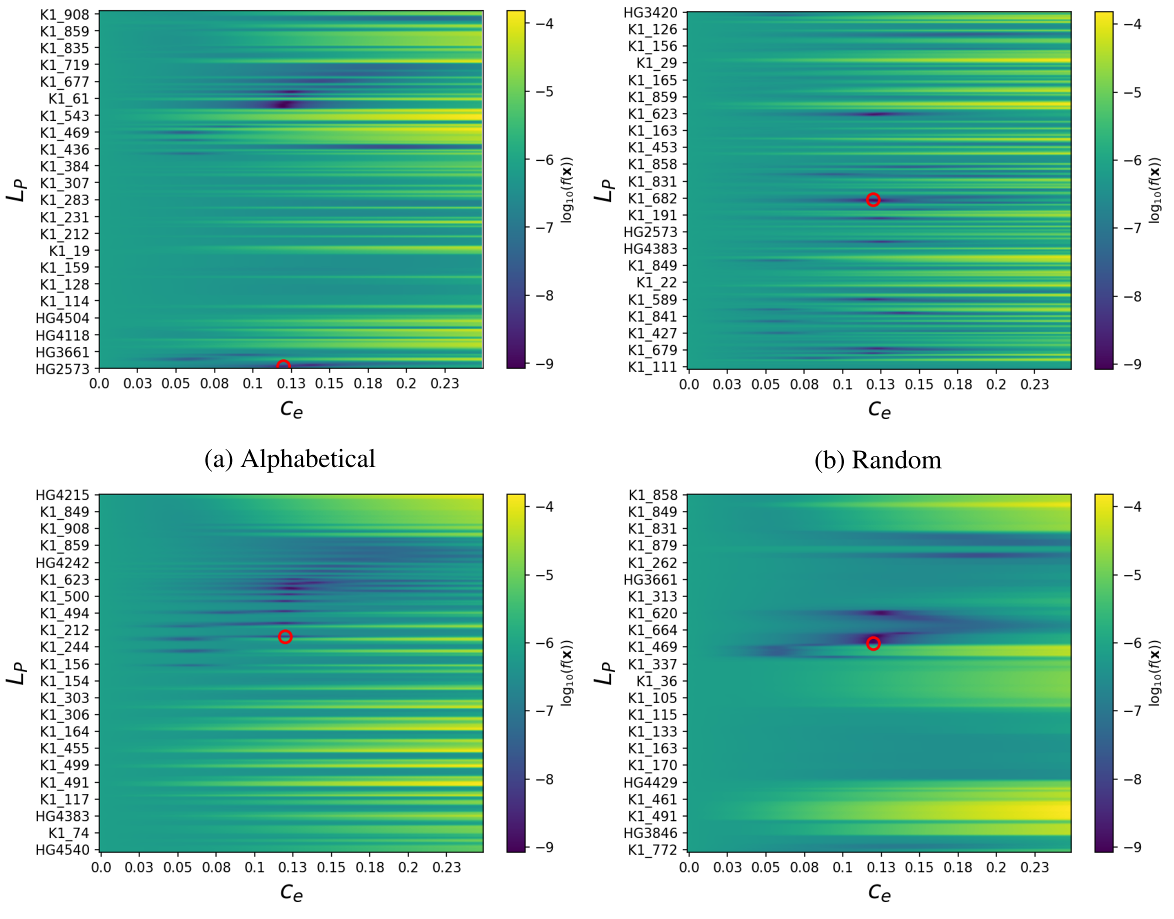
(d) Random

Figure 5.13.: Different sortings in the Ragnitz network

for the results presented in this section. Only selected combinations and not all results for distance metrics-orderings pairs are depicted for the sake of brevity.

Figure 5.15 shows the convergence plots for the topological distance d_T as a function of the number of $f(\mathbf{x})$ evaluations. The results are averaged over 200 optimization runs for different search space sortings and distance metrics. The corresponding leak localization problem is depicted in Figure 5.6. Specifically, Figure 5.15a shows the simulation results for Euclidean metric and the different sorting algorithms and Figure 5.15b shows results for Cuthill-McKee ordering and different metrics. This ordering was chosen, because it showed the best results for the Euclidean metric in Figure 5.15a.

Figure 5.15a clearly shows that Cuthill-McKee and DFS performs better than alphabetical ordering. Notably, random ordering converges very fast, too. Probably, the local optima in the random parameter space order are close enough to each other that the DE algorithm jumps from one optimum to another. In the alphabetical order, the optima are farer apart from each other. Note that this was found for Euclidean and Sorensen metric only—other metrics did not show the same behavior for the random parameter sorting. Cuthill-McKee performed the best over all metrics. This explains why Figure 5.15b shows the performance of different metrics in this ordering. Again, as already found in Section 5.3.2.3, lower p parameters of the Minkowski metric showed better convergence also in the real-network. The Canberra metric performed the worst and Sørensen and cosine metric performed as well as small p



(a) Alphabetical

(b) Random

(c) Cuthill-McKee

(d) DFS

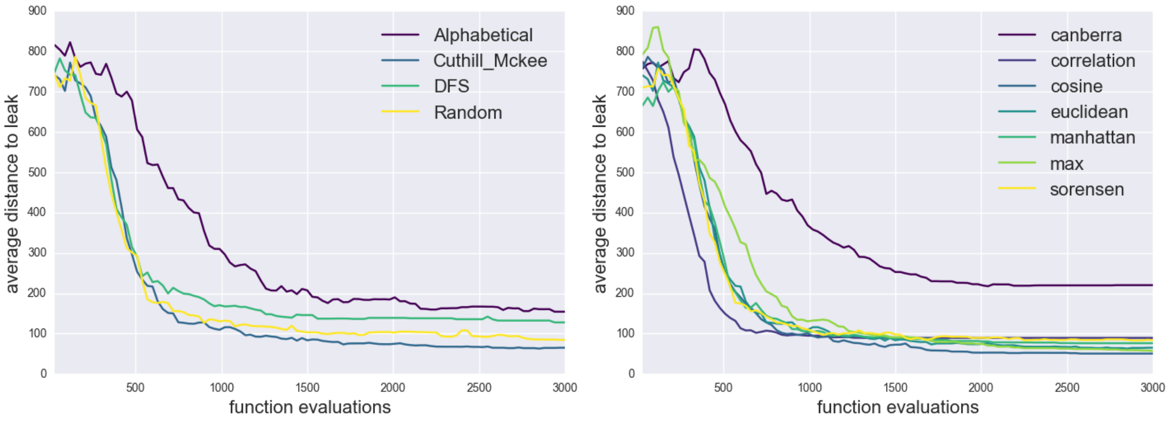
Figure 5.14.: Fitness landscape resulting for real-network with correlation distance metric. The leak's position in the fitness landscape is marked with a red circle.

Minkowski. However, the correlation metric performed best.

Furthermore, after approximately 500 function evaluations, 70 % of the found leak locations by the correlation distance metric are in the proximity of the real leak with almost 90 % in an area with a $d_T < 400$ meter . This result is closer than all results obtained by the other metrics. Finally, after 3000 function evaluations all distance metrics converged to almost equally good results. Only the Canberra metric performed worse.

5.3.3. Performance of sensor placements in the real-world

In Chapter 4, the question of which sensor placement algorithm delivers the best results was postponed to this chapter. Now, a potential answer to this question is proposed based on evaluation of different sensor placement algorithms in the real-world. Pressure sensors were installed at the optimal sensor positions at hydrants in the Ragnitz network (see Figure 2.21) as computed with the six different optimal sensor placement algorithms described in Section 4.2.2. This is the list of the different algorithms and their abbreviations used in this section



(a) Euclidean metric

(b) Cuthill-McKee ordering

Figure 5.15.: d_T convergence plots for different metrics and different search space orderings

- **SP1**: Shortest Path 1 algorithm as described in Section 4.2.2.2.1 and Schaetzen et al. 2000
- **SP2**: Shortest Path 2 algorithm as described in Section 4.2.2.2.2 and Schaetzen et al. 2000
- **SHA**: Shannon Entropy algorithm as described in Section 4.2.2.3.1 and Schaetzen et al. 2000
- **PER**: Pérez binarized-sensitivity matrix algorithm as described in Section 4.2.2.3.2 and Pérez, Puig, Pascual, Peralta, et al. 2009
- **CAS**: Casillas sensitivity projection algorithm as described in Section 4.2.2.3.3 and Casillas, Puig, et al. 2013
- **SPU**: Sensor placement algorithm considering demand uncertainties as described in Section 4.2.2.4 and Steffelbauer and Fuchs-Hanusch 2016a

The present data from the calibration night is used for benchmarking the sensor placement algorithms. On the one hand the small artificial leak scenarios—scenarios S_{15}, S_{16} and S_{17}—are used. The positions of these artificial leaks are positions A, E and F in Figure 2.22 for scenarios S_{15}, S_{16} and S_{17}. The measurement values for the inflow and pressure measurements for scenario S_{15} are illustrated in Figure 2.29. Similarly, scenario S_{16} is depicted in Figure 2.30, respectively, S_{17} can be found in Figure 2.31. The leak localization problem is formulated in the same way as described in Section 5.2.5, also with the simplification as described in equation (5.19), since the leak outflow is estimated as difference between the MNF and the inflow measurements at the corresponding hydrant opening times. The corresponding differences $\Delta\hat{\boldsymbol{\mu}}_t$—the measurement residual as stated in equation (5.16)—can be found in Table 2.8.

On the other hand, scenario S_4 is used. Because no third flow measurement device for measuring the outflow from the hydrants was available in the calibration night, the data of this scenario was not used for calibrating the hydraulic network. Therefore, it can be used at this point as artificial leak data simulating a big pipe burst with an outflow of more than 11 L/s. An algorithm that is capable of finding the position of a small leak should also be applicable on bigger pipe bursts. The measurement curves for the inflow and the twelve pressure measurement devices for scenario S_4 can be found in Figure 2.23. The scenario was generated by opening the hydrant at position B in Figure 2.22. The corresponding opening times can be found in Table 2.2. The corresponding differences $\Delta\hat{\boldsymbol{\mu}}_t$—the measurement residual as stated in equation (5.16)—are presented in Table 2.4 and Table 2.5. Scenario

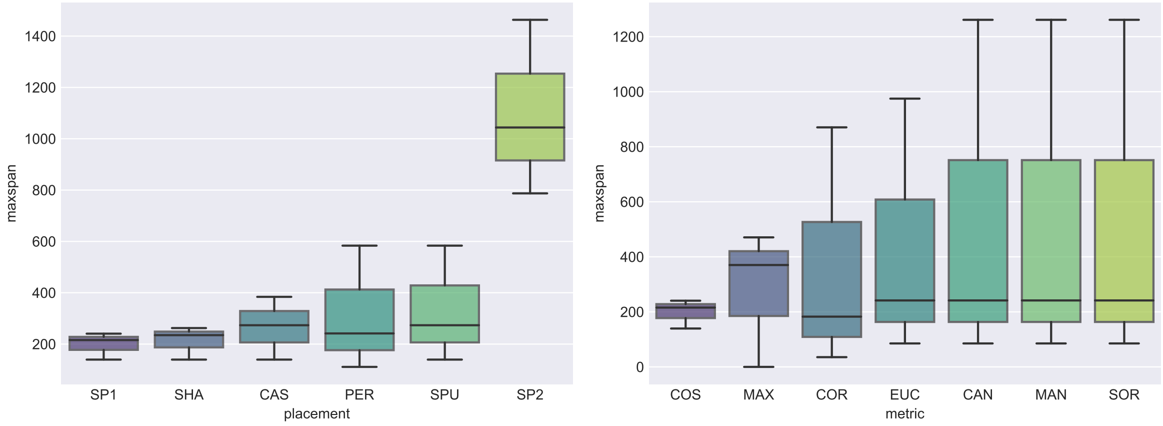
S_4 is not used for evaluating the different placements and metrics, it is only used for proving the method's applicability for big pipe bursts.

To evaluate the localization efficiency of the different metrics and placements, the first three performance measures defined in Section 5.2.6 are considered—the false positves FP, the topological distance between the found and the real leak d_T and maximum span of the false positives MS. The fourth measure is not taken into account since the estimate of the leak magnitude is used to reduce the complexity of the problem.

The results for all metrics and all placements averaged over scenarios S_{15}, S_{16} and S_{17} can be found in Table 5.4. The abbreviations for the metrics are following

- CAN - Canberra metric in equation (5.12)
- COR - Correlation metric in equation (5.10)
- COS - Cosine metric in equation (5.9)
- EUC - Euclidean metric in equation (5.7)
- MAN - Manhattan metric in equation (5.6)
- MAX - Maximum metric in equation (5.8)
- SOR - Sørensen metric in equation (5.11)

The best values for FP, d_T and MS are marked with boldface. Additionally, the results are sorted according to the FP parameter, thus, the best combinations of placement and metric can be found on the top of Table 5.4.



(a) Different placements of cosine metric.

(b) Different metrics for placement SP1

Figure 5.16.: Performance evaluation plots using the maximum span of different metrics and different placements for real-leak scenarios.

The **cosine metric** performed best in the FP and the MS criterion, **maximum metric** best in d_t. The best sensor positions resulted for the **Shortest Path 1** placement followed closely by Shannon entropy. Hence, the results for this metric and this placement are investigated in more detail. Box-plots for the cosine metric's performance measure MS over all three scenarios distinguished by the placements can be found in Figure 5.16a. A similar plot for Shortest Path 1 placement over all metrics can be found in Figure 5.16b. Yet, the box plots have to be considered with some reservation, since only three scenarios are used for the plots. Hence, the upper edge of the box plots represents the worst scenario performance, the lower part the best, and the median value depicted by the big black line in the middle of each box plot are the median performing scenario. The whiskers—representing the interquartile range—are basically meaningless. Despite the poor statistics, box plots were chosen for this qualititave analysis due to their visual ease in comparison with line or scatter plots.

The results do not indicate, however, that Shortest Path 1 algorithm delivers the best sensor positions compared to other algortihms. This is due to the following reasons: First of all, only three leak scenarios are evaluated. Second, at the moment when the sensor positions were computed, nobody was aware of the partially closed valve in the system. The real nature of the problem was only identified after extensive analysis of the calibration data. As Blesa et al. 2014 already mentioned, this has a high influence on sensitivity-based sensor placement algorithms, hence, the computed positions for algorithms SHA, PER, CAS, SPU might be far from optimal. Third, the differences between the placements are not very distinct, as can be seen in Figure 5.16a. Examining other leak scenarios might find other algorithms to perform better (Fuchs-Hanusch and Steffelbauer 2017). As a conservative interpretation, at least for these three leak scenarios and for this WDS, the Shortest Path 1 algorithm combined with cosine (or maximum) metric performed the best. This supports the picture from Table 4.7 where Shortest Path 1 algorithm also performed best considering F_1 and the overall F value. Additionally, the leak size was estimated from the measurement data. If the leak size becomes an optimization parameter, other metrics may lead to better results. Especially metrics with known potential in converging fast to the right leak outflow levels like, for example, the maximum metric (see Figure 5.11b) that performed already well in this example in finding the right leak spot. Hence, no decision can be made which metric performs best, but cosine and maximum metric are favored. However, one conclusion can be drawn: Shortest Path 2 algorithm, despite being quite similar in finding the optimal positions as Shortest Path 1, clearly performs worst.

The localization results for the cosine metric and Shortest Path 1 sensor placement algorithm are also depicted in Figure 5.17. The X marks the location of the real leak and the O marks the position that the algorithm has found. Additionally, the objective function values for all nodes are depicted through the coloring of the nodes at a logarithmic scale.

Figure 5.17a shows the results for the simulated big pipe burst in Scenario S_4. The d_T resulted in a distance of 109 meter. The leak is at a unmeasured dead-end branch. Additionally, the roughness values at this area of the network are very low (see Figure 2.26 respectively Figure 2.27 or Figure 2.25 and Table 2.6). Hence, the algorithm can not distinguish between the turnoff branch and the main pipe. The small roughness values additionally affecting the result to increase distance to the turnoff point. Interestingly, in spite of having the biggest leak outflow, this scenario leads to the worst false positive rate of $FP = 3.24\%$ and widest maximum span of $MS = 412$ meter of all examined scenarios, although, the results are still very satisfying.

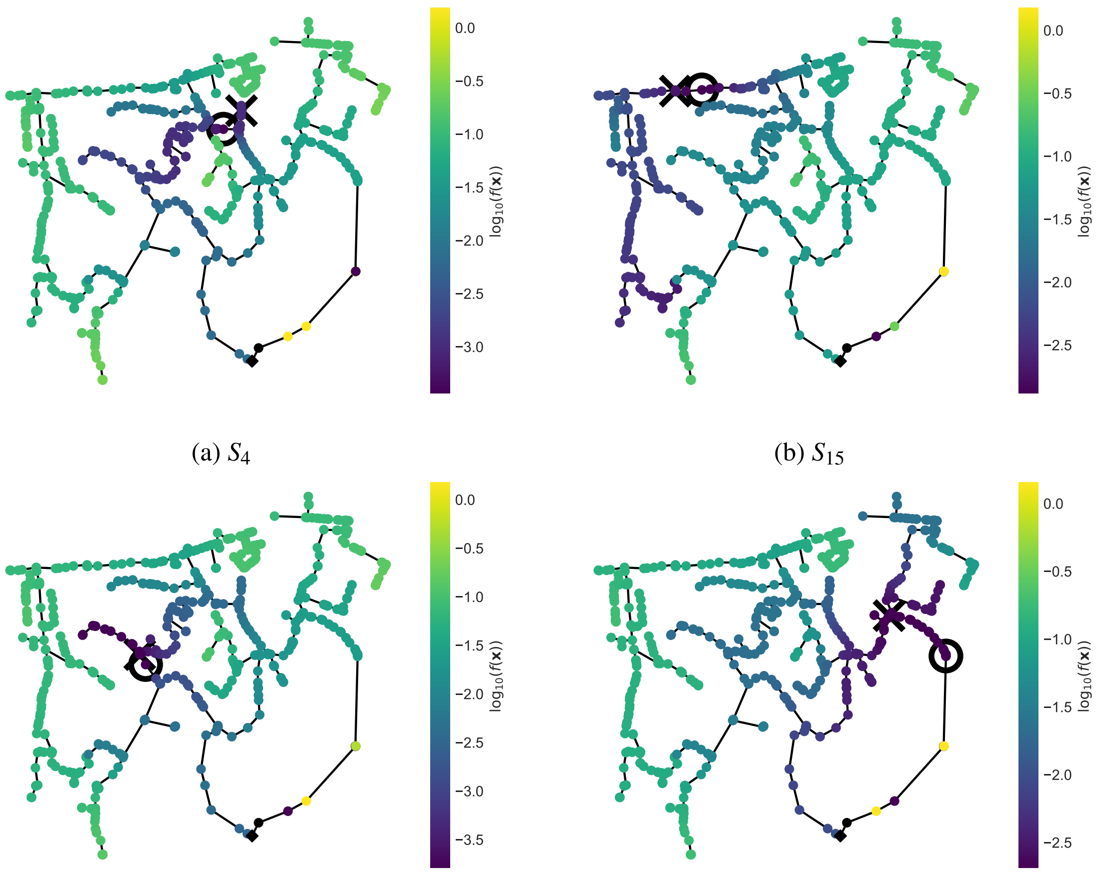
The results for scenario S_{15} are shown in Figure 5.17b. This scenario has the smallest leak outflow of $Q_L = 1.18$ L/s, but results in the best false positive value of $FP = 0.77\%$ and the smallest maximum span of $MS = 140$ meter. The distance between the estimated and the real leak is $d_T = 85$ meter.

Scenario S_{16} is depicted in Figure 5.19c and results in the smallest distance of $d_T = 41$ meter with $FP = 1.39\%$ and $MS = 216$ meter.

Finally, the last scenario S_{17} is depicted in Figure 5.19d. This scenario led to the least precise—but still acceptable leak localization performance compared for example to literature—of $FP = 1.54\%$, $d_T = 241$ and $MS = 241$ meter for all small leak scenarios. The found leak position is again situated at

Table 5.4.: Evaluation of the performance of different metrics and sensor placement algorithms averaged over leak scenarios S_{15} to S_{17}

	Placement	Metric	FP	d_T	MS
1	SP1	COS	**1.23**	122.43	**198.56**
2	SHA	COS	**1.23**	120.47	212.23
3	CAS	COS	3.39	85.15	265.66
4	SP1	MAX	2.31	**46.99**	280.31
5	PER	COS	3.24	125.47	312.03
6	SPU	COS	3.80	130.53	332.17
7	SHA	MAX	3.29	97.36	343.22
8	SP1	COR	6.63	123.48	351.03
9	SHA	COR	6.57	134.63	362.01
10	CAS	SOR	6.37	243.88	384.35
11	CAS	MAX	5.19	206.72	392.87
12	CAS	CAN	6.11	243.88	393.67
13	CAS	MAN	6.42	241.80	400.71
14	CAS	EUC	5.24	249.80	400.71
15	SPU	MAX	6.21	246.43	403.49
16	SPU	EUC	6.78	254.34	426.10
17	SP1	EUC	7.19	177.86	433.88
18	SPU	MAN	7.76	212.84	433.88
19	SPU	SOR	8.99	177.98	446.96
20	SPU	CAN	9.04	243.58	455.55
21	CAS	COR	7.70	114.77	463.38
22	SHA	EUC	5.39	181.49	471.94
23	PER	COR	6.16	100.88	519.85
24	SP2	SOR	6.21	221.93	525.49
25	PER	CAN	7.86	168.61	529.33
26	SHA	MAN	7.45	200.76	529.33
27	SP1	SOR	7.55	203.59	529.33
28	SHA	SOR	7.19	221.82	529.33
29	SP1	MAN	7.65	197.59	529.33
30	PER	MAN	7.60	168.61	529.33
31	PER	SOR	7.60	168.61	529.33
32	SP1	CAN	7.50	241.65	529.33
33	SHA	CAN	7.19	221.82	529.33
34	PER	EUC	7.50	170.09	529.33
35	SP2	MAN	9.76	254.19	602.88
36	PER	MAX	9.09	209.50	606.36
37	SP2	CAN	12.48	221.93	650.54
38	SPU	COR	11.35	177.95	804.89
39	SP2	EUC	15.87	425.42	891.69
40	SP2	MAX	22.03	433.34	1072.56
41	SP2	COS	18.03	570.59	1098.17
42	SP2	COR	34.77	1288.73	2169.54


0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
$\log_{10}(f(\mathbf{x}))$
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
$\log_{10}(f(\mathbf{x}))$
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
−3.5
$\log_{10}(f(\mathbf{x}))$
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
$\log_{10}(f(\mathbf{x}))$

(a) S_4

(b) S_{15}

(c) S_{16}

(d) S_{17}

Figure 5.17.: Leakage localization results for cosine metric and shortest path 1 sensor placement for different scenarios

an unmeasured dead-end branch with small roughness coefficients, hence, the same arguments as for S_4 apply: the sensors are not able to distinguish between the main pipe and the dead-end branch and this effect is exaggerated through small roughness values. This leads to the finding that the localization of this leak is more acceptable than the plain numbers suggest,

The results for all scenarios are summarized in Table 5.5.

5.3.4. Measuring the cost-benefit sensor placement curve

The Shortest Path 1 algorithm has an additional advantage. Since it is a "greedy" sensor placement algorithm by construction, it allows for examining the sensor placement performance for all placement consisting $N-i$ sensors if N is the number of the actually placed sensors. Subsequently, the performance as a function of the sensor number N allows to measure a cost-benefit function in the real-world. This is done on the example of the maximum metric and the FP performance averaged over leak scenarios S_{15} to S_{17}. The results for the different numbers of sensors can be found in Table 5.6. The FP values as a function of N are depicted in Figure 5.18. Again, the cost-benefit function

Table 5.5.: Performance evaluation of cosine metric and Shortest Path 1 placement for different leak scenarios

	FP	MS	d_T
S_4	3.2	412.2	109.0
S_{15}	0.8	139.5	85.3
S_{16}	1.4	215.6	41.4
S_{17}	1.5	240.6	240.6

has a power-law behavior

$$f(N) = a \cdot N^{-b} \quad . \tag{5.23}$$

The fitting parameters result in $a = (0.005811 \pm 0.000003)$ and $b = (0.4393 \pm 0.0009)$. Of course, the sample size with three leak scenarios is again very small.

Table 5.6.: Performance evaluation of maximum metric and Shortest Path 1 placement for different numbers of sensors

N	FP	d_T	MS
2.0	7.1	214.6	442.5
3.0	6.0	23.7	496.4
4.0	5.1	109.8	496.4
5.0	4.8	48.8	427.4

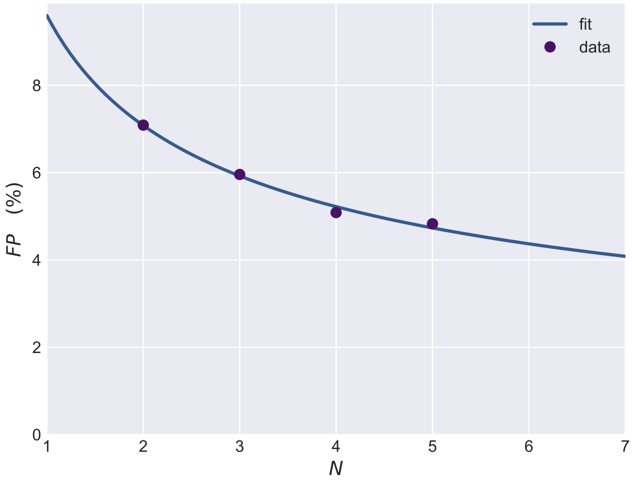
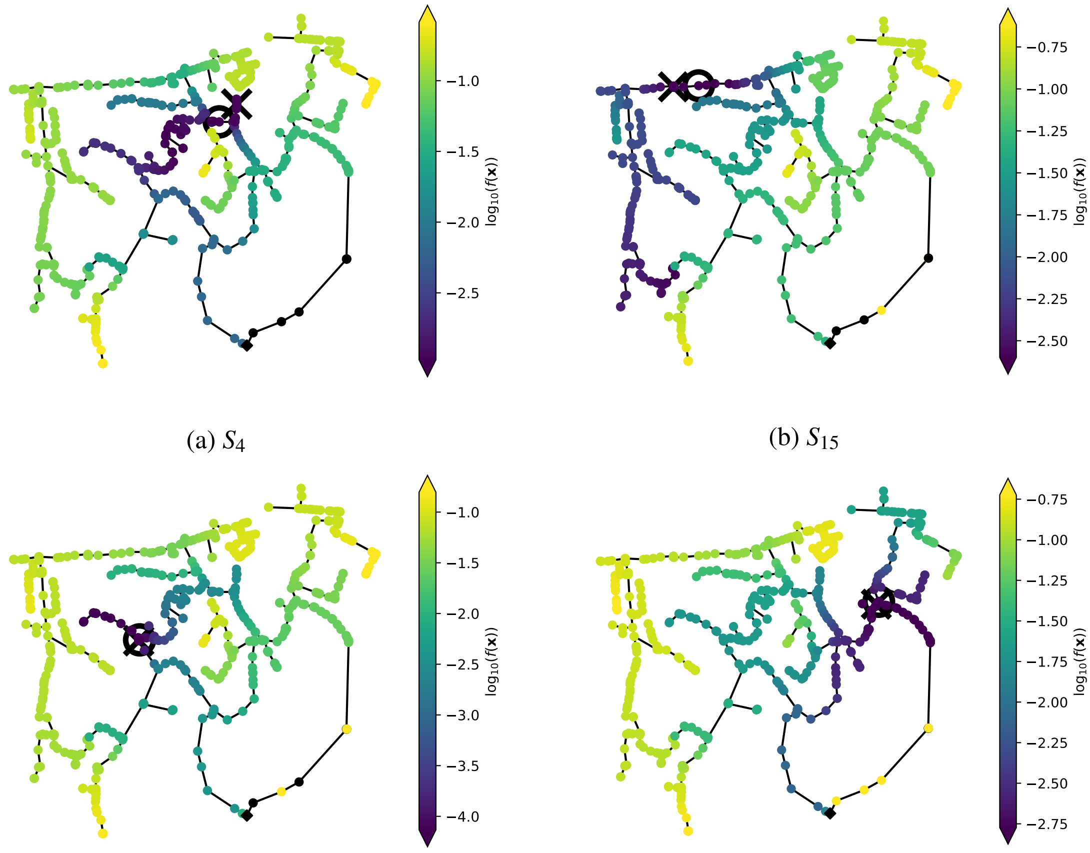


Figure 5.18.: Real-world sensor cost-benefit curve

5.3.5. Leak localization with pressure sensors only

For the previous leak localization real-world experiments, no optimization algorithm was required. The leak outflow was estimated using the differences in the inflow before and after the leak occurred.



(a) S_4

(b) S_{15}

(c) S_{16}

(d) S_{17}

Figure 5.19.: Leak Localization with pressure sensors only for various scenarios

Thus, the problem was one-dimensional since the leak position had to be found. This was achieved through total enumeration by simulating a leak at all nodes with the estimated leak size Q_L. The node with the minimum objective function value was considered as the leak node.

For the simulations, it is assumed that no inflow measurements is used. The question is if the leak can be found using pressure sensors only?

For that reason, a two stage optimization approach is used. First, DE is used once with a metric that has the property to converge fast to the right leak size, e.g. the maximum metric—utilizing only the pressure measurements. Through this single optimization run the probable leak size $\hat{Q}_L^{DE}$ is estimated. Additionally, a first estimate for the leak location is also retrieved with a corresponding distance to the real leak d_T^{DE}. $\hat{Q}_L^{DE}$ is then used for simulating a leak at every point in the system to obtain its location, similar to the approach in the previous section. For this simulations, a metric with good leak localization performance is used, e.g. cosine metric. For the results of the second optimization round, again, d_T, FP and MS is computed.

This leak localization approach using exclusively pressure information is simulated for the scenarios in the previous section ($S_4, S_{15}, S_{16}, S_{17}$). Only sensors at the Shortest Path 1 sensor placement are used. Maximum metric and Cuthill-McKee parameter space sorting is used for simulations with DE in the DE/rand/1 formulation with a population size of $\mu = 50$, $N_I = 50$ iterations, $CR = 0.5$ and $F_1 = 0.8$. The leak is simulated as additional demand at the leak node with search space boundaries from zero to 20 L/s. The total two stage optimization takes around three minutes, hence, the leak localization works in real-time supposing that every 15 minutes a new measurement value is retrieved.

The results can be found in Table 5.7 and Figure 5.19. The real leak location is marked with an X, the location found by the first stage of the optimization is marked with an O. The colors resemble the objective function values at the nodes using the cosine metric. Remarkably, the measured leak size Q_L and the leak size estimated by the optimization algorithm using only the pressure sensors $\hat{Q}_L^{DE}$ coincide very well. Additionally, the estimated leak location of the DE algorithm d_T^{DE} is equally good as using the cosine metric for scenario S_4 and S_{15} and even better for S_{16} and S_{17}. This can also be seen in Figure 5.19 comparing the X and the O's position and the node colors.

Consequently, the results show that model-based leak localization can be accomplished with pressure sensors only for leaks around 1 L/s and for larger pipe bursts. In Chapter 3 we have already seen that leak detection can be achieved with pressure sensors. Thus, pursuing research in this direction potentially saves WUs a lot of money, because flow sensors are more expensive and possess higher installation costs and maintenance expenses than pressure sensors.

Table 5.7.: Results for leak localization for diverse scenarios considering the pressure measurements only

Scenario	Q_L	$\hat{Q}_L^{DE}$	d_T	d_T^{DE}	FP	MS
S_4	11.49	12.08	108.97	108.97	5.08	428.31
S_{15}	1.17	1.18	85.29	85.29	1.08	139.53
S_{16}	1.14	1.27	143.00	4.43	0.62	143.00
S_{17}	1.34	1.35	27.37	13.32	1.08	65.49

5.3.6. Localization of small leaks

The model-based leak localization methods are also tested on smaller leaks with leak outflows of less than one liter per second $Q_L < 1$ L/s. For that reason, small leak outflow scenarios were generated in the case study area Graz Ragnitz. The artificial generated scenarios had long run-times of several days (see Table 2.9). In total, twelve different scenarios were generated with different leak outflows at four different positions—depicted in Figure 2.32—and leak run-times varying between 47 and 188 hours. The targeted leak outflows were between $Q_L = 0.25$ L/s and $Q_L = 1.0$ L/s. During the first ten scenarios, twelve pressure sensors were installed at different locations in the system (scenarios S_a to S_j). During the last two scenarios, six pressure sensors were still installed in the system, after a period, where the leak free system was investigated with those six sensors.

Unfortunately, for the first scenario, a storage buffer overflow in the pressure sensors deleted the measurement data, hence, S_a could not be used for the leak localizations. That is why scenario S_d was generated with the same outflow and position.

Additionally, two sensors of the remaining six sensors showed malfunctions. It could not be reconstructed, when these sensors started to deliver faulty measurement signals, hence, the sensors HG3420 and HG4215 are neglected for localizing small leaks.

Even more unluckily, the measurements of the unperturbed system proved to be very valuable for detection and localization. Without these measurements, no reliable reference pressure signals can be constructed—used for computing the residuals between the leak-free and the leaky system (see equation (5.16)). That is why only four sensors—six installed sensors during the longer leak free period minus the two malfunctioning sensors—could be taken for this leak experiments. Unfortunately, this sensor positions did not coincide with any optimal sensor placement algorithm.

The residuals are computed as the differences between the simple time series model $\hat{x}_t^S$ (see Section 2.3.5.3) and the actual measurements x_t. Subsequently, the mean of the residuals $\Delta\hat{\boldsymbol{\mu}}_t$ is taken over the whole time of the leak scenario . The results can be found in Table 5.8. The more complicated time series model—constructed by additionally extracting the correlations from the residuals with an ARMA model—can not be used since no leak free period exists before the leaks, which is necessary to fit the model to the measurements.

Table 5.8.: Residuals $\Delta\hat{\boldsymbol{\mu}}_t$ between the measurements during the leak scenarios x_t and the forecasted values from the simple time series model $\hat{x}_t^S$

	Inflow	HG3835	HG3933	HG4162	HG4383
	ΔQ [L/s]	Δp [bar]	Δp [bar]	Δp [bar]	Δp [bar]
S_b	0.827	-0.023	-0.025	-0.015	-0.030
S_c	0.491	-0.004	-0.005	-0.002	-0.004
S_d	0.544	-0.022	-0.026	-0.015	-0.025
S_e	0.129	-0.005	-0.003	-0.007	-0.005
S_f	0.281	-0.003	-0.001	-0.001	-0.006
S_g	0.828	-0.019	-0.020	-0.013	-0.024
S_h	0.772	-0.018	-0.016	-0.019	-0.017
S_i	0.946	-0.031	-0.028	-0.031	-0.031
S_j	0.280	0.001	0.002	-0.002	0.002
S_k	0.897	-0.023	-0.026	-0.018	-0.027
S_l	0.812	-0.028	-0.018	-0.035	-0.018

The pressure differences in scenarios S_c, S_e, S_f and S_j are decided to be too small for leak localization. Maybe better time series models are able in the future to extract the differences in the pressure signals in a more reliable way. Anyway, these scenarios are neglected during the further investigations. Simulations with these scenarios have shown that they result in FP rates between 25 % and 75 %. Possibly, more sensors at more sensitive positions are able to detect leaks of this magnitudes, but for now, leaks under 0.5 L/s are not locatable in this network with sensors at these positions.

The remaining scenarios—scenario S_b, S_d, S_g, S_h, S_i, S_k and S_l—are further investigated. First, it is tested which metric performs best on small leaks. The results can be found in Table 5.9. Clearly, the maximum metric shows the best localization results for small leaks—followed by the correlation metric regarding the FP rate. Thus, these two metrics are examined in more detail on each leak scenario separately. The results can be found in Table 5.10. Interestingly, scenarios where the maximum metric performs worse are good scenarios for the correlation metric—and vice versa. The mean FP value is around 6 % for the best scenarios considering both metrics. In other words—only a 1/16 of the system has to be searched for leaks in average after applying this method for pre-localizing the leak. These are very promising results—considering the use of the simple time series model as well as the small number of working sensors at non-optimal positions.

Table 5.9.: Performance of all metrics averaged over scenarios S_b, S_d, S_g, S_h, S_i, S_k and S_l

	Metric	FP	d_T	MS
1	MAX	**12.68**	**453.43**	**768.64**
2	EUC	25.07	471.60	1118.60
3	COS	20.10	477.11	1195.09
4	COR	18.27	578.65	1302.02
5	MAN	33.72	649.61	1526.29
6	SOR	34.54	646.33	1555.09
7	CAN	35.09	649.32	1557.52

Table 5.10.: Performance of maximum and correlation metric for different leak scenarios S_x

Metric	Maximum			Correlation			Leak	
Scen.	d_T	MS	FP	d_T	MS	FP	ΔQ	Pos.
S_b	139.53	206.20	**1.23**	1271.09	1527.77	10.02	0.83	I
S_d	578.00	693.35	**12.79**	341.49	1652.62	35.90	0.54	I
S_g	1035.52	1105.79	**6.93**	550.43	1528.73	29.28	0.83	II
S_h	741.56	1200.08	25.42	313.65	441.00	**2.62**	0.77	III
S_i	125.61	1200.76	32.05	372.58	1235.79	**8.47**	0.95	III
S_k	480.11	480.11	**7.09**	275.46	1527.77	16.80	0.90	I
S_l	73.69	494.17	**3.24**	925.85	1200.48	24.81	0.81	IV

Additionally, results using the maximum metric are depicted in Figure 5.20 for all scenarios. The regions of low maximum metric values correspond well with the true leak locations. It can be concluded that model-based leak localization is possible for leak sizes down to 0.5 L/s with satisfactory results.

5.3.7. Fast localization of a small leak

In Chapter 3 the importance of a fast detection of leaks is highlighted. Certainly, it is also important to locate a leak fast, once it is detected. In the previous section we have seen that leak localization is possible considering the whole available time series data of the leak scenarios. Now it will be investigated, if localization is also possible for a shorter time period and a limited amount of data.

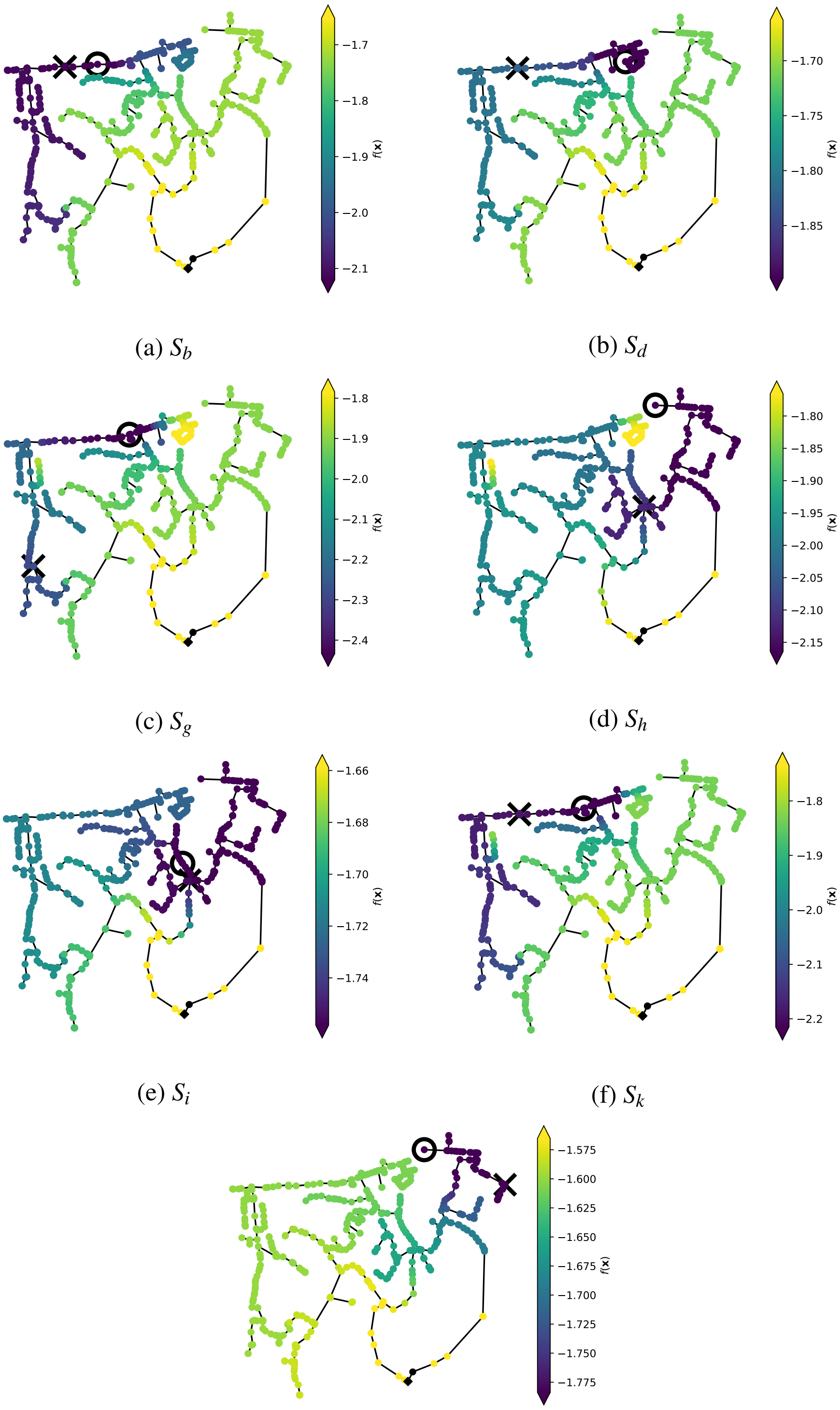
Hence, the short leak period that was already employed in Section 3.3.5.5 is used once more. The leak is the same as in scenario S_k. The leak was opened on the 4 of July at 12:20. It has been detected at 15:15. Consequently, the data until the detection is used for the localization. This will be called from now on the short period P_S. The data obtained from the simple time series model x_t^s is used, since the complex model x_t^c underestimates the leak outflows and the pressures—having a negative effect on the leak localization. The inflow and pressure data resulting from the Bayes MCMC approach described in Section 2.3.2 and 2.3.6.3 can be found in Table 3.10.

Additionally, leak localization simulations are performed for the long time period P_L until noon at the 8^{th} of July. This data can be also found in Table 3.10. Furthermore, estimates obtained from the Likelihood-ratio detection method P_R are also used for model-based leak localization. The purpose of investigating the likelihood-ratio is to test, if the estimated values deliver good results for subsequent leak localization.

The simulation results for the maximum metric for all studied periods can be found in Table 5.11. It can be seen that a fast localization of the leak is possible with a FP rate of 7 %. Furthermore, the distance d_T to the real leak decreases by taking longer measurement times into account. Additionally, the likelihood-ratio method leads to very good results with a final distance to the leak—in maximum metric—that is comparable to the distance found by the correlation metric in Table 5.10. Although, it leads to better MS and FP values. The results for the fast model-based leak localization using the maximum metric can be found in Figure 5.21.

Table 5.11.: Performance evaluation for fast localization of the leak in scenario S_k right after its detection

	d_T	FP	MS
P_S	438.35	7.09	480.11
P_L	317.76	7.09	1250.48
P_R	275.46	7.70	522.40



(a) S_b

(b) S_d

(c) S_g

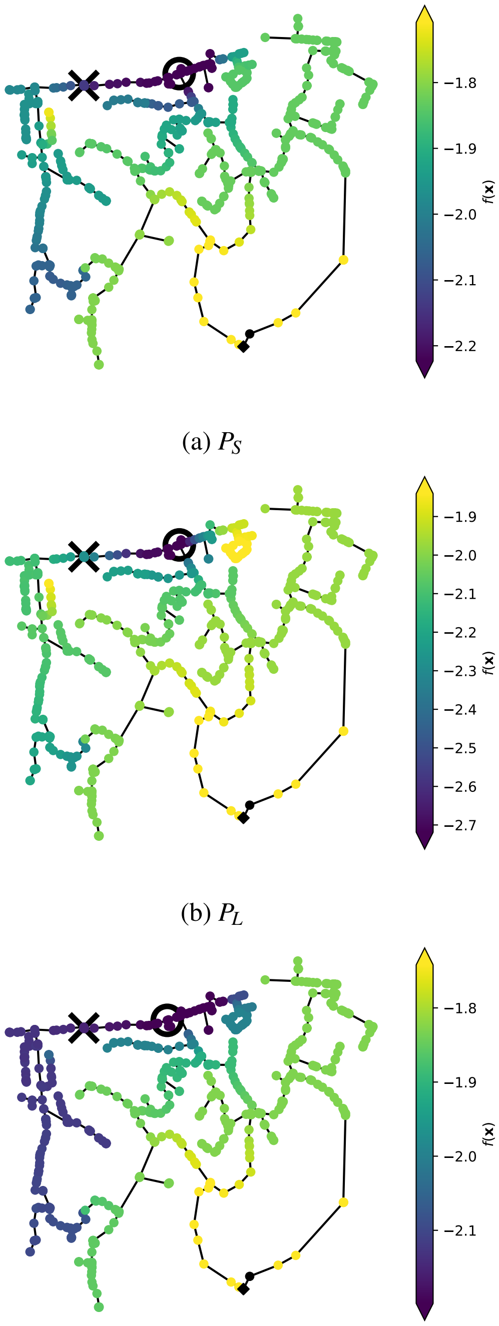
(d) S_h

(e) S_i

(f) S_k

(g) S_l

Figure 5.20.: Plots for small leak scenario with maximum metric



(a) P_S

(b) P_L

(c) P_R

Figure 5.21.: Maximum metric results for fast localization of the leak in scenario S_k for the short P_S and the long P_L period as well as the likelihood-ratio estimates P_R

5.4. Conclusion

This chapter tried to answer following three research questions linked to model-based leak localization:

Q.3.1 What is the actual type (form,shape) of the optimization problem?

Q.3.2 How to increase the performance of model-based leak localization?

Q.3.3 Is model-based leak localization applicable for finding small leaks in the real-world?

Additionally, a research question from Chapter 4 was postponed to be answered within this chapter:

Q.2.1 Which optimal sensor placement algorithm finds the best pressure sensor positions for model-based leak localization?

Till now, most scientific work exclusively focused on model-based leak localization in simulations. Only few studies applied leak localization on real-world examples trying to locate artificially generated or real leaks—that happened accidentally in a WDS while pressure and flow measurements were recorded. Usually, the leak sizes were large and the number of hydraulic sensors was high (see Table 5.1). No work till now tried to localize real-leaks with magnitudes smaller than $Q_L = 1.5$ L/s. This thesis is the first that tried to tackle the problem for leak sizes around 1 L/s and lower in a real-world system by utilizing hydraulic sensors (pressure and flow). This is not trivially since smaller leak sizes lead to exponentially smaller pressure responses and, thus, are consequently harder to locate.

Model-based leak localization is formulated in this thesis as an inverse optimization problem where, in general, two parameters have to be found: (i) the leak location and (ii) the leak magnitude—if not known in advanced through, for example, estimations of the change in the inflow measurements caused by the leak.

What is the actual type of the optimization problem? This question is answered through exhaustive analysis of the fitness landscape of the problem. Due to the parametrization—with continuous leak sizes and discrete leak locations[32]—it is a mixed-integer optimization problem. Furthermore, the fitness landscape analysis has revealed—for the first time ever in water related scientific literature—that the problem is in fact multi-modal. The less sensors are used—the more local optima evolve in the optimization problem. Consequently, deciding on using meta-heuristic algorithms (like DE) to find the leak is the right choice.

Furthermore, it was found that the leak's exact location is not traceable anymore. The leak can only be found in its proximity. Nevertheless, in practice, this is not a big problem since leak localization should always be followed by pinpointing methods. Of course, the better the approximated location of the leak coincide with the real leak, the faster the pinpointing of the leak gets—and consequently less labor and cost intensive it is.

How to increase the performance of model-based leak localization? The fitness landscape analysis revealed some potential to enhance the performance of the optimization algorithms significantly. First, different metrics can be used to express the objective function. Second, the search space can be rearranged prior to optimization. Both approaches make the optimization problem smoother in the

[32] . . . connected to the possible leak nodes in the hydraulic model of the WDS

solution space, hence, easier to find the global optimal solution for algorithms. This results in qualitatively better solutions found in less time. Although, the biggest improvement comes from combining the TSA methods introduced in Chapter 3 with the optimizer as described in Section 5.2.5. Estimating the leak magnitude prior to optimization reduces leak localization to a one-dimensional problem. Consequently, only as many hydraulic simulations as possible leak locations in a WDS are necessary to find the leak—no expensive optimization problem has to be solved anymore. This reduces the computation time to less than a minute for real-world systems and enables real-time localization of leaks.

Which sensor placement algorithm finds the best measurement locations for leak localization? The answer to this question has been postponed from the previous to this chapter. Interestingly, no scientific publication exists that already compared different sensor placement algorithms—neither in simulations nor in real case studies. This is the first time to the best of my knowledge that different sensor placement algorithms are compared, and that all in a real-world study. The different placements were tested on artificially generated leak scenarios through hydrant openings with leak outflows less than 1.3 L/s. Additionally, different distance metrics were tested on their leak localization performance. It was found that the Shortest Path 1 algorithm revealed the best results—the best metrics were found to be cosine and maximum metric. However, the results have to be taken with great caution: A partially closed valve certainly influenced the performance of sensitivity-based sensor placement methods, because their optimal positions have been calculated without this valve. Furthermore, every sensor placement algorithm and every fitness metric resulted in good localization performances—except for the Shortest Path 2 placement which clearly performed the worst. Obviously, the sensor at hydrant HG3933—shared by all placement despite Shortest Path 2—has a great influence on the performance for locating leaks. Additionally, the Shortest Path 1 placement enabled to actually measure a cost-benefit curve in the real-world, experimentally approving the theoretical considerations in Section 4.2.3.

Is the method applicable for finding small leaks in the real-world? The model-based leak localization method worked very well in the real-world case study. For leaks with a magnitude that was a little bit higher than 1 L/s, very promising results where obtained in the calibration night. Moreover, the false positive rates for the small leaks were still very good despite all experienced drawbacks, e.g. faulty sensors and partially closed valves. Note that a false positive rate of 10 % directly translates to a narrowed down possible leak area covering 10 % of the original system size. Thus, the leak is located ten times faster in average than by using conventional methods. Furthermore, the algorithm was able to locate leaks with a magnitude of down to $Q_L = 0.5$ L/s in the experiments, which is the smallest absolute leak size ever found by model-based algorithms till now. Improvements in the modeling approaches for both, time series models as well as hydraulic models, and more accurate pressure sensors might lower this barrier further in the near future.

Finally, the most surprising result in this chapter is that leak localization showed to be possible—and actually very accurate—utilizing only pressure measurements. Previously, it is shown in Chapter 3 that leak detection is also possible with pressure sensors. This has the potential to drastically reduce the installation costs of the proposed methods if WUs can withdraw the expensive permanent real-time inflow measurements—with sensor costs of up to 30 times more than a single pressure sensor. Definitely, this has to be investigated in future studies before final conclusions on this topic can be drawn.

Chapter 6

Conclusion

“To summarize the summary of the summary: people are a problem."

— Douglas Adams, *Life, the Universe and Everything*

Answers to the research questions and key findings This thesis tackled three distinct, but interconnected topics related to water losses in WDSs: (i) Leak Detection—techniques to become aware of a leak in a system shortly after it occurred, (ii) Optimal Sensor Placement—approaches to find optimal measurement positions which are sensitive to leaks whilst simultaneously being robust against uncertainties and (iii) Leak Localization—methods to obtain a first approximation of the leak's location right after it's detection to increase time efficiency for finding it's exact position. For each of these topics specific research question arose (see Section 1.5). Here is a concise summary of the respective questions and the proposed answers

(i) **Leak Detection**—Chapter 3

> **Q.1.1:** Is it possible to early detect small leaks (<1 L/s) in the real world with only pressure sensors in an automatic way?

In this thesis different methods were introduced to detect leaks within data obtained from pressure and flow sensors. Most of these methods were based on probability theory, e.g., CUSUM, likelihood-ratio or techniques utilizing Bayesian statistics. Through applying TSA techniques prior to detection—splitting the measurement time series into seasonal, random and trend components—all methods proofed to be reliable and robust in early detecting leaks, also when relying on pressure sensors only. Additionally, methods have been introduced to estimate the start of the leak with high precision. This can be used for automatic accurate water balance computations and to compute the whole state space probabilities of the problem. More advanced parameter estimation techniques can then be coupled with the leak localization techniques. All methods work in a complete automatic way. Through an efficient formulation the computational burden is small enough to allow the algorithms to work in real-time even on normal desktop computers.

(ii) **Optimal Sensor Placement**—Chapter 4

> **Q.2.1:** Which optimal sensor placement algorithm finds the best pressure sensor positions for model-based leak localization?

This question is answered by benchmarking six different optimal sensor placement algorithms in a real-world field study. Unfortunately, a closed-valve was found in the system after installing the sensors. Hence, the positions were no longer optimal for sensitivity-based sensor placement approaches. Nevertheless, the Shortest Path 1 algorithm (see Section 4.2.2.2.1) was found to result in the best leak localization performance. However, the sample size of the real-world leak scenarios was not big enough to make final decisions on this question.

Q.2.2: How to incorporate different sources of uncertainties in sensor placement algorithms to obtain more robust optimal measurement locations?

A novel OSP method—called the SPUDU algorithm in Section 4.2.2.4—was developed capable of incorporating any kind of uncertainties and was studied through the example of incorporating the effect of demand uncertainties on optimal measurement positions. The algorithm showed that it avoids to place sensors in regions of a network where high uncertainties are expected—leading to more robust sensor positions.

Q.2.3: How many sensors are needed for model-based leak localization?

This depends on the particular system, but methods have been developed to answer this questions through applying cost-benefit analysis. It was found that the the quality of an optimal sensor placement on the number of sensors follows a power-law behavior. Furthermore, this has been also observed in real-world experiments in this thesis which confirmed the theoretical considerations.

(iii) **Leak Localization**—Chapter 5

Q.3.1: What is the actual type of the optimization problem?

Fitness landscape analysis revealed that the problem is in fact multi-modal mixed-integer optimization problem. Consequently, it has to be solved with meta-heuristic algorithms. Additionally, the shape of the fitness function depends on the metric in which it is formulated as well as the order of the discrete dimension in the parameter space that is linked to the leak's position.

Q.3.2: How to increase the performance of model-based leak localization?

The former mentioned dependencies of the fitness landscape can be used to manipulate the problem's shape, enabling faster convergence as well as more robust findings of the optimal solution. Furthermore, through TSA techniques, the leak size can be estimated which reduces the search space of the problem to be one-dimensional. Both performance enhancements make the problem solvable in real-time with the proposed algorithms, allowing a fast and efficient localization of the leak in the system.

Q.3.3: Is model-based leak localization applicable for finding small leaks in the real-world?

Real-world experiments—and not just simulations—have shown that this is possible for leaks down to $Q_L \approx 0.5$ L/s. Leaks with a leak outflow of around $Q_L \approx 1.0$ L/s were located using pressure sensors only without any information from the inflow measurements. This has the potential to drastically reduce the costs of the proposed method and increase its acceptability in practice. Yet, further investigations are still necessary to reveal the method's limitations.

For a more detailed description of the findings and conclusions corresponding to the leak detection methodology see Section 3.4, further results of the optimal sensor placement approach are summarized in Section 4.4. A more detailed conclusion on leak localization can be found in Section 5.4.

How much water can be saved? This question cannot be answered in general. Yet, an answer is proposed based on the artificial leak that has been detected and localized in Sections 3.3 and 5.3 with an outflow of $Q_L \approx 0.7$ L/s at position HG3880 (position I in Figure 2.32), which started on the

4.7.2016 at 12:20 (see Figure 3.19). For that reason, the leak run-time (a) resulting from applying the fast detection and localization procedure developed in this thesis is compared with (b) estimates for leak run-times taken from literature. First, (i) the leak awareness time is addressed, second, (ii) the leak localization time is examined in more detail. The estimated run-times of leaks are taken from Knobloch 2014.

(i) Leak awareness time reduction:
According to Knobloch 2014 leak run-times are for visible leaks between 3 and 8 days, and for not visible unreported leaks these run-times go up to a range between 50 and 100 days. For visible leaks, a conservative estimate of the awareness time are thus 5 days compared to 3 hours by the methods introduced in Chapter 3. Consequently, the reduced leak awareness time is 117 hours for visible leaks, resulting in nearly 300 m^3 of water being saved through a faster awareness of the leak of size $Q_L = 0.7$ L/s. For an invisible leak, this savings are conservatively calculated for the lower boundary of 50 days. Without any leak detection methods being applied, the savings are approximately 3000 m^3. Noise loggers reduce the leak awareness times according to Clark 2012 down to 28 days for a readout interval of 14 days—and 3 days for a daily readout interval. For the first case, the novel methods in this thesis lead to savings of 1800 m^3 respectively 300 m^3 in the latter case.

(ii) Leak localization time reduction:
A team of two workers equipped with noise correlators is capable of searching an area of 3 kilometers pipe length per working day in a network consisting of PVC pipes (according to Hunaidi 2012). PVC is the actual main pipe material in the measurement zone in Graz Ragnitz with a total pipe length of 10 kilometers (not counting household connections). Thus, it will take the team on average 1.65 working days to locate the leak. For the pre-located leak through the model-based leak localization, the search area is reduced to seven percent. Consequently, the leak can be localized in one hour in average, leading to savings of additional 30 m^3 of water during the localization. The savings on water are minimal, but the savings on working time and hence costs of the personal are relevant.

In total, at least 300-3000 m^3 of water could have been saved with the methods described in this thesis compared to conventional methods for this single leak. Supposing a water price of €1.3/m^3 in Austria, the monetary savings are between €400 and €4000 for this single leak. Additionally, €300 labor costs can be saved assuming that the working time of the staff and costs of €15 per hour (again, this presents a conservative estimate, since the labor costs per hour are assumed very low for a high income country like Austria). Although four additional sensors, worth approximately €200 each, are required, the leak detection and localization clearly compensates these expenses by faster detection of the first leak.

Problems evolved during the actual field test The poor performance—or call it a not so good performance since the methods proved actually to be working—has been attributed to the following factors:

- A partially-closed valve in the system was detected after the installation of the sensors at the "ideal" sensor positions. The valve influences the sensitivity of the system leading to non-ideal sensor positions. Therefore, the optimal sensor placement comparisons in Section 5.3.3 have to be taken with great caution.

- No measurements for leak free observations of the system prior to installing the artificial leaks were available and the more advanced time series models were not be able to be tested on different leak scenarios. Thus, the leak detection algorithms were only applied on S_k since it was the only scenario with a priorly longer leak-free period.
- The choice of the sensor placement that was used for the six remaining sensors was not ideal one according to the sensor placement performance tests in Chapter 5.3. Additionally, sensor failures of pressure measurement devices at important locations (HG3420 and HG4215 in Figure 2.21) were certainly not helpful.

Despite all these drawbacks, the results for the leak detection and the model-based leak localization in real-world are astonishing. Real-time detection and localization of leaks of small sizes were possible and enabled a fast and timely response to leaks in the system.

Future studies should also focus on the applicability of these methods on larger as well as multiple inlet DMAs and/or pumped systems with changing boundary conditions. Certainly, this will increase the complexity of the proposed approaches and will have an influence on the performance of leak (i) detection and (ii) localization through

(i) increasing the complexity of the generation of time series models and, thus, decreasing the forecast accuracy due to changing system conditions. Additionally, throughout bookkeeping of all possible changes in the system is necessary since these changes will have to be incorporated into the methods. This is also the case in future smart water systems with automated control approaches utilizing control valves (Cattani et al. 2017).

(ii) the change of boundary conditions (Blesa et al. 2015), influencing the sensitivity of the system and hence decreasing the optimality of ideal measurement positions. Additionally, the hydraulic model requires updates according to the changes in the system.

Recommendations for real-world application of the methods To not repeat the failures made during our case study, some recommendations are given now to avoid them in real-world applications

1. At first, compute the possible measurement positions in the field testing area with the Shortest Path 1 algorithm, since it is highly efficient even for big systems, at locations where sensors are easily deployable.
2. Prior to installing the sensors, check if all boundary valves are closed in the measurement zone. Close all valves in the zone that are supposed to be closed.
3. Install temporary pressure measurement devices at these positions for calibrating the network later on. Additionally, install flow measurement devices with high time resolutions (e.g. 1 min) at all inflow and outflow points of the zone. Collecting this data at later time points is sufficient; there is no need to deliver it in real-time.
4. Observe the system for, e.g., a few weeks to get enough data to build demand patterns and to estimate the flow and pressure conditions in the measurement zone. This data can also be used for adjusting the heights of the pressure measurements. Are there any mysterious unexplainable effects? Keep in mind that this phase is the phase with the most work for the staff, because sensors have to be installed, data has to be collected maybe manually, batteries have to be changed.
5. Open hydrants during the MNF to produce high flows in the system at multiple locations for roughness calibration.

6. Apply the automatic calibration procedure as described in Section 2.4.3.2. Are there any pipes with very high roughnesses? Maybe the diameter of the pipe is wrong in the hydraulic model or a valve is closed. Check that.
7. Once the system's state is clear and the parameters are estimated with sufficient accuracy, a long term sensor deployment can be designed.
8. Use the now calibrated hydraulic model to find sensor positions and the right sensor number for model-based leak localization with the methods proposed in Chapter 4. Think about possible power supply at measurement locations, since that will tremendously decrease the maintenance effort if no batteries have to be changed. Measurement locations at households at water meters, for example, can be considered. Also consider the data transfer at these positions. Underground locations under big layers of concrete and steel might be bad for guaranteeing a robust data communication.
9. Install the system (pressure and flow sensors with data communication) and test it by generating leaks through small hydrant openings. If this works sufficiently, wait for the first real leak to be caught by the algorithms.
10. Sensor fault detection algorithms might prove useful, since they should detect any malfunctional sensor that can subsequently be changed as fast as possible. This to guarantees good leak localization performances.

Possible future research directions to improve the methods Future work—aiming to apply the proposed methods—should focus on reducing uncertainties in the system to enable the localization of even smaller leaks. The uncertainties can have multiple causes for model-based approaches. Since these approaches work through minimizing discrepancies between real-world measurements and simulated values obtained from a hydraulic model to find the approximate location of a leak, the uncertainties can be classified into two big parts: (i) model uncertainties and (ii) measurement inaccuracies. Enhancements are expected by improving the knowledge on both parts.

(i) The success of model-based leak localization is strongly connected to the quality of the hydraulic model itself. In general, models are fraught with uncertainties (Chatfield 1995), like structural uncertainties of the model, estimation uncertainties of its parameters as well as unexplained random variations in it. The aim of model calibration is to find good estimates for model parameters, reducing its uncertainty and leading to better consistency between measured and simulated values. In general, (a) pipe roughness and (b) nodal demands are the most uncertain input variables in a simulation, because usually they are not directly measurable (D. Kang and Lansey 2011).

   (a) Roughness calibration was performed in Section 2.4.3.2, but the generation of pipe groups was only done through a zoning approach. More elaborate groupings using more realistic pipe parameters from an engineer perspective (e.g. by consideration of installations like valves, bends, household connections, pipe materials and ages) might lead to better leak localization performances.

   (b) Demand calibration was not performed in this thesis. With better demand models, better leak localization can be expected (as already investigated Moors et al. 2018). The approach through demand component calibration by Sanz Estapé 2016 is very promising . This approach utilizes time series of pressure measurements for retrieving better locally demand pattern estimates. Thus, it perfectly fits the already necessary sensor deployments prerequisites for model-based leak localization. Another approach might be putting the cart before the horse—by using stochastic end-use models (Blokker et al. 2010) and linking them with hydraulic simulation software to enable a more realistic simulation approach of WDS. Additionally, smart water meter data—measuring customer demand at high time rates at household level—can be incorporated in the models to reduce the demand uncertainty.

(ii) Obviously, measurement inaccuracies can be decreased by using better and more accurate sensors. But also, better time series models that utilize the correlations between sensors at different positions might improve the accuracy of the estimated pressures and demands without additional costs. Promising results are also expected by combining the information on the probability distributions with MCMC modeling and techniques for state estimation, e.g. non-linear Kalman or particle filters. Moreover, further investigations of the sensor placement problem (see Chapter 4), especially on incorporating uncertainties, might lead to better and more accurate measurement positions without the need (and the costs) of improving the equipment itself. A still open question is the realistic size of the ω parameter in equation (4.25). Estimates for the magnitude of this parameter can be retrieved through analyzing the information contained in the residuals of the complex time series model in Section 3.3.3.3, for example.

Additionally, experiments that extended the approach described in Section 5.2.2 to multiple leaks and applying it on the calibration scenarios in Section 2.4.3.1 for the multiple hydrant opening scenarios showed promising results. Usually, the probability of multiple leaks evolving in different parts of the system at the same time is very low. However, a method for a quick localization of multiple pipe burst can be very important in regards to a fast responses to disasters, e.g. after earthquakes. This will facilitate fast repairs to enable again a working water infrastructure right after it collapsed.

Finally, growing computer power, novel measurement devices and enhanced data communication technologies might enable in the near future the use of more sophisticated modeling approaches for whole networks as presented in Section 2.2. The improvements caused by these emerging prospective technologies on WDS management is not imaginable yet.

Final statement The methods for reducing water losses presented within this thesis showed promising results for future applications in various ways: (i) supporting WUs in the decision where and how many sensors should be used, (ii) giving early warnings of unexpected incidents within the WDS and (iii) providing the staff with the most probable locations of a failure.

It has to be noted that not the algorithm discover leaks, people at the WU are the ones who actually find and repair them. The possible advantages of the herein proposed methods can only be utilized by guaranteeing the acceptability of the methods by the people who are actually using them. A tool is only as good as the people using it and their motivation to use it—no matter how valuable, useful, sophisticated or shiny it might be. Thus, the algorithms are designed to serve WUs by providing automated services to their staff: becoming precociously aware of a leak and supporting them by giving them hints where this leak is located. If these methods prove advantageous enough, convenient to use and, last but not least, successful in the long-term, this thesis may present a small but crucial step towards a more sustainable future in which our precious water resources are protected by minimizing unnecessary losses.

Acknowledgement

"So long, and thanks for all the fish."

— Douglas Adams, *So Long, and Thanks for All the Fish*

Everybody who has written a thesis knows the long and tedious process, the battle against an army of white pages, dry itching eyes while finding mistakes in endless lines of code in the middle of the darkest nights or the frustrating times, when hydraulic network equations seem to have conspired against you, while your body is aking, because he is badly nourished by frozen pizza and gallons of coffee for months. Certainly, it is impossible to go through all of this alone—it is unimaginable without the help of other people—even if sometimes they are just distracting you. Some of these people I want to mention here.

Firstly, I would like to express my sincere gratitude to my PhD supervisor, Assoc-Prof. Daniela Fuchs-Hanusch, for the continuous support of my PhD study and related research, motivation, guidance, fun and for providing me with her immense knowledge on water distribution—which was especially of importance to me as a complete newcomer to the field of civil engineering who hadn't got the faintest idea of urban water management at all—as well as giving me the opportunity to join her working group on Sustainable Optimization of Urban Water Infrastructures as a researcher four years ago. Without her, there would not have been a thesis at all.

Besides my advisor, I want to thank the head of the institute of Urban Water Management, Prof. Dirk Muschalla, with whom I especially enjoyed chatting about optimization algorithms, and who provided me with a glimpse on the dark side of the force—wastewater.

I would also like to express my deepest appreciation to the external reviewer of this thesis—Ramon Pérez Magrané. I really appreciate the feedback and the valuable comments offered by him, which extremely improved the quality of this thesis. Indeed, I am aware of the fact that the review process certainly took a lot of his precious time and that is why I want to thank him.

I am also very grateful to my working group colleagues: My successor Michael Karli "Laser" Pointl, my hard-working master student Jakim "O'Neal" Lippacher I had the honor to co-supervise on his master thesis on roughness calibration, the highly talented "junior" programmer Georg Arbesser-Rastburg, former student assistant and programmer Lorenz "Dubster" Kaufmann, certainly one of the best DJs in the world, my colleague Stefan Krakow, a very good practitioner who risked to go to jail for the field study (just kidding, he knows what I mean), my former office roommate Shaher Zyoud, who gave me different points of view on other parts of the world. But especially, I want to thank my fellow sufferer Markus Günther. Through hilarious conferences living the "scientific rockstar life" to late night weary-eyed writing sessions at the institute, certainly, a life-long friendship evolved right from the start.

Of course, I also want to thank the people from the "dark-side-of-the-force" working group. First, Ass. Prof. Günter Gruber, who employed me at the institute as a student worker nearly a decade ago. In hindsight, this has been the early starting point of this long travel from physics to water management. Second, my colleagues and Reggaeton band members: Johnny Fuego (Johannes Leimgruber), Hermano Bromano (Roman Maier), Daddy Aldi (Thomas Hofer) and El Guerillo (Gerald Krebs). Papa Caliente enjoyed every minute making music with you guys! And last but not least the "other" David—David B. Camhy. One spring day in 2009, he brought me to the institute and I had the pleasure to sit next to him locked in a dark chamber for the next couple of years, where he patiently teached me programming and bombarded me continuously with new, abstract and genius ideas and concepts. I think I have never learned more from a single person in my entire life and nearly all of my computer, programming, conception and abstraction skills I owe to him. He truly is my mister Miyagi.

A university does not consist of scientists only—behind closed doors work people that are often overseen, but secretly contribute to science's noble goals. I received generous support of Roland Fuchs—the institute's AC/DC loving MacGyver—who contributed to the success of this thesis with his detailed knowledge on electrical circuits and through his fine craftsmanship forcing every piece of technique to work as it should, but also I am very thankful to Bernd Stojanovic, Harald Gerhold and Arthur Fischer-Colbrie. They never hesitated to support me with their help, be it in the lab or in the field. And, of course, I want to thank the institute's secretary and good soul Roswitha Pauritsch. As everybody knows, the most powerful persons in universities are secretaries. She always used her powers for good not evil—to clear big piles of bureaucracy out of my way, and doing so every time with a smile.

I want to thank all my friends in Graz, Vienna, Weiz, Gleisdorf or abroad. They were always ready to distract me with endless late night discussions in woozy bars, when I needed it. Also they have been a strong moral support during rough times.

My deepest heartfelt appreciation goes to my beloved girlfriend Brigitte. I met her nearly at the beginning of this thesis—and as this project grew, at the same time my love to her did—everyday a little more, even if you think it can't possibly grow any larger. I really admire her—not just as my girlfriend, but also for her sharp mind and intelligence. And I also want to thank her family making me feel that I am warm welcomed right away from the first day.

Finally, I want to thank my family—my father, my mother and my older but smaller brother—for their support in all that years and letting me know, that they are proud of me. And I am especially thankful for my beloved grandmother, who unfortunately passed away few hours after I finished the first complete draft of this thesis. She taught me to work hard and always give your best, support others and enjoy the little things in life that make it worth living. And to always face life's difficult situations with a smile, humor and a cigarette. This thesis is dedicated to her—I miss you!

David B. Steffelbauer
Vienna 2018

Acronyms

ACF AutoCorrelation Function 51

AIC Akaike's Information Criterion 49

ANN Artifical Neural Networks 79

AR AutoRegressive 53

ARMA AutoRegressive-Moving Average 53

ASCE American Society of Civil Engineers 8

BFGS Broyden-Fletcher-Goldfarb-Shanno 32

BIC Bayes Information Criterion 49

CUSUM CUmulative SUM control chart 57

DE Differential Evolution 13

DFS Depth-First Search 118

DMA District-Metered Area 62

EPS Extended Period Simulation 43

GA Genetic Algorithm 13

GDA Gradient Descent Algorithm 23

GoF Goodness-of-Fit 13

HMC Hamiltonian-Monte-Carlo 48

IWA International Water Association 3

MA Moving Average 53

MAE Mean Absolute Error 56

MAPE Mean Absolute Percentage Error 56

MCMC Markov-Chain-Monte-Carlo 13

MCS Monte Carlo Simulations 13

MNF Minimum Night Flow 5

MSE Mean Squared Error 56

NCG Nonlinear Conjugate Gradient 22

NFL No-Free-Lunch 21

NRA Newton-Raphson Algorithm 22

NRW Non-revenue water 3

NUTS No-U-Turn Sampler 48

ODE Ordinary Differential Equation 44

OpenSDM Open-source Scientific Data Management System 65

OSP Optimal Sensor Placement 117

PACF Partial AutoCorrelation Function 51

RMSE Root Mean Squared Error 56

SCADA Supervisory Control and Data Acquisition 9

SPC	Single-Point-Crossover	26
SPUDU	Sensor Placement Under Demand Uncertainties	
SS	Forecast Skill Score	56
SVM	Support Vector Machines	79
TSA	Time Series Analysis	50
TSDB	Time Series Database	65
UFM	Ultrasonic Flow Meter	64
UIM	Uniform-Integer Mutation	27
WDN	Water Distribution Network	4
WDS	Water Distribution System	2
WU	Water Utility	1

Bibliography

“Ahenny (adj.) - The way people stand when examining other people’s bookshelves."

— Douglas Adams, *The Deeper Meaning of Liff*

Abraham, E. and I. Stoianov (2016). «Sparse Null Space Algorithms for Hydraulic Analysis of Large-Scale Water Supply Networks». In: *Journal of Hydraulic Engineering* 142.3, p. 04015058. DOI: 10.1061/(ASCE)HY.1943-7900.0001089 (cit. on pp. 40–43).

Aksela, K., M. Aksela, and R. Vahala (2009). «Leakage detection in a real distribution network using a SOM». In: *Urban Water Journal* 6.4, pp. 279–289. DOI: 10.1080/15730620802673079 (cit. on p. 81).

Bäck, T., D. B. Fogel, and Z. Michalewicz, eds. (2000). *Evolutionary computation*. OCLC: ocm44807816. Bristol ; Philadelphia: Institute of Physics Publishing. 2 pp. (cit. on pp. 21, 25 sqq.).

Bakker, M., J. H. G. Vreeburg, M. Van De Roer, and L. C. Rietveld (2014). «Heuristic burst detection method using flow and pressure measurements». In: *Journal of Hydroinformatics* 16.5, pp. 1194–1209. DOI: 10.2166/hydro.2014.120 (cit. on pp. 79 sq.).

Bakker, M., J. H. G. Vreeburg, K. M. van Schagen, and L. C. Rietveld (2013). «A fully adaptive forecasting model for short-term drinking water demand». In: *Environmental Modelling & Software* 48, pp. 141–151. DOI: 10.1016/j.envsoft.2013.06.012 (cit. on p. 80).

Basseville, M. and I. V. Nikiforov (1993). *Detection of abrupt changes: theory and application*. Vol. 104. Prentice Hall Englewood Cliffs (cit. on pp. 56 sqq.).

BBC (2018). «Cape Town ’averts’ water shut-off». In: *BBC News* (cit. on p. 2).

Blesa, J., F. Nejjari, and R. Sarrate (2014). «Robustness Analysis of Sensor Placement for Leak Detection and Location under Uncertain Operating Conditions». In: *Procedia Engineering*. 16th Water Distribution System Analysis Conference, WDSA2014Urban Water Hydroinformatics and Strategic Planning 89, pp. 1553–1560. DOI: 10.1016/j.proeng.2014.11.453 (cit. on pp. 119, 122, 130, 176).

– (2015). «Robust sensor placement for leak location: analysis and design». In: *Journal of Hydroinformatics*, jh2015021. DOI: 10.2166/hydro.2015.021 (cit. on pp. 119, 192).

Blokker, E. J. M., J. H. G. Vreeburg, and J. C. van Dijk (2010). «Simulating Residential Water Demand with a Stochastic End-Use Model». In: *Journal of Water Resources Planning and Management* 136.1, pp. 19–26. DOI: 10.1061/(ASCE)WR.1943-5452.0000002 (cit. on p. 194).

Bohatch, T. (2017). *What's causing Cape Town's water crisis?* GroundUp News. URL: https://www.groundup.org.za/article/whats-causing-cape-towns-water-crisis/ (visited on 04/24/2018) (cit. on p. 2).

Box, G. E. P., G. M. Jenkins, G. C. Reinsel, and G. M. Ljung (2015). *Time Series Analysis: Forecasting and Control*. Google-Books-ID: rNt5CgAAQBAJ. John Wiley & Sons. 709 pp. (cit. on p. 94).

Brandes, U. and T. Erlebach, eds. (2005). *Network Analysis: Methodological Foundations*. Theoretical Computer Science and General Issues. Berlin Heidelberg: Springer-Verlag (cit. on p. 46).

Camhy D., Gruber G., Steffelbauer D. B., Hofer T. F., and Muschalla D. (2014). «OpenSDM - An Open Sensor Data Management Tool». In: *11th International Conference on Hydroinformatics*. HIC (cit. on pp. 16, 65).

Camhy, D., V. Gamerith, D. B. Steffelbauer, D. Muschalla, and G. Gruber (2012). «Scientific Data Management with Open Source Tools – An Urban Drainage Example». In: *9th International Conference on Urban Drainage Modelling (UDM)*. Ed. by J. Plavšić. Belgrade, Serbia: Faculty of Civil Engineering, University of Belgrade (cit. on pp. 16, 65).

Camp, T. R. and H. L. Hazen (1934). «Hydraulic analysis of water distribution systems by means of an electric network analyzer». In: *Journal of New England Water Works Association (JNEWWA)* 48, pp. 383–404 (cit. on p. 39).

Casillas, M. V., L. E. Garza Castañón, and V. Puig (2014). «Model-based leak detection and location in water distribution networks considering an extended-horizon analysis of pressure sensitivities». In: *Journal of Hydroinformatics* 16.3, p. 649. DOI: 10.2166/hydro.2013.019 (cit. on pp. 148, 151, 162 sq.).

Casillas, M. V., L. E. Garza-Castanon, and V. Puig (2013). «Extended-horizon analysis of pressure sensitivities for leak detection in water distribution networks: Application to the Barcelona network». In: *Control Conference (ECC), 2013 European*. Control Conference (ECC), 2013 European, pp. 401–409 (cit. on pp. 148, 151, 160 sq., 163).

Casillas, M. V., L. E. Garza-Castañón, and V. Puig (2015). «Optimal Sensor Placement for Leak Location in Water Distribution Networks using Evolutionary Algorithms». In: *Water* 7.11, pp. 6496–6515. DOI: 10.3390/w7116496 (cit. on pp. 118 sq.).

Casillas, M. V., V. Puig, L. E. Garza-Castañón, and A. Rosich (2013). «Optimal Sensor Placement for Leak Location in Water Distribution Networks Using Genetic Algorithms». In: *Sensors* 13.11, pp. 14984–15005. DOI: 10.3390/s131114984 (cit. on pp. 117 sqq., 126 sqq., 143, 174).

Cattani, M., C. A. Boano, D. B. Steffelbauer, S. Kaltenbacher, M. Günther, K. Römer, D. Fuchs-Hanusch, and M. Horn (2017). «Adige: An Efficient Smart Water Network Based on Long-range Wireless Technology». In: *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*. CySWATER '17. New York, NY, USA: ACM, pp. 3–6. DOI: 10.1145/3055366.3055367 (cit. on pp. 15, 192).

Chatfield, C. (1995). «Model Uncertainty, Data Mining and Statistical Inference». In: *Journal of the Royal Statistical Society. Series A (Statistics in Society)* 158.3, pp. 419–466 (cit. on p. 193).

Cheng, W. and Z. He (2011). «Calibration of Nodal Demand in Water Distribution Systems». In: *Journal of Water Resources Planning and Management* 137.1, pp. 31–40. DOI: 10.1061/(ASCE)WR.1943-5452.0000093 (cit. on p. 122).

Christodoulou, S., A. Gagatsis, S. Xanthos, S. Kranioti, A. Agathokleous, and M. Fragiadakis (2013). «Entropy-Based Sensor Placement Optimization for Waterloss Detection in Water Distribution Networks». In: *Water Resources Management* 27.13, pp. 4443–4468. DOI: `10.1007/s11269-013-0419-8` (cit. on pp. 118 sq.).

Clark, A. (2012). «Increasing efficiency with permanent leakage monitoring». In: *Water Loss Reduction-Proceedings of the 7th IWA Specialist Conference*, pp. 26–29 (cit. on p. 191).

CNN, Lauren Said-Moorhouse {and} Gianluca Mezzofiore (2018). *Cape Town cuts water use limit by nearly half*. CNN. URL: `https://www.cnn.com/2018/02/01/africa/cape-town-water-crisis-intl/index.html` (visited on 04/24/2018) (cit. on p. 2).

Colombo, A. F. and B. Karney (2002). «Energy and costs of leaky pipes: Toward comprehensive picture». In: *Journal of Water Resources Planning and Management* 128.6, pp. 441–450 (cit. on p. 3).

Colombo, A. F., P. Lee, and B. W. Karney (2009). «A selective literature review of transient-based leak detection methods». In: *Journal of Hydro-environment Research* 2.4, pp. 212–227. DOI: `10.1016/j.jher.2009.02.003` (cit. on pp. 2, 9 sq., 45, 147, 152).

Cross, H. (1936). «Analysis of flow in networks of conduits or conductors». In: *University of Illinois. Engineering Experiment Station. Bulletin; no. 286* 34.22, p. 34 (cit. on p. 39).

Crouch, D. P. (1993). *Water Management in Ancient Greek Cities*. Oxford, New York: Oxford University Press. 400 pp. (cit. on p. 38).

Cugueró-Escofet, M., V. Puig, and J. Quevedo (2017). «Optimal pressure sensor placement and assessment for leak location using a relaxed isolation index: Application to the Barcelona water network». In: *Control Engineering Practice* 63, pp. 1–12. DOI: `10.1016/j.conengprac.2017.03.003` (cit. on pp. 118 sq.).

Cuthill, E. and J. McKee (1969). «Reducing the bandwidth of sparse symmetric matrices». In: *Proceedings of the 1969 24th national conference*. ACM, pp. 157–172 (cit. on p. 158).

Diestel, R. (2017). *Graph Theory*. 5th. Springer Publishing Company, Incorporated (cit. on p. 159).

Dijkstra, E. W. (1959). «A note on two problems in connexion with graphs». In: *Numerische Mathematik* 1.1, pp. 269–271. DOI: `10.1007/BF01386390` (cit. on pp. 46, 123, 162).

Donkor, E. A., T. A. Mazzuchi, R. Soyer, and A. J. Roberson (2014). «Urban Water Demand Forecasting: Review of Methods and Models». In: *Journal of Water Resources Planning and Management* 140.2, pp. 146–159. DOI: `10.1061/(ASCE)WR.1943-5452.0000314` (cit. on p. 89).

Droste, S., T. Jansen, and I. Wegener (1999). «Perhaps not a free lunch but at least a free appetizer». In: *GECCO*, pp. 833–839 (cit. on p. 155).

Duarte, M. (2015). *Notes on Scientific Computing for Biomechanics and Motor Control*. GitHub (cit. on p. 102).

Elhay, S. and A. R. Simpson (2011). «Dealing with Zero Flows in Solving the Nonlinear Equations for Water Distribution Systems». In: *Journal of Hydraulic Engineering* 137.10, pp. 1216–1224. DOI: `10.1061/(ASCE)HY.1943-7900.0000411` (cit. on pp. 41, 46).

Eliades, D. G. and M. M. Polycarpou (2012). «Leakage fault detection in district metered areas of water distribution systems». In: *Journal of Hydroinformatics* 14.4, pp. 992–1005. DOI: `10.2166/hydro.2012.109` (cit. on pp. 80, 114).

Epp, R. and A. G. Fowler (1970). «Efficient Code for Steady-State Flows in Networks». In: *Journal of the Hydraulics Division* 96.1, pp. 43–56 (cit. on p. 39).

EurEau (2017). *Europe's water in figures: An overview of the European drinking water and waste water sectors*. 2017 edition. Brussels, Belgium: EurEau The European Federation of National Associations of Water Services, p. 22 (cit. on p. 3).

Farley, B., S. R. Mounce, and J. B. Boxall (2008). «Optimal Locations of Pressure Meters for Burst Detection». In: *Water Distribution Systems Analysis 2008*. American Society of Civil Engineers, pp. 1–11 (cit. on pp. 118 sq.).

– (2010). «Field testing of an optimal sensor placement methodology for event detection in an urban water distribution network». In: *Urban Water Journal* 7.6, pp. 345–356. DOI: 10.1080/1573062X.2010.526230 (cit. on pp. 11, 118 sq.).

– (2013). «Development and Field Validation of a Burst Localization Methodology». In: *Journal of Water Resources Planning and Management* 139.6, pp. 604–613. DOI: 10.1061/(ASCE)WR.1943-5452.0000290 (cit. on pp. 118 sq.).

Farley, M. and S. Trow (2003). *Losses in water distribution networks: a practitioner's guide to assessment, monitoring and control*. London: IWA Publ. 282 pp. (cit. on p. 8).

Ferrante, M., S. Meniconi, and B. Brunone (2014). «Local and Global Leak Laws: The Relationship Between Pressure and Leakage for a Single Leak and for a District with Leaks». In: *Water Resources Management* 28.11, pp. 3761–3782. DOI: 10.1007/s11269-014-0708-x (cit. on p. 153).

Fletcher, R. (1987). *Practical methods of optimization*. Google-Books-ID: 3EzvAAAAMAAJ. Wiley. 464 pp. (cit. on p. 32).

Fortin, Félix-Antoine, De Rainville, Marc-André Gardner Gardner, Marc Parizeau, Christian Gagné, et al. (2012). «DEAP: Evolutionary algorithms made easy». In: *The Journal of Machine Learning Research* 13.1, pp. 2171–2175 (cit. on p. 37).

Fox, J. A. (1977). *Hydraulic analysis of unsteady flow in pipe networks*. Google-Books-ID: jeZSAAAAMAAJ. Halsted Press. 238 pp. (cit. on pp. 44 sq.).

Freeman, J. R. (1892). «The arrangement of hydrants and water-pipes for the protection of a city against fire». In: *Journal of New England Water Works Association (JNEWWA)* 7.1, pp. 49–81 (cit. on p. 39).

Fuchs-Hanusch, D. and D. B. Steffelbauer (2017). «Real-world Comparison of Sensor Placement Algorithms for Leakage Localization». In: *Procedia Engineering*. XVIII International Conference on Water Distribution Systems, WDSA2016 186, pp. 499–505. DOI: 10.1016/j.proeng.2017.03.262 (cit. on pp. 15, 117, 129, 176).

Fuchs-Hanusch, D., D. B. Steffelbauer, and M. Günther (2015). «Drucksensoren zur Leckagelokalisierung - Optimale Platzierung der Sensoren durch Berücksichtigung von Verbraucherunsicherheiten». In: *Aqua & Gas* 11, pp. 26–31 (cit. on p. 14).

Fuchs-Hanusch, D., D. B. Steffelbauer, M. Günther, and D. Muschalla (2015). «Systematic material and crack type specific pipe burst outflow simulations by means of EPANET2». In: *Urban Water Journal*, pp. 1–11. DOI: 10.1080/1573062X.2014.994006 (cit. on pp. 5, 14).

Giustolisi, O. and D. Laucelli (2011). «Water Distribution Network Pressure-Driven Analysis Using the Enhanced Global Gradient Algorithm (EGGA)». In: *Journal of Water Resources Planning and Management* 137.6, pp. 498–510. DOI: 10.1061/(ASCE)WR.1943-5452.0000140 (cit. on p. 42).

Giustolisi, O., D. Savic, and Z. Kapelan (2008). «Pressure-Driven Demand and Leakage Simulation for Water Distribution Networks». In: *Journal of Hydraulic Engineering* 134.5, pp. 626–635. DOI: 10.1061/(ASCE)0733-9429(2008)134:5(626) (cit. on p. 42).

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. (cit. on p. 25).

González-Gómez, F., R. Martínez-Espiñeira, M. A. García-Valiñas, and M. A. García-Rubio (2012). «Explanatory factors of urban water leakage rates in Southern Spain». In: *Utilities Policy* 22 (Supplement C), pp. 22–30. DOI: 10.1016/j.jup.2012.02.002 (cit. on p. 3).

Goulet, J. A., S. Coutu, and I. F. C. Smith (2013). «Model falsification diagnosis and sensor placement for leak detection in pressurized pipe networks». In: *Advanced Engineering Informatics* 27.2, pp. 261–269. DOI: 10.1016/j.aei.2013.01.001 (cit. on pp. 149, 151).

Granjon, P. (2013). «The CuSum algorithm-a small review». In: (cit. on pp. 57 sq.).

Grant, S. B., T. D. Fletcher, D. Feldman, J. Saphores, P. L. M. Cook, M. Stewardson, K. Low, K. Burry, and A. J. Hamilton (2013). «Adapting Urban Water Systems to a Changing Climate: Lessons from the Millennium Drought in Southeast Australia». In: *Environmental Science & Technology* 47.19, pp. 10727–10734. DOI: 10.1021/es400618z (cit. on pp. 1 sq.).

Gruber, G., V. Gamerith, J. Olsson, D. Camhy, D. B. Steffelbauer, M. Hochedlinger, P. Dihé, S. Schlobinski, and L. Gidhagen (2012). «SUDPLAN: Developing a Decision Support System to Cope with Climate Change – Urban Drainage Pilot Linz». In: *IWA World Water Congress*. Busan, Corea (cit. on p. 16).

Günther, M., D. Camhy, D. B. Steffelbauer, M. Neumayer, and D. Fuchs-Hanusch (2015). «Showcasing a Smart Water Network Based on an Experimental Water Distribution System». In: *Computing and Control for the Water Industry (CCWI2015) Sharing the best practice in water management* 119, pp. 450–457. DOI: 10.1016/j.proeng.2015.08.857 (cit. on p. 15).

Günther, M., D. B. Steffelbauer, and D. Fuchs-Hanusch (2016). «Fault detection data creation using an experimental water distribution system». In: *Proceedings of the 2016 3rd Conference on Control and Fault-Tolerant Systems*. Systol. Barcelona, Spain: IEEE, pp. 331–336 (cit. on p. 15).

Günther, M., D. B. Steffelbauer, M. Neumayer, and D. Fuchs-Hanusch (2014). «Experimental Setup to Examine Leakage Outflow in a Scaled Water Distribution Network». In: *Procedia Engineering*. 16th Water Distribution System Analysis Conference, WDSA2014 Urban Water Hydroinformatics and Strategic Planning 89, pp. 311–317. DOI: 10.1016/j.proeng.2014.11.193 (cit. on p. 15).

Hanak, E., J. Mount, C. Chappelle, J. Lund, J. Medellín-Azuara, P. Moyle, and N. Seavy (2015). «What if California's drought continues». In: *Public Policy Institute of California* (cit. on p. 2).

Hart, W. E. and R. Murray (2010). «Review of Sensor Placement Strategies for Contamination Warning Systems in Drinking Water Distribution Systems». In: *Journal of Water Resources Planning and Management* 136.6, pp. 611–619. DOI: 10.1061/(ASCE)WR.1943-5452.0000081 (cit. on p. 118).

Herrera, M., E. Abraham, and I. Stoianov (2016). «A Graph-Theoretic Framework for Assessing the Resilience of Sectorised Water Distribution Networks». In: *Water Resources Management* 30.5, pp. 1685–1699. DOI: 10.1007/s11269-016-1245-6 (cit. on p. 46).

Hoag, L. N. and G. Weinberg (1957). «Pipeline Network Analysis by Electronic Digital Computer». In: *Journal (American Water Works Association)* 49.5, pp. 517–524 (cit. on p. 39).

Hoffman, M. D. and A. Gelman (2014). «The No-U-turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo». In: *J. Mach. Learn. Res.* 15.1, pp. 1593–1623 (cit. on pp. 48 sq.).

Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Vol. viii. Oxford, England: U Michigan Press. 183 pp. (cit. on p. 25).

Hunaidi, O. (2012). «Acoustic leak detection survey strategies for water distribution pipes». In: *Construction Technology Update* 79, pp. 1–5 (cit. on p. 191).

Hutton, C. and Z. Kapelan (2015). «Real-time Burst Detection in Water Distribution Systems Using a Bayesian Demand Forecasting Methodology». In: *Procedia Engineering*. Computing and Control for the Water Industry (CCWI2015) Sharing the best practice in water management 119, pp. 13–18. DOI: 10.1016/j.proeng.2015.08.847 (cit. on pp. 81, 114).

Hutton, C., Z. Kapelan, L. Vamvakeridou-Lyroudia, and D. Savić (2014). «Dealing with Uncertainty in Water Distribution System Models: A Framework for Real-Time Modeling and Data Assimilation». In: *Journal of Water Resources Planning and Management* 140.2, pp. 169–183. DOI: 10.1061/(ASCE)WR.1943-5452.0000325 (cit. on p. 119).

Hyndman, Rob J and George Athanasopoulos (2014). *Forecasting: principles and practice*. OTexts (cit. on p. 56).

IWA, The International Water Association (2017). *Desalination – the Australian Experience*. International Water Association. URL: http://www.iwa-network.org/news/desalination-australian-experience/ (visited on 04/24/2018) (cit. on p. 2).

Izquierdo, J., P. A. López, F. J. Martínez, and R. Pérez (2007). «Fault detection in water supply systems using hybrid (theory and data-driven) modelling». In: *Mathematical and Computer Modelling* 46.3, pp. 341–350. DOI: 10.1016/j.mcm.2006.11.013 (cit. on pp. 150 sq.).

Jones, E., T. Oliphant, P. Peterson, et al. (2001). *SciPy: Open source scientific tools for Python* (cit. on p. 33).

Jung, B. S. and B. Karney (2016). «A practical overview of unsteady pipe flow modeling: from physics to numerical solutions». In: *Urban Water Journal*, pp. 1–7. DOI: 10.1080/1573062X.2016.1223323 (cit. on pp. 40, 43 sq.).

Jung, D., D. Kang, J. Liu, and K. Lansey (2015). «Improving the rapidity of responses to pipe burst in water distribution systems: a comparison of statistical process control methods». In: *Journal of Hydroinformatics* 17.2, p. 307. DOI: 10.2166/hydro.2014.101 (cit. on p. 80).

Jung, D. and J. H. Kim (2018). «State Estimation Network Design for Water Distribution Systems». In: *Journal of Water Resources Planning and Management* 144.1, p. 06017006. DOI: 10.1061/(ASCE)WR.1943-5452.0000862 (cit. on p. 115).

Jung, D. and K. Lansey (2015). «Water Distribution System Burst Detection Using a Nonlinear Kalman Filter». In: *Journal of Water Resources Planning and Management* 141.5, p. 04014070. DOI: 10.1061/(ASCE)WR.1943-5452.0000464 (cit. on p. 80).

Kalman, R. E. (1960). «A new approach to linear filtering and prediction problems». In: *Journal of basic Engineering* 82.1, pp. 35–45 (cit. on p. 55).

Kaltenbacher, S., D. B. Steffelbauer, M. Cattani, D. Fuchs-Hanusch, K. Römer, and M. Horn (2017). «A Dynamic Model for Smart Water Distribution Networks». In: pp. 1–8 (cit. on pp. 15, 44).

Kang, D. S., M. F. K. Pasha, and K. Lansey (2009). «Approximate methods for uncertainty analysis of water distribution systems». In: *Urban Water Journal* 6.3, pp. 233–249. DOI: 10.1080/15730620802566844 (cit. on p. 47).

Kang, D. and K. Lansey (2010). «Optimal Meter Placement for Water Distribution System State Estimation». In: *Journal of Water Resources Planning and Management* 136.3, pp. 337–347. DOI: 10.1061/(ASCE)WR.1943-5452.0000037 (cit. on p. 117).

– (2011). «Demand and Roughness Estimation in Water Distribution Systems». In: *Journal of Water Resources Planning and Management* 137.1, pp. 20–30. DOI: 10.1061/(ASCE)WR.1943-5452.0000086 (cit. on p. 193).

Kapelan, Z. (2002). «Calibration of water distribution system hydraulic models». PhD thesis. University of Exeter (cit. on pp. 39 sq., 43 sq.).

Kapelan, Z., D. Savic, and G. Walters (2003). «Multiobjective Sampling Design for Water Distribution Model Calibration». In: *Journal of Water Resources Planning and Management* 129.6, pp. 466–479. DOI: 10.1061/(ASCE)0733-9496(2003)129:6(466) (cit. on pp. 122, 144).

– (2005). «Optimal Sampling Design Methodologies for Water Distribution Model Calibration». In: *Journal of Hydraulic Engineering* 131.3, pp. 190–200. DOI: 10.1061/(ASCE)0733-9429(2005)131:3(190) (cit. on pp. 121, 135).

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983). «Optimization by Simulated Annealing». In: *Science* 220.4598, pp. 671–680. DOI: 10.1126/science.220.4598.671 (cit. on p. 25).

Knobloch, A. (2014). «Automatisierte Wassermengenanalyse in der Trinkwasserversorgung». PhD thesis. Karlsruhe, Germany: Bauingenieur-, Geo- und Umweltwissenschaften des Karlsruher Instituts für Technologie (KIT) (cit. on p. 191).

Lambert, A. and W. Hirner (2000). *Losses from water supply systems. Standard terminology and recommended performance measures [online]. IWA Blue Pages, International Water Association* (cit. on pp. 3 sq.).

Leeuwen, C. J. van (2017). «Water governance and the quality of water services in the city of Melbourne». In: *Urban Water Journal* 14.3, pp. 247–254. DOI: 10.1080/1573062X.2015.1086008 (cit. on p. 3).

Leimgruber, J., D. B. Steffelbauer, G. Krebs, and D. Muschalla (2017). «Back to the roots: A storm event-based assessment of LID performance to restore the natural water balance». In: ICUD 2017. Prague (cit. on p. 16).

Lippacher, J. (2018). *Methoden der Kalibrierung von Trinkwasserverteilnetzen und deren Einfluss auf die modellbasierte Leckage Lokalisierung*. Graz, Austria: Master Thesis, Graz University of Technology (cit. on p. 70).

Low, K. G., S. B. Grant, A. J. Hamilton, K. Gan, J. Saphores, M. Arora, and D. L. Feldman (2015). «Fighting drought with innovation: Melbourne's response to the Millennium Drought in Southeast Australia». In: *Wiley Interdisciplinary Reviews: Water* 2.4, pp. 315–328. DOI: 10.1002/wat2.1087 (cit. on p. 1).

Maier, H. R., Z. Kapelan, J. Kasprzyk, J. Kollat, L. S. Matott, M. C. Cunha, G. C. Dandy, M. S. Gibbs, E. Keedwell, A. Marchi, A. Ostfeld, D. Savic, D. P. Solomatine, J. A. Vrugt, A. C. Zecchin, B. S. Minsker, E. J. Barbour, G. Kuczera, F. Pasha, A. Castelletti, M. Giuliani, and P. M. Reed (2014). «Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions». In: *Environmental Modelling & Software* 62, pp. 271–299. DOI: `10.1016/j.envsoft.2014.09.013` (cit. on pp. 154 sq.).

Maier, R., T. Hofer, D. B. Steffelbauer, G. Gruber, and D. Muschalla (2015). «Integrated real-time control of a central storage tunnel for minimization of sewer emissions in Graz, Austria». In: EJSW 2015 (cit. on p. 16).

Mala-Jetmarova, H., A. Barton, and A. Bagirov (2015). «A history of water distribution systems and their optimisation». In: *Water Science & Technology: Water Supply* 15.2, p. 224. DOI: `10.2166/ws.2014.115` (cit. on pp. 38 sq.).

Mallick, K. N., I. Ahmed, K. S. Tickle, and K. E. Lansey (2002). «Determining Pipe Groupings for Water Distribution Networks». In: *Journal of Water Resources Planning and Management* 128.2, pp. 130–139. DOI: `10.1061/(ASCE)0733-9496(2002)128:2(130)` (cit. on p. 70).

Martin, D. W. and G. Peters (1970). «The application of Newton's method to network analysis by digital computer». In: *Journal of the institute of Water Engineers* 17.2, pp. 115–129 (cit. on p. 39).

Mashford, J., D. De Silva, S. Burn, and D. Marney (2012). «Leak Detection in Simulated Water Pipe Networks Using Svm». In: *Applied Artificial Intelligence* 26.5, pp. 429–444. DOI: `10.1080/08839514.2012.670974` (cit. on pp. 150 sq.).

McIlroy, M. S. (1950). «Direct-Reading Electric Analyzer for Pipeline Networks». In: *Journal (American Water Works Association)* 42.4, pp. 347–366 (cit. on p. 39).

Moe, C. L. and R. D. Rheingans (2006). «Global challenges in water, sanitation and health». In: *Journal of Water and Health* 4 (S1), pp. 41–57 (cit. on p. 1).

Moors, J., L. Scholten, J. P. van der Hoek, and J. den Besten (2018). «Automated leak localization performance without detailed demand distribution data». In: *Urban Water Journal*, pp. 1–8. DOI: `10.1080/1573062X.2017.1414272` (cit. on pp. 162, 193).

Morris, M. (2017). «City of Cape Town's water 'bungle'». In: *Weekend Argus* (cit. on p. 2).

Mounce, S. R., J. B. Boxall, and J. Machell (2010). «Development and Verification of an Online Artificial Intelligence System for Detection of Bursts and Other Abnormal Flows». In: *Journal of Water Resources Planning and Management* 136.3, pp. 309–318. DOI: `10.1061/(ASCE)WR.1943-5452.0000030` (cit. on p. 81).

Mounce, S. R. and J. Machell (2006). «Burst detection using hydraulic data from water distribution systems with artificial neural networks». In: *Urban Water Journal* 3.1, pp. 21–31. DOI: `10.1080/15730620600578538` (cit. on p. 79).

Mounce, S. R., R. B. Mounce, and J. B. Boxall (2011). «Novelty detection for time series data analysis in water distribution systems using support vector machines». In: *Journal of Hydroinformatics* 13.4, pp. 672–686. DOI: `10.2166/hydro.2010.144` (cit. on pp. 79 sq.).

Mulligan, Gabriella (2018). «Will Cape Town be the first city to run out of water?» In: *BBC News* (cit. on p. 2).

Naumann, U. (2012). *The Art of Differentiating Computer Programs: An Introduction to Algorithmic Differentiation*. Google-Books-ID: OgQuUR4nLu0C. SIAM. 349 pp. (cit. on pp. 22, 122).

Nejjari, F., R. Sarrate, and J. Blesa (2015). «Optimal Pressure Sensor Placement in Water Distribution Networks Minimizing Leak Location Uncertainty». In: *Procedia Engineering*. Computing and Control for the Water Industry (CCWI2015) Sharing the best practice in water management 119, pp. 953–962. DOI: 10.1016/j.proeng.2015.08.979 (cit. on pp. 118 sq.).

Neumayer, M., D. B. Steffelbauer, M. Günther, and D. Fuchs-Hanusch (2014). «Computational efficient small signal model for fast hydraulic simulations». In: 11th International Conference on Hydroinformatics. New York City, USA (cit. on p. 16).

Nicklow, J., P. Reed, D. Savic, T. Dessalegne, L. Harrell, A. Chan-Hilton, M. Karamouz, B. Minsker, A. Ostfeld, A. Singh, and E. Zechman (2009). «State of the Art for Genetic Algorithms and Beyond in Water Resources Planning and Management». In: *Journal of Water Resources Planning and Management* 136.4, pp. 412–432. DOI: 10.1061/(ASCE)WR.1943-5452.0000053 (cit. on p. 25).

Niemczynowicz, J. (1999). «Urban hydrology and water management – present and future challenges». In: *Urban Water* 1.1, pp. 1–14. DOI: 10.1016/S1462-0758(99)00009-6 (cit. on p. 1).

Nocedal, J. and S. Wright (2006). *Numerical Optimization*. Google-Books-ID: 7wDpBwAAQBAJ. Springer Science & Business Media. 651 pp. (cit. on pp. 19 sqq., 23, 32).

Oki, T. and S. Kanae (2006). «Global Hydrological Cycles and World Water Resources». In: *Science* 313.5790, pp. 1068–1072. DOI: 10.1126/science.1128845 (cit. on p. 1).

Ormsbee, L. (2008). «The History of Water Distribution Network Analysis: The Computer Age». In: *Water Distribution Systems Analysis Symposium 2006*. American Society of Civil Engineers, pp. 1–6 (cit. on pp. 38 sq.).

Palau, C.V., F.J. Arregui, and M. Carlos (2011). «Burst detection in water networks using principal component analysis». In: *Journal of Water Resources Planning and Management* 138.1, pp. 47–54. DOI: 10.1061/(ASCE)WR.1943-5452.0000147 (cit. on p. 81).

Pathirana, A. (2011). «EPANET2 Desktop Application for Pressure Driven Demand Modeling». In: American Society of Civil Engineers, pp. 65–74. DOI: 10.1061/41203(425)8 (cit. on p. 42).

Pérez, R., M. A. Cugueró, J. Cugueró, and G. Sanz (2014). «Accuracy Assessment of Leak Localisation Method Depending on Available Measurements». In: *Procedia Engineering*. 12th International Conference on Computing and Control for the Water Industry, CCWI2013 70, pp. 1304–1313. DOI: 10.1016/j.proeng.2014.02.144 (cit. on pp. 118 sq.).

Pérez, R., V. Puig, J. Pascual, A. Peralta, E. Landeros, and L. Jordanas (2009). «Pressure sensor distribution for leak detection in Barcelona water distribution network». In: *Water Science & Technology: Water Supply* 9.6, p. 715. DOI: 10.2166/ws.2009.372 (cit. on pp. 118 sq., 128, 143, 174).

Pérez, R., V. Puig, J. Pascual, J. Quevedo, E. Landeros, and A. Peralta (2011). «Methodology for leakage isolation using pressure sensitivity analysis in water distribution networks». In: *Control Engineering Practice* 19.10, pp. 1157–1167. DOI: 10.1016/j.conengprac.2011.06.004 (cit. on pp. 121 sq., 125, 148, 151, 153).

Pérez, R., J. Quevedo, V. Puig, F. Nejjari, M. A. Cuguero, G. Sanz, and J. M. Mirats (2011). «Leakage isolation in water distribution networks: A comparative study of two methodologies on a real case study». In: *2011 19th Mediterranean Conference on Control Automation (MED)*. 2011 19th Mediterranean Conference on Control Automation (MED), pp. 138–143. DOI: 10.1109/MED.2011.5982979 (cit. on pp. 122, 148, 151).

Pichler, A., D. B. Steffelbauer, and A. Nazarov (2014). «Examples for Genetic Algorithm based optimal RFID tag antenna design». In: *2014 IEEE RFID Technology and Applications Conference (RFID-TA)*. 2014 IEEE RFID Technology and Applications Conference (RFID-TA), pp. 223–227. DOI: 10.1109/RFID-TA.2014.6934232 (cit. on p. 16).

Pitzer, E. and M. Affenzeller (2012). «A Comprehensive Survey on Fitness Landscape Analysis». In: *Recent Advances in Intelligent Engineering Systems*. Studies in Computational Intelligence. Springer, Berlin, Heidelberg, pp. 161–191. DOI: 10.1007/978-3-642-23229-9_8 (cit. on p. 155).

Pitzer, E., M. Affenzeller, A. Beham, and S. Wagner (2011). «Comprehensive and automatic fitness landscape analysis using heuristiclab». In: *International Conference on Computer Aided Systems Theory*. Springer, pp. 424–431 (cit. on p. 155).

Poli, R., J. Kennedy, and T. Blackwell (2007). «Particle swarm optimization: An overview». In: *Swarm Intelligence* 1.1, pp. 33–57. DOI: 10.1007/s11721-007-0002-0 (cit. on p. 25).

Poulakis, Z., D. Valougeorgis, and C. Papadimitriou (2003). «Leakage detection in water pipe networks using a Bayesian probabilistic framework». In: *Probabilistic Engineering Mechanics* 18.4, pp. 315–327. DOI: 10.1016/S0266-8920(03)00045-6 (cit. on pp. 10, 60 sq., 149, 151, 165).

Pudar, R. and J. Liggett (1992). «Leaks in Pipe Networks». In: *Journal of Hydraulic Engineering* 118.7, pp. 1031–1046. DOI: 10.1061/(ASCE)0733-9429(1992)118:7(1031) (cit. on pp. 45 sq., 121, 147, 149, 151 sq., 160, 162).

Puust, R., Z. Kapelan, D. Savic, and T. Koppel (2010). «A review of methods for leakage management in pipe networks». In: *Urban Water Journal* 7.1, pp. 25–45. DOI: 10.1080/15730621003610878 (cit. on pp. 7–10, 45).

Quevedo, J., M. Cugueró, R. Pérez, F. Nejjari, V. Puig, and J. Mirats (2011). «Leakage location in water distribution neworks based on correlation measurement of pressure sensors». In: 8th IWA Symposium on System Analysis and Integrated Assessment (Watermatex). San Sebastian, Spain: International Water Association (IWA), pp. 290–297 (cit. on pp. 118, 125).

Ramalingam, D., S. Lingireddy, and L. Ormsbee (2002). «History of Water Distribution Network Analysis: Over 100 Years of Progress». In: *Environmental and Water Resources History*. American Society of Civil Engineers, pp. 55–67 (cit. on pp. 38 sq.).

Rastrigin, L. A. (1974). *Systems of extremal control*. Moscow: Mir (cit. on p. 35).

Rathi, S. and R. Gupta (2014). «Sensor Placement Methods for Contamination Detection in Water Distribution Networks: A Review». In: *Procedia Engineering*. 16th Water Distribution System Analysis Conference, WDSA2014 89, pp. 181–188. DOI: 10.1016/j.proeng.2014.11.175 (cit. on p. 118).

Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog. 182 pp. (cit. on p. 25).

Reynolds, O. (1883). «An Experimental Investigation of the Circumstances Which Determine Whether the Motion of Water Shall Be Direct or Sinuous, and of the Law of Resistance in Parallel Channels». In: *Philosophical Transactions of the Royal Society of London* 174, pp. 935–982 (cit. on p. 38).

Romano, M., Z. Kapelan, and D. Savić (2013). «Geostatistical techniques for approximate location of pipe burst events in water distribution systems». In: *Journal of Hydroinformatics* 15.3, p. 634. DOI: 10.2166/hydro.2013.094 (cit. on pp. 10, 79 sq., 150 sq.).

Rosenbrock, H. H. (1960). «An Automatic Method for Finding the Greatest or Least Value of a Function». In: *The Computer Journal* 3.3, pp. 175–184. DOI: 10.1093/comjnl/3.3.175 (cit. on p. 32).

Rossman, L. A. (2000). «EPANET 2: users manual». In: (cit. on pp. 21, 39 sq., 42, 61, 149 sq.).

Ruanaidh, J. J. K. O. and W. J. Fitzgerald (1996). *Numerical Bayesian Methods Applied to Signal Processing*. Statistics and Computing. New York: Springer-Verlag (cit. on pp. 47, 59).

Saludes, J., J. Quevedo, and V. Puig (2017). «Demand Forecasting for Real-Time Operational Control». In: *Real-time Monitoring and Operational Control of Drinking-Water Systems*. Advances in Industrial Control. Springer, Cham, pp. 99–111. DOI: 10.1007/978-3-319-50751-4_6 (cit. on p. 89).

Salvatier, J., T. V. Wiecki, and C. Fonnesbeck (2016). «Probabilistic programming in Python using PyMC3». In: *PeerJ Computer Science* 2, e55. DOI: 10.7717/peerj-cs.55 (cit. on p. 110).

Sanz Estapé, G. (2016). «Demand modeling for water networks calibration and leak localization». PhD thesis. Barcelona, Spain: Universitat Politècnica de Catalunya (cit. on pp. 43 sq., 122, 151, 193).

Sanz, G., R. Pérez, Z. Kapelan, and D. Savic (2015). «Leak Detection and Localization through Demand Components Calibration». In: *Journal of Water Resources Planning and Management*, p. 04015057. DOI: 10.1061/(ASCE)WR.1943-5452.0000592 (cit. on p. 149).

Sarrate Estruch, R., J. Blesa Izquierdo, F. Nejjari Akhi-Elarab, and J. Quevedo Casín (2013). «Sensor placement for leak detection and location in water distribution networks». In: (cit. on pp. 118 sq.).

Sarrate, R., J. Blesa, and F. Nejjari (2014). «Clustering techniques applied to sensor placement for leak detection and location in water distribution networks». In: *2014 22nd Mediterranean Conference of Control and Automation (MED)*. 2014 22nd Mediterranean Conference of Control and Automation (MED), pp. 109–114. DOI: 10.1109/MED.2014.6961356 (cit. on pp. 118 sq., 145).

Sarrate, R., J. Blesa, F. Nejjari, and J. Quevedo (2014). «Sensor placement for leak detection and location in water distribution networks». In: *Water Science and Technology: Water Supply* 14.5, pp. 795–803. DOI: 10.2166/ws.2014.037 (cit. on p. 118).

Sarrate, R., F. Nejjari, and A. Rosich (2012). «Sensor placement for fault diagnosis performance maximization in Distribution Networks». In: *2012 20th Mediterranean Conference on Control Automation (MED)*. 2012 20th Mediterranean Conference on Control Automation (MED), pp. 110–115. DOI: 10.1109/MED.2012.6265623 (cit. on pp. 118 sq., 128).

Savic, D., Z. Kapelan, and P. Jonkergouw (2009). «Quo vadis water distribution model calibration?» In: *Urban Water Journal* 6.1, pp. 3–22. DOI: 10.1080/15730620802613380 (cit. on pp. 70, 117 sq.).

Schaetzen, W. B. de, G. Walters, and D. Savic (2000). «Optimal sampling design for model calibration using shortest path, genetic and entropy algorithms». In: *Urban Water*. Developments in water distribution systems 2.2, pp. 141–152. DOI: 10.1016/S1462-0758(00)00052-2 (cit. on pp. 118, 123 sq., 139, 174).

Scolnicov, H. (2013). *A long journey to 12% Non-Revenue Water at Yarra Valley Water*. The Water Monitor. URL: https://takadu.wordpress.com/2013/03/02/a-long-journey-to-12-non-revenue-water-at-yarra-valley-water/ (visited on 04/24/2018) (cit. on p. 3).

Shinozuka, M., J. Liang, and M. Q. Feng (2005). «Use of supervisory control and data acquisition for damage location of water delivery systems». In: *Journal of engineering mechanics* 131.3, pp. 225–230 (cit. on pp. 149, 151).

Shumway, R. H. and D. S. Stoffer (2006). *Time series analysis and its applications: with R examples*. 2nd [updated] ed. Springer texts in statistics. OCLC: ocm63514199. New York: Springer. 575 pp. (cit. on pp. 50 sq., 53, 57).

Simpson, A. R. and S. Elhay (2011). «Jacobian Matrix for Solving Water Distribution System Equations with the Darcy-Weisbach Head-Loss Model». In: *Journal of Hydraulic Engineering* 137.6, pp. 696–700. DOI: 10.1061/(ASCE)HY.1943-7900.0000341 (cit. on pp. 41 sq.).

Soldevila, A., J. Blesa, S. Tornil-Sin, R. Fernandez-Canti, and V. Puig (2018). «Sensor placement for classifier-based leak localization in water distribution networks using hybrid feature selection». In: *Computers & Chemical Engineering* 108, pp. 152–162. DOI: 10.1016/j.compchemeng.2017.09.002 (cit. on p. 119).

Sophocleous, S., D. Savic, Z. Kapelan, Y. Shen, and P. Sage (2016). «A Graph-based Analytical Technique for the Improvement of Water Network Model Calibration». In: *Procedia Engineering*. 12th International Conference on Hydroinformatics (HIC 2016) - Smart Water for the Future 154, pp. 27–35. DOI: 10.1016/j.proeng.2016.07.415 (cit. on p. 46).

Steffelbauer, D. B. and D. Fuchs-Hanusch (2015). «OOPNET: An object-oriented EPANET in Python». In: *Procedia Engineering*. Computing and Control for the Water Industry (CCWI2015) Sharing the best practice in water management 119, pp. 710–718. DOI: 10.1016/j.proeng.2015.08.924 (cit. on pp. 15, 60, 130, 153, 158).

– (2016a). «Efficient Sensor Placement for Leak Localization Considering Uncertainties». In: *Water Resources Management* 30.14, pp. 5517–5533. DOI: 10.1007/s11269-016-1504-6 (cit. on pp. 14, 117, 127 sqq., 131–137, 174).

– (2016b). «Fitness landscapes and distance metrics for model-based leakage localization». In: 3rd International Conference on Control and Fault-Tolerant Systems, SysTol'16. Barcelona, Spain (cit. on pp. 15, 152, 165).

Steffelbauer, D. B., M. Günther, and D. Fuchs-Hanusch (2017). «Leakage Localization with Differential Evolution: A Closer Look on Distance Metrics». In: *Procedia Engineering* 186, pp. 444–451. DOI: 10.1016/j.proeng.2017.03.251 (cit. on pp. 15, 152, 165).

Steffelbauer, D. B., M. Günther, M. Neumayer, and D. Fuchs-Hanusch (2014a). «Leakage Localization In Virtual District Metered Areas With Differential Evolution». In: *International Conference on Hydroinformatics* (cit. on pp. 15, 152).

– (2014b). «Sensor Placement and Leakage Isolation with Differential Evolution». In: *World Environmental and Water Resources Congress 2014*. American Society of Civil Engineers, pp. 408–416 (cit. on pp. 15, 152).

Steffelbauer, D. B., M. Neumayer, M. Günther, and D. Fuchs-Hanusch (2014). «Sensor Placement and Leakage Localization Considering Demand Uncertainties». In: *Procedia Engineering*. 16th Water Distribution System Analysis Conference, WDSA2014 Urban Water Hydroinformatics and

Strategic Planning 89, pp. 1160–1167. DOI: 10.1016/j.proeng.2014.11.242 (cit. on pp. 15, 117, 152).

Storn, R. (1996). «On the usage of differential evolution for function optimization». In: *Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American*. Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American, pp. 519–523. DOI: 10.1109/NAFIPS.1996.534789 (cit. on pp. 29, 34, 37).

Storn, R. and K. Price (1995). *Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces*. Google-Books-ID: ypNYmwEACAAJ. ICSI. 12 pp. (cit. on pp. 28 sqq.).

– (1997). «Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces». In: *Journal of Global Optimization* 11.4, pp. 341–359. DOI: 10.1023/A:1008202821328 (cit. on pp. 28 sqq.).

Strang, G. (1986). *Introduction to Applied Mathematics*. Wellesley-Cambridge Press. 776 pp. (cit. on pp. 21, 43).

Thornton, J., R. Sturm, and G. Kunkel (2008). *Water loss control*. Vol. 2. McGraw-Hill New York (cit. on pp. 5 sqq.).

Todini, E. and S. Pilati (1987). «A gradient algorithm for the analysis of pipe networks». In: ed. by B. Coulbeck and C. Orr. Taunton, UK, UK: Research Studies Press Ltd., pp. 1–20 (cit. on pp. 39, 43).

Tzatchkov, V. G., V. H. Alcocer-Yamanaka, and V. Bourguett Ortíz (2006). «Graph Theory Based Algorithms for Water Distribution Network Sectorization Projects». In: *Water Distribution Systems Analysis Symposium 2006*. Proceedings. DOI: 10.1061/40941(247)172 (cit. on p. 46).

Vuuren, L van (2014). «Water loss : are we wasting our way into a potential water crisis? : water demand management». In: *Water Wheel* 13.6, pp. 34–37 (cit. on p. 3).

Walski, T. M. (2000). «Model calibration data: the good, the bad, and the useless». In: *Journal - American Water Works Association* 92.1, p. 94 (cit. on p. 66).

– (2006). «A History of Water Distribution». In: 98.3, pp. 110–121 (cit. on pp. 38 sq.).

Walski, T. M., D. V. Chase, D. Savic, W. Grayman, S. Beckwith, and E. Koelle (2003). *Advanced Water Distribution Modeling and Management* (cit. on pp. 41 sq., 84).

Williams, A. P., R. Seager, J. T. Abatzoglou, B. I. Cook, J. E. Smerdon, and E. R. Cook (2015). «Contribution of anthropogenic warming to California drought during 2012–2014». In: *Geophysical Research Letters* 42.16, 2015GL064924. DOI: 10.1002/2015GL064924 (cit. on p. 2).

Wolpert, D. H. and W. G. Macready (1997). «No free lunch theorems for optimization». In: *IEEE Transactions on Evolutionary Computation* 1.1, pp. 67–82. DOI: 10.1109/4235.585893 (cit. on pp. 21, 155).

Wolski, P. (2018). «How severe is Cape Town's "Day Zero" drought?» In: *Significance* 15.2, pp. 24–27. DOI: 10.1111/j.1740-9713.2018.01127.x (cit. on p. 2).

Wood, D. J. and C. O. A. Charles (1972). «Hydraulic Network Analysis Using Linear Theory». In: *Journal of the Hydraulics Division* 98.7, pp. 1157–1170 (cit. on p. 39).

Wright, S. (1932). «The roles of mutation, inbreeding, crossbreeding and selection in evolution». In: *Proceedings of the Sixth International Congress of Genetics* 1, pp. 356–366 (cit. on p. 154).

Wu, Y. and S. Liu (2017). «A review of data-driven approaches for burst detection in water distribution systems». In: *Urban Water Journal* 14.9, pp. 972–983. DOI: 10.1080/1573062X.2017.1279191 (cit. on p. 81).

Wu, Y., S. Liu, K. Smith, and X. Wang (2018). «Using Correlation between Data from Multiple Monitoring Sensors to Detect Bursts in Water Distribution Systems». In: *Journal of Water Resources Planning and Management* 144.2, p. 04017084. DOI: 10.1061/(ASCE)WR.1943-5452.0000870 (cit. on p. 81).

Wu, Z., P. Sage, and D. Turtle (2009). «Pressure-Dependent Leak Detection Model and Its Application to a District Water System». In: *Journal of Water Resources Planning and Management* 136.1, pp. 116–128. DOI: 10.1061/(ASCE)0733-9496(2010)136:1(116) (cit. on pp. 149, 151, 162).

Wylie, E. B. and V. L. Streeter (1978). *Fluid transients*. Google-Books-ID: JvtRAAAAMAAJ. McGraw-Hill International Book Co. 416 pp. (cit. on pp. 44 sq.).

Ye, G. and R. A. Fenner (2011). «Kalman Filtering of Hydraulic Measurements for Burst Detection in Water Distribution Systems». In: *Journal of Pipeline Systems Engineering and Practice* 2.1, pp. 14–22. DOI: 10.1061/(ASCE)PS.1949-1204.0000070 (cit. on p. 80).

Zheng, F., A. R. Simpson, A. C. Zecchin, and J. Deuerlein (2013). «A graph decomposition-based approach for water distribution network optimization». In: *Water Resources Research* 49.4, pp. 2093–2109. DOI: 10.1002/wrcr.20175 (cit. on p. 46).