UNIVERSITY OF BELGRADE

FACULTY OF CIVIL ENGINEERING

Željko A. Vasilić

DECISION SUPPORT ALGORITHMS FOR SECTORIZATION OF WATER DISTRIBUTION NETWORKS

Doctoral Dissertation

Belgrade, 2018

УНИВЕРЗИТЕТ У БЕОГРАДУ

ГРАЂЕВИНСКИ ФАКУЛТЕТ

Жељко А. Василић

АЛГОРИТМИ ЗА ПОДРШКУ ОДЛУЧИВАЊУ ПРИ СЕКТОРИЗАЦИЈИ МРЕЖА ПОД ПРИТИСКОМ

докторска дисертација

Београд, 2018

SUPERVISOR AND THE COMMITTEE:

Supervisor:

Assoc. Prof. Miloš Stanić, PhD Civ. Eng.
University of Belgrade, Faculty of Civil Engineering

Committee:

Assoc. Prof. Miloš Stanić, PhD Civ. Eng.
University of Belgrade, Faculty of Civil Engineering

Prof. Dušan Prodanović, PhD Civ. Eng.
University of Belgrade, Faculty of Civil Engineering

Prof. Zoran Kapelan, PhD Civ. Eng.
University of Exeter, College of Engineering, Mathematics and Physical Sciences, United Kingdom

Assoc. Prof. Tina Dašić, PhD Civ. Eng.
University of Belgrade, Faculty of Civil Engineering

Assistant Prof. Branislav Babić, PhD Civ. Eng.
University of Belgrade, Faculty of Civil Engineering

Dissertation defense date: ________________________

ПОДАЦИ О МЕНТОРУ И ЧЛАНОВИМА КОМИСИЈЕ:

Ментор:

В. проф. др Милош Станић, дипл. грађ. инж.
Универзитет у Београду, Грађевински факултет

Чланови комисије:

В. проф. др Милош Станић, дипл. грађ. инж.
Универзитет у Београду, Грађевински факултет

Проф. др Душан Продановић, дипл. грађ. инж.
Универзитет у Београду, Грађевински факултет

Проф. др Зоран Капелан, дипл. грађ. инж.
Универзитет у Exeter – у , Велика Британија

В. проф. др Тина Дашић, дипл. грађ. инж.
Универзитет у Београду, Грађевински факултет

Доцент др Бранислав Бабић, дипл. грађ. инж.
Универзитет у Београду, Грађевински факултет

Датум одбране докторске дисертације:______________________

Захвалница

Највећу захвалност за успешан завршетак ове дисертације дугујем свом ментору, проф. др Милошу Станићу, који ме је увео у свет научно истраживачког рада и кроз њега водио на прави начин. Неизмерно сам му захвалан на подршци, стрпљењу и посвећености током истраживања за свакодневне дискусије. Захваљујем се проф. др Зорану Капелану на спремности за заједничку сарадњу и сугестијама које ми је небројено пута пружио током овог истраживања. Посебну захвалност дугујем проф. др Душану Продановићу на подршци у претходном периоду, пажљивом исчитавању дисертације и предлозима за њено побољшање. Хвала и осталим члановима Комисије чији коментари су ми помогли у писању дисертације.

Желео бих да се захвалим свим колегама са Института за хидротехнику на позитивном радном окружењу. Задовољство је бити део таквог колектива. Посебно бих се захвалио Ањи и Љиљи које су ме храбриле у тренуцима када ми је понестајало мотивације. Дамјану и Буди се захваљујем на корисним дискусијама и помоћи у решавању практичних проблема.

Неизмерну захвалност за безрезервну подршку током година дугујем својој породици и пријатељима. Мајци Снежани се захваљујем на безграничној родитељској љубави коју ми је пружила. Тијани сам захвалан на емотивној подршци и разумевању које је имала током писања ове дисертације, када живот са мном није био једноставан.

Београд, 2018

Жељко

DECISION SUPPORT ALGORITHMS FOR SECTORIZATION OF WATER DISTRIBUTION NETWORKS

Abstract

Many water utilities, especially ones in developing countries, continue to operate low efficient water distribution networks (WDNs) and are consequently faced with significant amount of water (e.g. leakage) and revenue losses (i.e. non-revenue water – NRW). First step in reducing the NRW is assessment of water balance in WDN aimed to establish the baseline level of water losses. Then, water utilities can plan NRW reduction activities according to this baseline. Sectorization of WDN into District Metered Areas (DMAs) is the most cost-effective strategy used for active leakage (i.e. water loss) control, achieved by monitoring the flow data on DMAs' boundaries. Sectorization of WDN has to be designed carefully, as required network interventions can endanger network's water supply and pressure distribution.

In this thesis new methods and algorithms, aimed to support making more effective and objective decisions regarding the WDN sectorization procedure, are presented, tested and validated. Presented methods and algorithms are part of proposed decision support methodology compensating for disadvantages in available methods, valuable to practicing engineers commencing implementation of sectorization strategy in WDN.

Main sectorization objective adopted in methodology presented in this thesis is to design layout of DMAs that will allow efficient tracking of water balance in the network. Least investment for field implementation and maintaining the same level of WDN's operational efficiency are adopted as main design criteria. New sectorization algorithm, named DeNSE (Distribution Network SEctorization), is developed and presented, adopting above-named objective and design criteria. DeNSE algorithm utilizes newly developed uniformity index which drives the sectorization process and identifies clusters. New engineering heuristic is developed and used for placing the flow-meters and isolation valves on clusters' boundary edges, making them DMAs. Post-sectorization operational efficiency of WDN is evaluated using adopted performance indicators (PIs). Top-down approach to hierarchical sectorization of WDN, particulary convenient for water utilities constrained with limited funding and insufficient reliable input data, is also implemented in DeNSE algorithm. New method for hydraulic simulation, named TRIBAL-ΔQ is developed to address the issue of low computational efficiency, recognized in available sectorization methodologies employing optimization. TRIBAL-ΔQ is a loop-flow based method which combines the novel TRIangulation Based ALgorithm (TRIBAL) for loop identification with efficient implementation of the loop-flow hydraulic solver (ΔQ).

TRIBAL-ΔQ method is tested on various networks of different complexities and topologies. This thesis reports only results of testing on literature benchmark networks, used to validate methods' performance. TRIBAL-ΔQ method based hydraulic solver is compared to the node based solver implemented in EPANET, most prominent software for hydraulic calculation of WDN. New TRIBAL-ΔQ solver showed significant dominance in computational efficiency, with stable numerical performance and same level of prediction accuracy.

DeNSE algorithm is benchmarked against other available sectorization methodologies on real-sized WDN. Obtained results demonstrate the ability of DeNSE algorithm to identify good set of feasible solutions, without worsening operational status of the WDN compared to its baseline condition. Reported computational efficiency of the algorithm is one of its strong points, as it allows generation of feasible solutions for large WDN in reasonable time. In this field, algorithm particularly outperforms methods employing multi-objective optimization (e.g. minutes compared to hours).

Key words: Sectorization, water distribution network, hydraulic simulation, district meter areas, WDN, DeNSE, TRIBAL-ΔQ, DMA, loop-flow

Scientific field: Civil Engineering

Scientific subfields: Hydroinformatics, Fluid Mechanics and Hydraulics

UDC: 624:532(043.3)

АЛГОРИТМИ ЗА ПОДРШКУ ОДЛУЧИВАЊУ ПРИ СЕКТОРИЗАЦИЈИ МРЕЖА ПОД ПРИТИСКОМ

Сажетак

Комунална предузећа која управљају водоводним системима, нарочита она у земљама у развоју, суочена су са проблемима дотрајале и лоше одржаване дистрибутивне мрже који за последицу имају значајне количине воде која се губи у дистрибуцији. Први корак ка смањењу губитака у водоводном систему је процена водног биланса у дистрибутивној мрежи како би се утврдило почетно стање система, а затим и приступило планирању и предузимању мера за смањење губитака како би се то стање поправило. Најисплативија, и опште прихваћена, стратегија за остваривање овог циља је подела дистрибутивне мреже, односно њена секторизација, на тзв. основне зоне билансирања (ОЗБ). ОЗБ се у мрежи успостављају јасним дефинисањем њихових граница, на којима се инсталирају изолациони затварачи и мерачи протока. Избор ОЗБ није једнозначан, и приликом њиховог дефинисања мора се водити рачуна о планираним интервенцијама у мрежи које могу имати негативан утицај на водоснабдевање потрошача и распоред притисака у мрежи.

У овој дисератацији су приказане и тестиране нове методе и алгоритми намењени за подршку одлучивању приликом секторизације водоводне дистрибутивне мреже на ОЗБ. Презентоване методе и алгоритми надомешћују недостатке постојећих метода и могу бити од користи инжењерима који се у пракси баве задатком секторизације дистрибутивних мрежа.

Основни циљ методологије за секторизацију приказане у овој дисертацији је дефинисање распореда ОЗБ који ће омогућити ефикасно праћење водног биланса у дистрибутивној мрежи. Основни критеријуми за вредновање и избор оптималног решења су минимална улагања у неопходне интервенције у мрежи и очување поузданости система. У дисертацији је приказан нови алгоритам за секторизацију водоводне мреже, назван DeNSE (Distribution Network SEctorization), заснован на претходно наведеном основном циљу и критеријумима. Секторизација применом DeNSE алгоритма је базирана на употреби новог

индекса униформности мреже, који омогућава идентификацију зона у мрежи уједначених према потрошњи. За дефинисање ОЗБ, на границе претходно идентификованих зона потребно је поставити мераче протока и изолационе затвараче. За ове потребе развијена је и приказана методлогија засновна на практичним инжењерским принципима. За процену поузданости система након секторизације коришћени су усвојени индикатори перформанси (PIs – Performance Indicators). Предвиђена је и могућност за хијерархијску секторизацију дистрибутивне мреже, нарочито привлачна за комунална предузећа која располажу ограниченим финансијским средствима и имају потребу да процес секторизације изведу у неколико фаза. Услед проблема са значајним рачунарским временом који имају постојеће методе за секторизацију које користе оптимизацију, у оквиру истраживања је развијен и нови метод за хидраулички прорачун мрежа под притиском, назван TRIBAL-ΔQ. TRIBAL-ΔQ метод је заснован на примени новог алгоритма за идентификацију прстенова у мрежи базираног на триангулацији (TRIBAL – TRIangulation Based ALgorithm) и ефикасној имплементацији нумеричког модела хидрауличког прорачуна базираног на методи прстенова (ΔQ).

TRIBAL-ΔQ метод је тестиран на бројним дистрибутивним мрежама различите сложености. У овој дисертацији су приказани само резултати добијени применом на тест-мрежама познатим из литературе, како би се потврдила њихова ваљаност. TRIBAL-ΔQ метод је упоређен са методом коју користи најпознатији софтвер за хидраулички прорачун мрежа под притиском – EPANET. Резултати приказују значајну предност новог метода у погледу рачунарске ефикаснонсти, уз очување нумеричке стабилности и тачности решења хидрауличког прорачуна.

DeNSE алгоритам је упоређен са постојећим методама за секторизацију дистрибутивних мрежа. Резултати потврђују да је нови алгоритам у стању да идентификује скуп могућих решења, која не угрожавају поузданост система и снабдевање потрошача. Рачунарска ефикаснонст DeNSE алгоритма је једна од његових најзначајнијих предности јер омогућава идентификацију не једног, већ скупа могућих решења за реалне дистрибутивне мреже у релативно кратком

рачунарском времену. Ова чињеница посебно долази до изражаја када се рачунарско време DeNSE алгоритма упореди са рачунарским временом метода које користе оптимизационе алгоритме (минути у поређењу са сатима).

Кључне речи: Секторизација, алгоритам, дистрибутивна мрежа, хидраулички прорачун, основне зоне билансирања, ОЗБ, DeNSE, TRIBAL-ΔQ, метода прстенова

Научна област: Грађевинарство

Уже научне области: Хидроинформатика, Механика нестишљивих флуида и хидраулика

УДК: 624:532(043.3)

PUBLICATION LIST FROM THIS RESEARCH

Journal papers:

Vasilić, Ž., Stanić, M., Kapelan, Z., Ivetić, D., & Prodanović, D. (2018). Improved Loop-Flow Method for Hydraulic Analysis of Water Distribution Systems. *Journal of Water Resources Planning and Management (ASCE)*, *144*(4), 1–11. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000922

Ivetić, D., **Vasilić, Ž**., Stanić, M., & Prodanović, D. (2016). Speeding up the water distribution network design optimization using the ΔQ method. *Journal of Hydroinformatics*, *18*(1), 33-48. https://doi.org/10.2166/hydro.2015.118

Conference papers:

Vasilić, Ž., Stanić, M., Kapelan, Z. & Prodanović, D. (2018). Advanced Loop-flow Method for Fast Hydraulic Symulations. 13th International Hydroinformatics Conference – HIC2018, 1 – 6 July 2018. Palermo (Italy)

Vasilić, Ž., Stanić, M., Prodanović, D., & Kapelan, Z. (2017). Network Sectorization Through Aggregation of Strong Connected Components. In proceedings from 18th International Conference on Water Distribution Systems – WDSA2016, 24 – 28 July 2016. Cartagena (Colombia), Procedia Engineering, 186, 244–251. https://doi.org/10.1016/j.proeng.2017.03.238

Vasilić, Ž., Stanić, M., Ivetić, D., & Prodanović, D. (2015). Improving the Hydraulic Analysis of a Water Distribution Network with the ΔQ Method Using Automatic Identification of Minimal Basis Loops. In 7th IWA Eastern European Young Water Professional Conference, 17 – 19 September 2015. Belgrade, Serbia.

TABLE OF CONTENTS:

LIST OF FIGURES:

LIST OF TABLES:

LIST OF ABBREVIATIONS

ASL - Arbitrary Set of Loops

BFS - Breadth First Search

BWSN - Battle of the Water Sensor Networks

CDT - Constrained Dalunay Triangulation

CEs - Constrained Edges

CTM - Co-Tree Method

DAG - Directed Acyclic Graph

DeNSE - Distribution Network SEctorization

DFS - Depth First Search

DIGRAPH - DIrectional GRAPH

DLL - Dynamic Link Library

DMA - District Meter Area

DT - Delaunay Triangulation

DW - Darcy-Weisbach

eDMAs - Elementary DMAs

EPA - Environmental Protection Agency

FCPA - Forest Core Partitioning Algorithm

GA - Genetic Algorithm

GGA - Global Gradient Algorithm

GMPA - Graph Matrix Partitioning Algorithm

HW - Hazen-Williams

i-DMAS - Isolated DMAs

IWA - International Water Association

MCB - Minimum Cycle Basis

MDD - Maximum Day Demand

MLRB - Multi-Level Recursive Bisection

MNL - Main Network Layout

MO - Multi Objective

NCEs - Non-Constrained Edges

NDFS - Nested Depth First Search

NR - Newton Raphson

NR-GA - Newton Raphson-Gradient Algorithm

NR-LF - Newton Raphson Loop Flow

NRW - Non-Revenue Water

NZE - Non-Zero Elements

OF - Objective Function

ONL - Original Network Layout

PC - Personal Computer

PFS - Priority First Search

PIs - Performance Indicators

RCTM - Reformulated Co-Tree Method

SCCs - Strongly Connected Components

ST - Spanning Tree

TG - Triangles Graph

TRIBAL - TRIangulation BAsed Loops

TSL - Topologically Sorted List

WAA - Water Authorities Association

WDN - Water Distribution Network

WRC - Water Research Center

CHAPTER 1:

INTRODUCTION

1 INTRODUCTION

Table of contents:

1.1 BACKGROUND

Many water utilities, especially those in developing countries, continue to operate low efficient water distribution networks (WDNs) and are consequently faced with significant amount of water and revenue losses Babić et al.(2014). Factors contributing to high water losses in WDN are various: poor infrastructure, high pressures in the network, illegal water usage etc. Generally, they are divided into apparent and real losses (e.g. leakage). Together with unbilled authorized consumption, water losses make up the non-revenue water (NRW) in WDN. It is reported that in some cases NRW percentage is as high as 50% of total water entering the WDN (Kanakoudis et al., 2011). NRW from WDNs worldwide is estimated at 48 billion m^3 per year (Kingdom et al., 2006), most of it accounted in developing countries. Beside significantly high NRW in developing countries, ratio of apparent and real losses in NRW are usually similar. In developed countries NRW is mostly caused by real losses.

Water companies can significantly reduce NRW by employing available methodologies for WDN benchmarking and water losses control. Audit methodology aimed for assessment of WDN efficiency was suggested by International Water Association (IWA) and published in Alegre et al. (2006). It includes standardized methodology for water balance assessment and database of 170 performance indicators (PIs), whose calculation is based on 232 variables that have to be monitored in WDN. Direct implementation of IWA methodology led to a number of difficulties in practice due to large discrepancies in the development of WDNs, local conditions and characteristics, data availability and reliability, operational practices etc. Tailoring of IWA methodology, through modification of existing and introduction of new PIs, was required in order to properly implement it (Babić et al., 2014; Kanakoudis et al., 2011).

First step in reducing the NRW is assessment of water balance in WDN aimed to establish the baseline level of water losses. Then, water utilities can plan NRW reduction activities according to this baseline. NRW reduction strategy for apparent losses is very much dependent on local socio-economic and political aspects. Improving customer meter accuracy, reading and billing of consumption and most importantly, rising public awareness about the importance of good governance, are some of the measures that can be implemented. On the other hand, strategy for real losses is strictly a set of technical measures that can be implemented in any WDN, such as: active leakage control (i.e. continuous monitoring of flows in the network to detect leaks and prioritize interventions), infrastructure management (e.g. rehabilitation plans) and pressure management.

1.2 Sectorization of Water Distribution Network

Sectorization of WDN into zones (sectors, clusters or District Metered Areas - DMAs) has become the most cost-effective strategy for the control of real water losses. A DMA is defined as a distinct hydraulic area of the WDN, separated from the rest of the supply system by isolation valves or pressure reducing valves and one or more metered inlets and outlets (Burrows et al., 2000). Active leakage control is carried out by monitoring installed flow meters' data and pressure can be managed using installed valves. Installation of valves and flow meters inevitably affects WDN's topology and can possibly endanger networks' operational performance. Network interventions have to be designed and implemented carefully as it must be ensured that they do not jeopardize water supply reliability and quality.

Sectorization strategy was introduced in the United Kingdom in the late 80's and has been successfully implemented in many WDNs worldwide since. Each WDN has unique topology and characteristics, meaning there is no common design procedure for

WDN sectorization. Series of guidelines provided by different water authorities are available to engineers to support their design making process (Butler, 2000; Farley, 2001; Morrison et al., 2007; WAA & WRC, 1985). In practice, sectorization process starts with the identification of key sectorization objectives (e.g. monitoring of water balance in network, reduction of network pressures, reduction of leakage) and design criteria, followed by the identification of performance indicators (PIs) that will be used to assess impact of implemented interventions in the network.

Designing an optimal sectorization solution for existing and operating WDN is an extremely hard task to do, which still usually assumes manual "trial an error" approach conducted by local experts with good knowledge of the WDN's specifics. Engineering reasoning is, although valuable, still very subjective and can produce arbitrary sectorization solutions far from the optimal one. With ever increasing computational power, the use of optimization methods seemed like a logical next step. In the past 10 years many different algorithms for automatic sectorization of WDN, employing certain type of optimization, have been presented in scientific literature. Numerous objectives and constraints were added with each new method in the attempt to better describe sectorization problem. Extensive lists of objectives and constraints only highlighted a well-known problem of all optimization methods – computational burden. Solution search space exponentially increases with the complexity of a network, and perhaps this is why recently presented methods employing optimization are lacking results supporting their application on real-sized networks. Adequate balance between engineering judgement and available state-of-the-art optimization methods is yet to be found.

Employing optimization requires multiple hydraulic simulations to calculate PIs, adopted when sectorization objectives were initially set. The efficiency of hydraulic solver adopted by different sectorization algorithms, which inevitably affects computational burden of entire optimization procedure, is not discussed at all. All available algorithms for automatic sectorization use Global Gradient Algorithm (GGA) presented by Todini & Pilati (1987), accepted as most prominent solver for node based system of equations describing hydraulics of WDN. Unrelated to sectorization problem, number of papers have recently suggested solvers based on loop-flow formulation of system of equations as an alternative to node based ones, highlighting their dominance in computational efficiency if implemented properly.

Shortage of system and flow information, encountered in poorly managed and low efficient WDNs, is usually not addressed with existing methodologies for sectorization. Water utilities managing such WDNs usually do not have sufficient funds to invest in large number DMAs at once, so sectorization strategy should be planned hierarchically and implemented in phases. Establishing a few DMAs in WDN should enable tracking of water balance in the network and gathering basic data about system dynamics, without significant effect on network's operational conditions. Initially established DMAs can be further partitioned to obtain finer sectorization resolution, which will in turn enable better leakage control and pressure management.

1.3 OVERALL AIM OF THE THESIS

Previous discussion reveals that available sectorization methodologies are more suitable for well managed and monitored WDNs. Usually they require too many input data, often lacking in poorly operated WDNs in developing countries. Low computational efficiency problems, imposed by using optimization methods, restrict full applicability to real sized WDNs which is yet to be proven.

Overall aim of this thesis is to develop, test, validate and demonstrate new methods to support making more effective and objective decisions regarding the WDN sectorization. Primary sectorization objective is to design such DMAs layout that will allow efficient tracking of water balance in the network. Least investment for field implementation and maintaining the same level of WDN's operational efficiency are main design criteria.

The aim was achieved through following specific objectives:

1. To develop sectorization algorithm that will, beside general recommendations given by aforementioned design guidelines, include some heuristic engineering principles relevant to WDN,

2. To develop new, or improve existing method for hydraulic simulation used to solve network hydraulics, which will consequently improve computational efficiency of sectorization design procedure,
3. To consider possible implementation of hierarchical sectorization and
4. To benchmark proposed methodology on number of case studies and validate its results by comparison to other available methodologies.

1.4 OUTLINE OF THE THESIS

Chapter 2 presents literature review of available methods for network sectorization and hydraulic simulation. The main focus of this chapter is to summarize previous research done on this thematic and identify key knowledge gaps. Research questions to be answered in this thesis, as well as hypothesis and methods used, are given at the end of the chapter.

Chapter 3 addresses second specific objective listed above. This chapter provides presentation of improved method for hydraulic simulation based on the loop-flow formulation of governing equations for pressure and flow distribution in the network. New algorithm for identification of the loops in the network, which is a prerequisite for loop-flow based methods, is presented in this chapter. Chapter is concluded with detailed explanation of method’s implementation, which enables high computational efficiency to be achieved.

Chapter 4 addresses first and third specific objective. This chapter presents new algorithm for network sectorization based on newly introduced uniformity index and engineering heuristic. Presentation of the sectorization algorithm is followed by explaining the implementation of improved method for hydraulic simulation, presented in Chapter 3. Chapter is concluded with discussion on extensions of the developed algorithm, including hierarchical sectorization and optimization.

Chapter 5 addresses fourth specific objective giving benchmarking results of algorithms and methods presented in chapters 3 and 4. Improved method for hydraulic simulation (presented in Chapter 3) is tested on 4 case study networks of different topology and complexity. Performance of new sectorization algorithm (presented in Chapter 4) is evaluated through benchmarking on large real-sized WDN, well known and often used in the literature for various modeling tasks. Obtained results are validated by comparison with other available methods.

Chapter 6 summarize thesis’ key findings and conclusions, proposing further research.

CHAPTER 2:

LITERATURE REVIEW

2 LITERATURE REVIEW

Table of contents:

2.1 INTRODUCTION

This chapter presents a broader literature review on research topics covered in this thesis. As indicated in introductory chapter, overall aim of the research is to develop a decision support methodology for successful implementation of sectorization strategy in WDNs. Section 2.2 summarizes previous research related to WDN sectorization problem. Following general discussion on sectorization and DMAs, adequate size of a DMA (2.2.1) and common sectorizaton design procedure (2.2.2) are discussed. Section is concluded with an overview of available methods for WDN sectorization (2.2.3), differentiating sectorization support tools (2.2.3.1) and fully automatic algorithms (2.2.3.2).

Primary research focus is to design the cost effective sectorization solution which will not affect water supply and normal operating conditions in WDN. Secondary requirement is identifying solution in reasonable computational time, which would allow engineers to analyze lot more different solutions and come up with a better overall solution. This would provide practical applicability of methodology to real-sized WDNs. Achieving this goal is possible by improving the computational efficiency of the hydraulic solver used to perform hydraulic calculations. Previous research related to hydraulic simulation of WDN is presented in section 2.3. First part of this section presents basics of mathematical modeling of WDN (2.3.1). Second part (2.3.2) gives historical overview of fundamental modelling methods (2.3.2.1), their systematization and summarization of advantages and drawbacks (2.3.2.2) and notable improvements in the efficiency of hydraulic solvers made in recent years (2.3.2.3). Section is concluded with overview of available loop identification procedures required to solve network hydraulics based on the loop-flow approach (2.3.3).

Literature review presented in this chapter is focused on identification of knowledge gaps in existing methods and room for possible improvements. Main research questions to be answered in this thesis are summarized in Section 2.4 followed by the presentation of working hypothesis.

2.2 Sectorization of Water Distribution Network

Sectorization of water distribution network (WDN) into zones (sectors, clusters or District Metered Areas - DMAs) has become one of the main strategies for efficient management of WDNs. It was introduced in the United Kingdom in the late 80’s and has been implemented in many WDNs worldwide since. Decomposition has been done traditionally to address two main objectives:

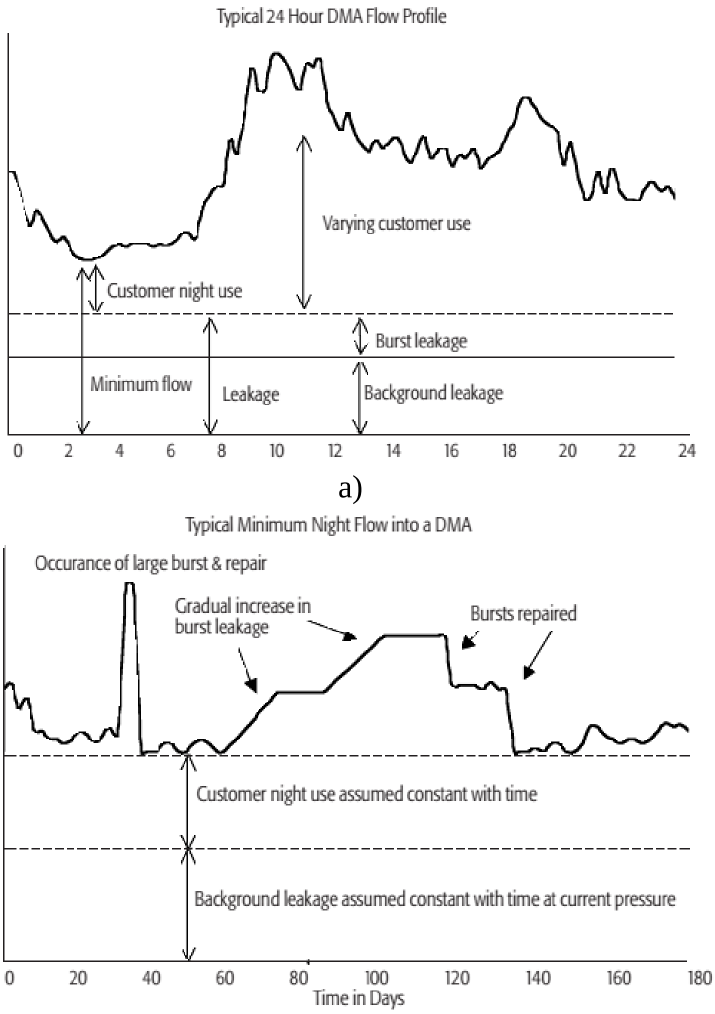
1) better control of water losses in the network and

2) efficient management of pressures in the network.

First objective can be achieved solely by means of observing all inflows and outflows from the zone, especially in the night time when anomalies in the water balance can point to the existence of significant water losses within the DMA. This is illustrated on Figure 2.1, which shows typical 24-h DMA flow profile (a) and typical minimum night flow into a DMA during a longer period of time (b). From the Figure 2.1-a it is clear that the leakages, especially burst ones, are easier to detect during the night, as customer night use is relatively low and constant making it comparable to the leakage rate. During the day time, the customer use is increased and notably varying, also being significantly higher than the leakages, making it almost impossible to separate leakages from the consumption. Figure 2.1-b illustrates occurrence of large (accidental) and gradual (slowly increased in time) bursts, successfully detected via observation of the night time flow data and repaired. Second objective is achieved with the installation of isolation (boundary) valves, which separate previously connected parts of network, now enabling different pressure levels to be maintained within newly created zones.

Best definition of a DMA, given by Burrows et al. (2000), is that it is a distinct hydraulic area of the WDN, separated from the rest of the supply system by isolation valves and one or more metered inlets and outlets. There are two main types of DMAs (Farley, 2001): a) isolated DMAs with one or multiple feeds and b) DMAs that cascade into adjacent DMAs. Typical DMAs design options are shown in Figure 2.2. DMAs can be permanent or temporary with the reference to the time-frame for which they are intended (Di Nardo & Di Natale, 2011). Sectorization can be carried out in different levels of details, and it is better to adopt hierarchical sectorization applicable for different purposes. Hierarchical sectorization is useful in situations where network is naturally hierarchically ordered and each identified DMA can be further partitioned to obtain finer division (Schaeffer, 2007). However, only one paper addresses the concept of hierarchical sectorization (Scarpa et al. 2016). Traditionally, DMAs are designed as permanent, but recently the concept of dynamic DMAs is presented (Wright et al. 2014) that implies the use of status changing boundary valves. In this manner, network reverts back to the original DMA design only at night for leakage detection purposes and preserves its original topology during the day in order to avoid possible negative effects introduced by the creation of DMAs, which will be discussed further down.

Generally speaking, network sectorization is in conflict with the main design criteria used in past when existing WDNs were designed and expanded – water supply reliability. To achieve high reliability, WDNs are historically designed as extremely looped systems with high pipe redundancy, and decomposition into DMAs which inevitably requires closure of some pipes, can considerably affect their topology. Network interventions required for network sectorization into DMAs (installation of isolation valves and flow metering devices) have to be implemented carefully, as they can jeopardize the network supply reliability, water quality, fire-flow supply and system response in the case of accidental bursts and other failures. For example, water quality will be affected as installation of valves will cause longer water retention time in some parts of the network.



a)

b)

Figure 2.1: Using DMAs to detect leakages: a) Typical 24-h DMA flow profile; b) Typical minimum night flow into a DMA (adapted from Morrison (2004))

Designing an optimal system of DMAs for the existing and operating WDN is a hard task to do. Every WDN is unique in its topology and characteristics so there is no common procedure for performing its decomposition, but rather a series of guidelines provided by the different water authorities (Butler, 2000; Farley, 2001; Morrison et al., 2007; WAA & WRC, 1985) and used in this process by practice engineers. If planning of DMAs (e.g. their number and size) is carried out during the new WDN design phase, it is much easier to come up with the solution that will be efficient both in terms of 1) sectorization main objectives and 2) satisfaction of network's hydraulic and other requirements. However, in the case of existing and already operating WDN, problem is much more complex due to the aforementioned influence of network interventions on its performance and many existing arrangements that have been introduced at different points in time, typically to address specific issues.

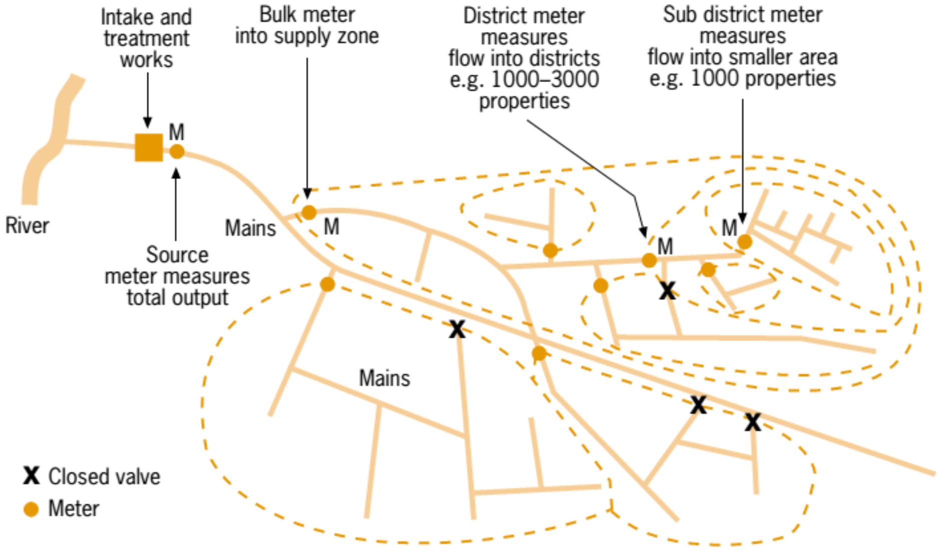

Intake and treatment works
Bulk meter into supply zone
District meter measures flow into districts e.g. 1000–3000 properties
Sub district meter measures flow into smaller area e.g. 1000 properties
M
River
Mains
M
Source meter measures total output
M
M
Mains
X Closed valve
● Meter

Figure 2.2 Typical DMA design options (adapted from Farley (2001))

2.2.1 Definition of DMA size

The size of DMAs in a WDN, as a main design parameter can vary significantly depending on the sectorization purpose (e.g. larger zones are recommended for network's global water balance monitoring). Several factors will influence preferable size of the DMA such as 1) acceptable economic level of leakage, 2) demographic factors, 3) variation of elevation, 4) individual water company preference (e.g. discrimination of small bursts (service bursts) in favor of smaller installation and maintenance costs) (Farley, 2001). Acceptable economic level of leakage is defined by the individual water company operating the WDN and it directly affects the DMA’s size parameter. For the highly efficient WDNs, in which water losses are already significantly decreased with the measures taken in the past, economic level of leakage will be set to a lower value. On the other side, in the case of low efficient WDNs in which water losses are high and no measures were implemented in the past to tackle this issue, economic level of leakage will be set to a higher value. This is closely related to the geographic position and the global economics as the former (highly efficient WDNs) relates to the developed countries and the later (low efficient WDNs) to the developing and countries in transition. Low economic level of leakage will enable the definition of smaller DMA size, which in turns enables: 1) identification of bursts more quickly, 2) identification of smaller bursts and 3) maintaining the total DMA leakage at the lower level. On the other hand, using high economic level of leakage will produce smaller number of larger DMAs, that can be used to keep track of the global water balance in the network. This is suitable and recommended for low efficient WDNs as a first step towards more efficient management. Additionally, interventions required for the creation of smaller size DMAs are not easy to plan, as it is hard to foresee their influence on the whole WDN without enough measurement data and properly calibrated hydraulic model of the network (which is usually the case for WDNs in low developed countries). In terms of its demographics, each WDN is unique making it difficult to give general recommendation about the size of the DMAs. There are networks with large urban areas with high population density, and there are rural networks with scattered settlements covering larger geographic area. In all cases, elevation of the nodes within the same DMA should be in the predefined specific range (Morrison et al., 2007).

Aforementioned guidelines give some rough framework about the “manageable DMA size” in terms of number of consumers and links or network length. WAA & WRC (1985) suggest between 1000 and 3000 costumer connections within the DMA and Butler (2000) recommends 2500 – 12500 consumers or 5 – 30 km of total network length. Guideline of the World Health Organization (Farley, 2001) classifies DMAs, based on the number of costumer connections, into a) small (<1000), b) medium

(1000 – 3000) and c) large (3000 – 5000). Similar recommendations can be found in the IWA guideline (Morrison et al., 2007). It is considered that having DMAs larger than 5000 connections is not practical as it becomes difficult to distinguish leakages from the night flow data, while taking more time to allocate them.

It is clear from the discussion above that the preferable DMA size is network specific, influenced by many factors and has to be determined based on a thorough analysis of the specific data relevant to the network in consideration.

2.2.2 Designing the sectorization solution for WDN

Complexity of the real life WDN results in many different alternatives in which network sectorization into DMAs can be done. Usually, sectorization is governed by the criteria of having zones of “manageable size” in terms of number of consumers, links or network length. It can be also subjected to many other criteria and limitations leading to arbitrary solutions, usually obtained by the “trial and error” technique conducted by a local expert, familiar with all of the WDN specifics. Practical application of such approach is illustrated in Grayman et al. (2009) where two large case study networks are redesigned to 1) implement typical DMA design as guidelines provided in Baker (2007) and 2) to allow additional control and isolation of the system in order to improve water security. Acquired division into DMAs were verified using four metrics: 1) system ability to provide sufficient fire flow supply, 2) water age, 3) water security in case of accidental contamination (a – number of residents exposed to a contamination and b – total network length contaminated) and 4) system reliability measured by resilience index (Todini, 2000). Study concluded that the implementation of DMAs can significantly improve network’s water security, while preserving its other design criteria. Conclusions made in that research cannot be generalized and mapped onto other distribution networks, but can give valuable insights on the effects of sectorization.

Generally, sectorization process should be governed by general criteria in terms of zone size, but also other case specific criteria and requirements which should include evaluation of potential investments, energy consumption for pumping, increased water leakage, exceeded or insufficient pressures etc. In practice, sectorization process starts with the identification of key sectorization objectives (e.g. monitoring of water balance in network, reduction of network pressures, reduction of leakage) and design criteria, followed by the identification of performance indicators (PIs) that will be used to assess impact of implemented interventions in the network. It must be ensured that clustering interventions in the WDN, required to create sectors, do not worsen its operational performance and reliability in terms of water supply.

Different algorithms for automated decomposition of the WDN into DMAs have been presented in recent years, as well as the tools that can be used to support this process (Deuerlein, 2008; Perelman & Ostfeld, 2012). All existing algorithms for automated sectorization have three general steps: 1) Division of the network into clusters, 2) Placing the valves and flow meters on cluster's boundary pipes to create the DMAs and 3) Evaluate solution based on the adopted PIs. For the purpose of initial division of the WDN (1st Step), majority of presented methodologies rely on the Graph Theory algorithms (Alvisi, 2015; Ferrari et al., 2014; Hajebi et al., 2016), while others are using the modularity index (Giustolisi & Ridolfi, 2014b) or community structure metrics (Diao et al., 2013; Zhang et al., 2017). So far presented methods employing graph theory include only cluster (DMA) size range (min-max) and reachability from the transmission main as the sectorization governing variables. On the other hand, modularity and community structure metrics are introduced from other fields of research and are based on similarity between clusters based on the weights assigned to the links. These approaches, although able to determine DMAs, are sensitive to the selection of links weights (Diao et al., 2013) and, more importantly, do not provide clear connection to key drivers/PIs used in engineering practice.

In a real-sized WDNs a large number of possible alternatives exist for positioning the valves and flow meters in order to create the DMAs (2nd Step), many of which are not feasible as they do not meet the basic hydraulic requirements for WDN operation. For the purpose of selecting the (near) optimal alternative, decomposition algorithm is usually coupled with some type of optimization method (Hajebi et al., 2016; Zhang et al., 2017) which requires significant amount of computational time. So far, computational efficiency has been regarded as a method's secondary requirement, compared to the quality of the obtained solution, as division into DMAs is usually a onetime strategic(planning) task and there is no need to partition the WDN in real-time. Still, identifying sectorization solution in reasonable time, (minutes – compared to hours/days/weeks) would allow practicing engineers to analyze lot more different solutions, suiting to different strategies, and come up with better overall solution.

Comparison of computational efficiencies of these approaches is given here for illustration purposes. Large benchmark network BWSN2 (12527 nodes and 14831 links) has been used as a case study both in Hajebi et al. (2016) and Zhang et al. (2017). Hajebi et al. (2016) reported the running time of their algorithm to be about 15 hours on a standard PC. In the research of Zhang et al. (2017), for the same network running time was about 278 hours (approximately 11.6 days), even though the PC with newer generation processor and double the RAM memory was used, compared to the one used in Hajebi et al. (2016). It can be said that computational time for both algorithms are extremely high, from the user point of view. Later one especially, due to the fact that it uses evolutionary algorithm to solve multi-objective (MO) optimization problem, thus requiring extremely large number of hydraulic runs and objective evaluations. Former one, heuristically determines location of the valves and meters based solely on their topology, which reduces the solution search space resulting in significantly lower computational time.

In the process of developing new methods, various limitations and constraints, important for the proper functioning of the WDN, were implemented in optimization procedures. Initially, only DMA size and network pressure constraints were considered (Di Nardo & Di Natale, 2011), with each method adding additional sectorsation parameters and network's PIs to their lists of limitations and constraints. Probably the most comprehensive such list is presented in Hajebi et al. (2016), having 13 objectives and 11 constraints. It may be even said that these lists have grown too much, exhausting all practical aspects important for normal every day operation of the WDN.

From the previous discussion it can be concluded that, despite all recent advancements made, scope exist to further improve existing water network sectorization algorithms, especially in terms of usability for practicing engineers. Two main aspects in which these improvements can be made are: 1) computational efficiency of the algorithm and 2) the implementation of practical engineering principles relevant to the WDN.

Computational efficiency can be improved significantly if heuristic is used to narrow down the solution search space, instead of MO optimization algorithms, as discussed above. There are some algorithms that have already adopted this approach (Diao et al., 2013), but used heuristics are simply topology based and do not address the feasibility of the solution in terms of practical field implementation. What is meant by this is that, even though the solution may be hydraulically feasible, selection of connection pipes that will be closed and ones that will be equipped with flow measuring devices is affected by the possibility of measuring discharge on different pipe diameters.

In the following section, extensive review of available methods for sectorization of the WDN is given, highlighting their benefits and shortcomings.

2.2.3 Overview of available methods for WDN sectorization

Available methods can be generalized into two different categories, based on the required user interaction: 1) support algorithms for the definition of DMAs and 2) fully automatic algorithms. Support algorithms serve as an aid tool to the user defining the DMAs in the WDN and they require iterative user interaction during the process (e.g. to define preferable number of zones and flow meters). Fully automatic algorithms, as the name suggests, require all user input data to be supplied at the start and algorithm will come up with the best solution, according to the implemented criteria and limitations.

2.2.3.1 Support algorithms for sectorization of WDN

Deuerlein (2008) introduced new decomposition concept of the network graph according to its connectivity properties. This concept allows simplified hydraulic modeling of the network and overview of different graph specific elements (such as network-core, bridges, forest). Interpretation of these elements within WDN can derive significant information about network connectivity, water supply paths and interactions between different WDN parts. Simplification of the network in this manner can be utilized for different applications in the field water supply networks, one of them being assistance in the initial stages of sectorization process.

Perelman & Ostfeld (2012) presented another methodology that uses topological (connectivity) analysis for the purpose of better understanding of large WDN behavior and function. It relies on clustering approach, dividing the network graph into strongly and weakly connected components using graph theory algorithms. Depth First Search algorithm (DFS, (Tarjan, 1971)) is utilized to identify strongly connected components and Breadth First Search (BFS, (Pohl, 1969)) to find weakly connected ones. Resulting clusters may give a simplified representation of the network, however described clustering procedure may result in significantly size varying clusters. To address this issue, algorithm was extended to group smaller clusters. Cluster-layout of the network is time dependent. As the orientation of the pipe flow changes during the simulation time, so will the identified strongly and weakly connected clusters. As the authors discuss, presented methodology is intended to give a simplified representation of the WDN, possibly beneficial for the solution of other type of problems. Since the clustering algorithm basically gives system's connectivity change in time, it can be used for applications such as: 1) DMAs design procedure, 2) Sensor location placement problem, 3) Contamination source detection and 4) Response modelling.

Di Nardo & Di Natale (2011) presented heuristic design support methodology for sectorization of WDN into permanent DMAs. Methodology is intended to help identify position of the isolation valves and flow meters and it is based on graph theory. Algorithm starts with the analysis of minimum dissipated power paths from each source to each node in the Original Network Layout (ONL). Nodal minimum dissipated power paths are determined using Dijkstra's shortest path algorithm (Dijkstra, 1959), which requires link weights as an input according to which path search is done. Links are assigned with weights proportionate to the head losses resulting from the hydraulic simulation of the ONL. Then, pipe appearance in the paths frequencies are calculated and pipes with low frequency are regarded as "less important", as they are probably not on the main supply paths. Removing pipes with the path frequency equal to zero, Main Network Layout (MNL) is identified. At this point, it is required of user to provide preferred number of DMAs and flow meters. Isolation valves and flow meters are positioned by the algorithm based on the 2 criteria: 1) minimize number of isolation valves in the MNL as this will lead to the change of main supply paths to the nodes, thus altering the energy dissipation in the network and possibly leading to the hydraulically unfeasible solution and 2) place isolation valves on the pipes with lowest path frequencies. Identified solution is tested for satisfaction of the adopted PIs and number of flow meters. If some of the PIs are not satisfied, planner has to redefine preferred number of DMAs. Otherwise, if only the number of flow meters is higher than the preferred number, solution can be finely tuned by further removal of the flow meters and their replacement with the isolation valves, each time checking the PIs. PIs used in this research are energetic (as described in Todini (2000)), statistical and basic hydraulic (e.g. pressure deviation). Presented methodology can pose a valid technical support for the DMA planners. However, it requires significant user interaction. In addition to that, the two criteria used for initial positioning of the flow meters and valves are not convincing enough that there is not another solution with the same number of DMAs that can satisfy PIs. This methodology was also used in Di Nardo et al. (2013a) to investigate WDN partitioning effects on safety and security. The goal of the research was to simulate contamination incident and assess the benefit of contaminated DMA isolation from the rest of the network. Results showed that timely DMA isolation can decrease contaminant diffusion and protect one part of consumers from contamination.

2.2.3.2 Automatic algorithms for sectorization of WDN

Diao et al. (2013) presented new approach, based on the network community structure, to divide WDN into DMAs. Motivation for application of community structure approach comes from the fact that many complex systems, WDN being one of them, have a property of higher links density within the communities than between them. Community (or DMA in the case of WDN) detection is based on the modularity metrics presented in Clauset et al. (2004) and Newman & Girvan (2004), and used to create a dendrogram illustrating network graph decomposition into communities at all levels. Prior to this, water distribution system has to be mapped into an undirected weighted graph. Links diameters were used as weights in this research. In order to tailor the dendrogram for the application of DMAs detection, average water use per connection is calculated based on the water demand data. To identify DMAs in the network top-down search of the dendrogram is conducted. In the top-down search process, dendrogram is cut at each level and corresponding DMAs division is evaluated in term of their size. At the most top levels DMAs upper size constraint will not be satisfied, and the search will continue downwards until all DMAs satisfy that constraint. Selection of the feed lines and isolation valves for each DMA is based on the heuristic two-stage method, as it was recognized that optimization procedure involving each possible alternative would be extremely time consuming. Methodology was tested on the large BWSN2 network and the results were compared to the manual methodology presented by Grayman et al. (2009). Resulting DMA division proved to be almost identical to the solution obtained by Grayman et al. (2009). Running time of the algorithm for the tested network was about 20 min, proving it to be viable alternative to the manual-expert method requiring more time and engineering experience. Although this is one of the first fully automatic algorithms, a note has been made about significantly different results when different weights are used for network links, expert knowledge about the studied system is still required.

Di Nardo et al. (2013b) presented an automated tool for smart water network partitioning based on graph partitioning method and Genetic Algorithm (GA) optimization. Partitioning of the network is done based on the Multi-Level Recursive Bisection (MLRB) algorithm as implemented in METIS software (Karypis & Kumar, 1998). It divides network graph, based on the nodal and link weights, into a number of desired partitions following criteria of: 1) minimizing sum of partitions' interconnecting links weights and 2) obtaining partitions with the same sum of nodal weights within them. Different network properties can be used as weights (e.g. links-diameters, pipe flow, dissipated power, nodes-water demands), and it is recommended that different weights are investigated as they can affect the result of the algorithm. GA is used to find the best position of the isolation valves and flow meters, by minimization of dissipated power in the network. Results are reported for one relatively simply and small network, for which partitioning is done into five DMAs, and it is not clear how would the algorithm cope with real-sized large networks. Full automatization of the partitioning process was indicated as main advantage of this algorithm compared to the methodology presented by the same authors (Di Nardo & Di Natale, 2011), which required some user interaction during the process itself.

Another method that tries to surpass the trial and error approach is presented by Di Nardo et al. (2014), which is essentially an extension of the research presented in Di Nardo & Di Natale (2011). Methodology is focused on the identification of isolated DMAs (i-DMAs) rather than the standard DMAs. Isolated DMAs are defined as parts of the WDN that are fed from its own source (or sources) and are completely isolated from the rest of the network. Benefit of having i-DMAs is better pressure control within the zones, since they are not affected by the other sources in the network (such as the case for ordinary DMAs). Algorithm makes use of the graph theory to define hierarchical ordering of the network graph, starting from each source node. Hierarchical representation of the graph used here is similar as the dendrogram representation used in Diao et al.(2013). Independent sectors in the network are identified next, and i-DMAs are defined using heuristic approach for isolation valves positioning, based on hydraulic simulation results and GA optimization. In GA optimization objective function, being the sum of dissipated power in the network, is minimized. Finally, PIs are calculated for partitioned network and compared to the ones calculated for the original network layout. Results presented in this paper were compared to the other sectorization solutions obtained by the manual approach presented in Tzatchkov et al. (2006), and proved to be better in terms of post-sectorization PIs. Presented methodology is intended for identification of i-DMAs, but it was highlighted that the ordinary DMAs can be easily derived from the i-DMAs solution just by leaving some pipes between the i-DMAs open. This is true, but there is one limitation of this approach regarding the preferred DMA size. Generally, this approach will yield coarse division of the network, as it is influenced by the number of available sources and network size, and it may not be suitable for every case study (design requirement).

Ferrari et al. (2014) presented another graph theoretic based approach for the design of DMAs. Methodology presented in this research incorporated additional important factors for the design of DMAs, other than just the DMA size used in most previous papers, such as DMA's connectivity to the main transmission system, flow exchange between adjacent DMAs and satisfaction of minimum pressure requirements. Recursive bisection algorithm is used for identification of desired number of DMAs and the definition of their boundaries (valves and flow meters). Algorithm is tailored for defining DMAs that are not allowed to exchange flows, thus larger number of pipes that should be closed will occur. Stochastic component is implemented in the method in order to yield different solution with each algorithm run, allowing user to obtain different feasible alternatives for the same input parameters. However, results reported only one solution for the case study network. In the process of defining the DMAs methodology ignores groups of nodes having the total water demand lower than the predefined lower DMA size limit. This implies that those parts of the network are not intended for flow monitoring and thus represents difficulty for identification of water losses in the network. In the large network used as a research case study total demand in such disqualified areas was about 10% of the total network demand. Following on this research, Savić & Ferrari (2014) reported number of feasible solutions (116 to be exact), based on methodology presented in Ferrari et al. (2014), and compared them with the manual approach solution presented by Grayman et al. (2009). Comparison is made in terms of three PIs: 1) number of closed pipes (as a solution cost representative), 2) water age and 3) Resilience index (Todini, 2000). Solution cost used in this study is relatively descriptive, as it doesn't take into account the variation of the valve price with the diameter, or the price of the flow meters that should be installed. Results suggested that partitioning of the network into DMAs does cause minor decrease of the WDN's performance, which is however irrelevant compared to the benefits (e.g. reduction of leakage and better pressure control). Aforementioned benefits were not investigated or quantified in this research.

Alvisi & Franchini (2014) presented a three step modular algorithm for automatic creation of DMAs. In first step graph theory BFS algorithm is employed to define broad set of possible solutions using DMA size, in terms of total water demand, as the only design criterion. In the second step another graph theory algorithm, Dijkstra's shortest path algorithm, is used to narrow down the broad set of solutions defined in the previous step. Pipe resistance is used as weight in the Dijkstra's algorithm, as a measure of pipe conductance. In the final step, each solution from the narrowed down set is hydraulically analyzed and resilience index is used as PI to prove its feasibility. Presented methodology was applied to relatively small case study network (465 links and 413 nodes), resulting in solution with three DMAs, and PIs were comparable to the ones obtained with the chosen reference method of Di Nardo et al. (2011). Algorithm computational time is reasonable at first sight (50 min), but it remains unclear how it would deal with large networks containing several thousand links and nodes.

Giustolisi & Ridolfi (2014b) introduced modularity metrics for the purpose of WDN segmentization, based on the original definition of Newman (2004). Classic modularity definition was tailored to account for the WDN specific characteristics, yielding new Modularity-based index. New formulation allows division of the network into sectors that have similar internal pipe attributes (e.g. diameters, head losses), opposite to the original definition which is more suitable to division into sectors that are similar to each other. Additionally, actual position of the installed device (valve or flow meter) is accounted for. Network modularity-based index and solution cost are used as objective functions in MO optimization to find the optimal solution. Research showed that modularity-based metrics can be successfully used for rough estimate of potential DMAs and its boundaries. However, it did not address the actual selection of devices to be installed on the boundary edges (isolation valves and flow meters) and how it would reflect on the performance and hydraulic capacity of the WDN. This research was extended in Giustolisi & Ridolfi (2014a) with the introduction of new, infrastructure modularity-based index. As stated by the authors, WDN-tailored modularity metrics presented in Giustolisi & Ridolfi (2014b) suffers from resolution limit that increases with network size. This means that definition of small DMAs is not possible and methodology is suitable only for general planning, as stated previously.

Alvisi (2015) presented procedure based on MLRB graph partitioning algorithm, the same one used in Di Nardo et al. (2013). Novelty of the proposed method is that it couples the tasks of network partitioning and positioning of the flow meters and isolation valves, opposed to other methods that treat these tasks separately. It is hypothesized that this approach would allow finding better near optimal sectorization solution. Optimization process is performed using SCE-UA algorithm (Duan & Gupta, 1992) maximizing systems post-sectorization resilience. Reported results suggest that higher values of minimal pressures, and consequently higher values of resilience index, are achieved when compared to methods of Di Nardo et al. (2011) and Alvisi & Franchini (2014). Only one relatively small network is used as a case study (391 pipes and 273 nodes), considering its division into 3 DMAs, without reports regarding computational time. Applicability of the algorithm for real-sized networks hence remains uncertain.

Ferrari & Savic (2015) investigated economic benefits of sectorization, expressing them with three PIs: water leakage reduction, burst frequency reduction and water sensitive demand reduction. Large BWSN2 network (Ostfeld et al., 2008) is used as a case study and 73 feasible sectorization solutions were identified using algorithm presented in Ferrari et al. (2014). Results show overall declining trend for all PIs with increased number of DMAs. Consequently, this requires higher number of flow meters and valves. Without having all data about the cost of water, implementation and maintenance of the DMAs, net economic benefit of different solutions was not reported.

Hajebi et al. (2016) coupled network partitioning algorithm, named WDN-Partition, with many-objective optimization to perform network sectorization. WDN-Partition method uses structural graph partitioning technique to divide network into group of nodes referred to as islands. Distinction is made between minor and major islands, latter ones' being subjected to many-objective optimization technique based on enumeration in order to determine the locations of flow meters and valves. Key advantage of this methodology compared to others is the ability to provide a set of feasible solutions, rather than a single one. One of the claimed strong points of the algorithm is a comprehensive list of objectives used in optimization procedure (14 in total). However, for the case study tested only three of them were used. Employing optimization for positioning DMA isolation devices, this method suffers from high computational burdens for real sized networks, as discussed in previous section (2.2.2).

Laucelli et al. (2016) presented a two-step strategy for optimal sectorization, aimed specifically for reduction of leakages in the network. In the first step network partitioning is done based on the WDN tailored modularity index presented by Giustolisi & Ridolfi (2014b). Optimization in this step involves two objectives: minimization of connecting links and maximization of modularity index. In the second step optimization procedure targets minimization of number of flow meters, minimization of unsupplied nodal demands and minimization of background leakages. Dealing with leakage assessment, the use of pressure-driven hydraulic model is necessary, and so far this is the only research adopting such model in the methodology for sectorization. Reported results are encouraging, however heavily use of optimization and the lack of large case study investigation pose a question on methods applicability for real-sized networks.

Scarpa et al. (2016) presented hierarchical sectorization methodology based on progressive union of initially identified elementary DMAs (eDMAs). Process of joining the identified eDMAs is driven by maximization of resilience criterion and satisfaction of DMA size constraints. This can be regarded as bottom-up sectorization approach, since initially identified small eDMAs are aggregated into large ones. Even though this methodology is convenient for sectorization of WDN in phases, a top-down sectorization approach would be closer to engineering perception and more in accordance with approach generally taken in practice by water companies. This means that large DMAs should be setup at first (e.g. for tracking of network's water balance) and then partitioned into smaller DMAs per future requirements.

Ciaponi et al. (2016) presented yet another methodology relying on modularity index metrics to perform initial partitioning of the network. Iterative heuristic method is used to determine which pipes will be closed and which one equipped with flow meters. Partitioning of the network based on modularity is highly dependent on weights assigned to nodes. Even though the authors recognize this, presented results are based purely on topological partitioning (e.g. all links have weights equal to 1, meaning that nodal weights are equal to nodal degree in graph), lacking investigation of alternative weights relevant to WDN sectorization (e.g. nodal vertical position or pressure).

Algorithm of Zhang et al. (2017) also employs modularity metrics to partition the WDN. Links are assigned with weights calculated as average pressure head of adjacent nodes, resulting from hydraulic simulation. In this manner, nodes within the clusters will have similar pressures. This is the improvement compared to the method of Ciaponi et al. (2016) which uses only topological weights. BORG algorithm (Hadka & Reed, 2013) is used for determination of DMA boundaries, without any considerations to reduce solution search space prior to the optimization itself. Consequently, algorithm takes 278 hours to complete the analysis on the large case study network, making it highly computationally inefficient.

Chronological review of sectorization algorithms presented above shows that over the time methods become more and more complex. Optimization methods are computationally expensive by their nature, and the addition of new objective functions by each sectorization method only highlights this effect. Performing extended period hydraulic simulations assuming the pressure-driven analysis (instead of demand-driven) has further negative influence on the computational efficiency of a method. Solution search space exponentially increases with the complexity of a network, and perhaps this is why recently presented methods employing optimization are lacking results supporting their application on real-sized networks.

Point made in the previous paragraph is best reflected in the paper of Salomons et al. (2017). Case study presented was a part of “Battle of water networks DMAs” contest prepared for WDSA2016 international conference. Problem required redesign of real water distribution network in Colombia. The aim was to repartition the network into manageable DMAs complying to imposed goals (e.g. improvement of water quality) and limitations (e.g. anticipated future demands and seasonal production capabilities). In total there were 8 equally weighted objectives for the problem. Any network interventions were allowed, such as adding and removal of the pipes, installation of valves, managing the tank volumes and pumping stations operating rules etc. Problem solution presented in Salomons et al. (2017) was the only one obtained using multi-stage engineering approach (i.e. “trial-and-error” approach) and won the competition. Other participants that reached for various types of automated procedures and optimization algorithms failed to deliver satisfactory solutions. This is due to the fact that, as discussed, such algorithms reported in the literature implemented only a few objectives into consideration. Real problem intrinsic as this one, and with so many objectives, cannot be solved with any fully automatic algorithm available at this point. This points out the importance of engineering reasoning in the WDN sectorization process, that cannot be replaced solely with utilization of optimization. Obviously, the goal is to find the balance between the engineering judgement and available state-of-the-art scientific tools.

Better computational time of sectorization procedures utilizing optimization can be achieved by improving the computational efficiency of the hydraulic solver used to perform multiple hydraulic calculations. Review on available methods for hydraulic simulation of WDN is presented in the following sections.

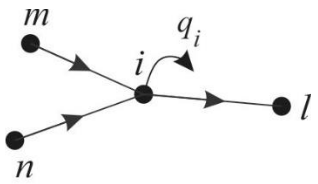
2.3 HYDRAULIC SIMULATION OF WATER DISTRIBUTION NETWORK

2.3.1 Basic equations describing the WDN

Two basic conservation principles describe distribution of pipe flows (*Q*) and nodal heads (*H*) in the WDN, which is specific to the given nodal demands (*q*). To satisfy conservation of mass principle, for each node in the network (*i*), following relation, also known as continuity equation, has to stand:

$$\sum_{j}^{n_i} Q_{ij} + q_i = 0 \tag{2.1}$$

where Q_{ij} is pipe flow in the pipe connecting *i* and *j* nodes in the network, q_i is the nodal demand of the node *i* and n_i is the number of pipes coinciding in the node *i*. Sign convention adopted here is that the inflows in the node *i* are negative and outflows are positive. Simple illustration of 3 pipes coinciding in one node is used to describe application of continuity equation (Figure 2.3).



$$-Q_{mi} - Q_{ni} + Q_{il} + q_i = 0$$

Figure 2.3 Application of continuity equation

Second conservation principle is conservation of energy defined by Bernoulli’s principle:

$$H_i - H_j - f\left(Q_{ij}\right) = 0 \tag{2.2}$$

Where H_i and H_j are the heads at the end nodes of the pipe and $f\left(Q_{ij}\right)$ is head loss across the pipe, resulting from the friction, which is a function of pipe flow Q_{ij}. A power function of flow is usually used to calculate head loss across the pipe:

$$f\left(Q_{ij}\right) = R_{ij} Q_{ij} \left|Q_{ij}\right|^{n-1} \tag{2.3}$$

Where R_{ij} is the coefficient which encompasses different characteristics of the pipe (e.g. diameter and length), hence further on it will be referred to as the pipe characteristic. Value of the term *n* in the exponent depends on whether the Darcy-Weisbach (DW) or Hazen-Williams (HW) equation is used to describe head loss due to the friction. In the case when DW equation is used, the term *n* takes the value of 2.0 and 1.852 in the case of HW. Substituting equation (2.3) in the head loss equation (2.2) yields its nonlinear form:

$$H_i - H_j - R_{ij} Q_{ij} \left| Q_{ij} \right|^{n-1} = 0 \tag{2.4}$$

Equations (2.1) and (2.4) present elementary equations describing the flow and pressure distribution in the WDN under steady-state conditions. Derivations presented here are based on the assumption that only pipes are present as the link elements in the WDN. In a real WDN, other link elements such as pumps, valves and localized losses have to be accounted for in the equation (2.4).

It should be noted that in the case of branched network (i.e. network without loops), flow and head distribution can easily be obtained with two propagations through the network (Figure 2.4). First, backward propagation is done and flow distribution is determined simply by applying the continuity equation at each node starting from the most downstream ones. Afterwards, forward propagation starts from the node with known head (e.g. reservoir) and all nodal heads can be calculated as per equation (2.4), since flow distribution is already defined.

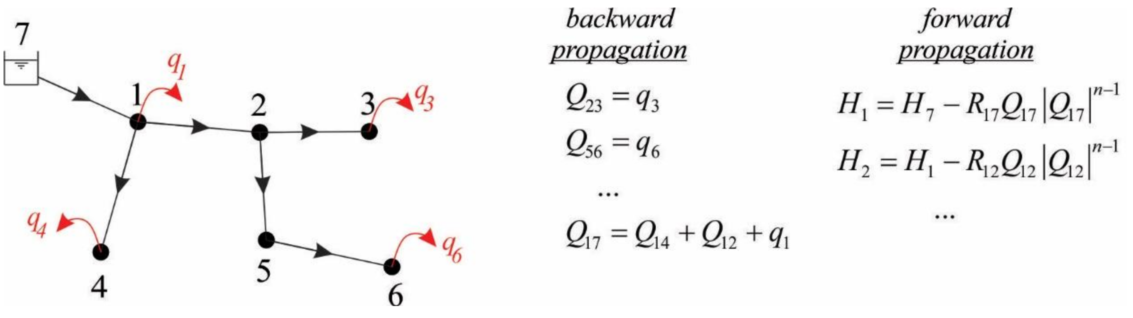

7
1
2
3
4
5
6
q_1
q_3
q_4
q_6
backward propagation
$Q_{23} = q_3$
$Q_{56} = q_6$
...
$Q_{17} = Q_{14} + Q_{12} + q_1$
forward propagation
$H_1 = H_7 - R_{17} Q_{17} \left| Q_{17} \right|^{n-1}$
$H_2 = H_1 - R_{12} Q_{12} \left| Q_{12} \right|^{n-1}$
...

Figure 2.4 Solving hydraulics for branched network

For looped networks, writing equation (2.1) for each node in the network and equation (2.4) for each link will form a mixed system of linear and nonlinear equations which has to be solved for unknown pipe flows (Q) and nodal heads (H). Formulation of the system will define the number of equations, depending on which unknown is chosen as the primary one. There has been numerous methods and algorithms developed for the purpose of solving this system, as it will be discussed in the next section.

2.3.2 Overview of available methods for hydraulic simulation

In the past, many different methods and algorithms have been developed for the purpose of solving the flow and pressure distribution problem in the network, represented with the nonlinear system of equations (2.1) and (2.4). All of them are well documented in the literature. Brief and interesting historical overview of methods was made by (Ormsbee, 2008) where they are divided in three periods: 1) Pre-computer period, 2) The dawn of computer age and 3) The age of advanced methods. In this section, complete overview of available methods developed during these three periods will be given in the historical overview section, followed by their systematization. Overview of recently presented algorithms, which are essentially based on already available methods and focused on their improvement, will be given in separate section.

2.3.2.1 Historical overview

The problem of water distribution system analysis was systematized for the first time by Hardy Cross (Cross, 1936) in his publication "Analysis of Flow in Networks of Conduits or Conductors" published in University of Illinois Bulletin. Based on this work, over the following years many different algorithms and methods have evolved, reaching to a point that American Water Works Association's (AWWA) committee on distribution systems reported: "*Literally dozens of technical papers have been published over the last few years dealing with mathematical aspects of distribution system simulation, seemingly approaching a point of saturation*" (Walsky, 1983). Prior to giving the historical overview and development of all available methods for hydraulic analysis of the WDN, brief description of the original work of Hardy Cross will be given.

Cross (1936) proposed two different methods for the solution of the network hydraulic analysis problem. First method is named "Method of Balancing Heads" and the second one "Method of Balancing Flows". Names given to these methods in essence describe the approaches used to solve system of equations describing the problem. In the former one, pipe flows always satisfy continuity equations for nodes, but they are iteratively corrected to balance the heads on nodes and satisfy the condition of zero change of total head around each closed circuit (loop) in the network. This is achieved by calculation of loop flow corrections for each loop, that are used to correct the flows in the pipes forming the loops. In the later method, condition of zero change of total head around each closed circuit always stands, and the pipe flows in the loop are iteratively adjusted until continuity equations in nodes are satisfied. For the implementation of both methods some initial assumptions for the variable values have to be made. Initial pipe flows, satisfying continuity equation, are assumed for the Method of balancing heads, and initial nodal heads for the Method of balancing flows. Comparing the two approaches, advantage is given to the Method of balancing heads (later also known as the ΔQ/ loop method) stating that its convergence is “for practical purposes sufficiently rapid”, while for the Method of balancing flows convergence is “slow and not very satisfactory” due to difficulty of guessing good initial nodal heads. This, combined with the fact that loop method was more “natural” in its application and more acceptable for hand on calculation, resulted in its wide acceptance in engineering practice. With the dawn of the computer era, Cross’ loop method was being implemented in computer programs (Adams, 1961; Graves & Branscome, 1958; Hoag & Weinberg, 1957), allowing it to be used for larger and more complicated networks that could not be solved efficiently by hand calculation. However, problems of solvability and convergence for larger networks, caused by different flow conditions (e.g. large diameter pipes or small flow rates), were reported (Dillingham, 1967). Cao (1963) pointed out the problem of non-uniqueness of identified loops, as there can be more than one closed paths between any two nodes in the network. He proved that the inadequate identification of loops can lead to slower convergence or even divergence of the solution. Additionally, original method presented by Cross included only pipes, without the discussion about other types of links such as pumps and valves.

These problems gave the incentive to many researches in the following years to search for the more efficient ways to implement Cross’ methods and benefit from the computer power that has become available. Martin & Peters (1963) were the first to investigate the approach in which node equations, describing the Method of Balancing Flows, were solved simultaneously for all nodes in the network, rather than one by one as in original work by Cross. They reported no issues on the convergence, but simultaneous solution of the system meant the increase of required computer memory, as the coefficient matrix of size [N_n ,N_n] is needed for the solution of the system. For finding the solution, Newton Raphson (NR) iterative method was employed. Shamir & Howard (1968) showed that the same modelling method can be used to accommodate different types of links such as pumps and valves, but also demonstrated the possibility to solve the network problem for different type of unknowns, other than heads (e.g. pipe resistance or nodal demands, which was useful for calibration purposes). Epp & Fowler (1970) applied the approach of Martin & Peters (1963) to simultaneously solve equations, only this time they used it for the original Method of balancing heads ($\triangle Q$ method/loop method) based on solving the loop equations. They presented an efficient algorithm that had some significant innovations at the time, such as: automatic method for reducing the storage requirements and automatic method for determining the initial flows in the network that will enable fast convergence to the final solution. Hamam & Brameller (1971) developed so called hybrid method, which is intended to combine the advantages of both approaches for system analysis – nodal approach and loop approach. In general, it is easier to formulate the nodal approach as it will result in solution matrix with maximum sparsity. On the other hand, loop approach provides better convergence. Osiadacz (1988) compared the hybrid method of Hamam & Brameller (1971) with the simultaneous solution for loop method (as described in Epp & Fowler (1970)). Comparison was done on the examples of gas networks and it was concluded that loop method is more suitable for larger networks (with thousands of pipes and loops), given that an efficient algorithm for identification of loops is used.

All methods mentioned so far apply NR method to linearize and solve nonlinear system. To achieve convergence, this type of linearization requires reasonably assumed initial solution (Liu, 1969; Martin & Peters, 1963; Shamir & Howard, 1968). To address this issue, Lemieux (1972) presented efficient algorithm based on the combination of modified NR method and specific Gaussian elimination to provide fast convergence that is independent of the starting assumption. In this work, solution is found with respect to the nodal heads. Kesavan & Chandrashekar (1972) presented method based on the concepts from linear graph theory. Utilization of both head loss equations for loops and continuity equations for nodes simultaneously was highlighted as main advantage of this approach. They used terms from the graph theory, such as tree and co-tree, to define matrices that are used to formulate the system of equations to be solved. System is solved for the unknown co-tree flow vector (vector of loop flow corrections). Comparison is made with the original Cross method, although only on one example network, and it was found that method of Kesavan & Chandrashekar requires 3.5 times less iterations to converge and it is 2 times faster. Method also converged when the initial assumption was not good and original Cross method failed to reach the final solution. Wood & Charles (1972) presented yet another approach, based on the use of linear theory to solve network hydraulics. System is solved for the unknown pipe flows. Rapid and assured convergence and no need for guessing the initial distribution of flows are highlighted as main advantages of the proposed methodology. Collins et al. (1978) presented, what they called, "*revolutionary new approach*" using optimization technique to solve the problem of network hydraulics. They introduced two models, "*content*" and "*co-content*, both defined as a nonlinear functional of variable for which the problem is solved. Naming convention for the models comes from the works on nonlinear systems of Cherry (1951) and Millar (1951). The "*content*" model is defined as a functional of pipe flows and the "*co-content*" model is defined as a functional of the nodal head values. In the former model the goal is to find the set of flows which satisfy flow conservation and minimize the system content, and for the later one to find the set of head losses that will sum to zero around each loop in the network and minimize the system's co-content. For the minimization of nonlinear functional (objective function), three different nonlinear algorithms have been tested: Frank-Wolfe method, piece-wise linear approximation and the convex simplex method. Later two methods proved to be dominant over the Frank-Wolfe method, piece-wise linear approximation exhibiting the best behavior. Advantage of this approach is that combination of objective function's convexity and linear constraints guarantees the existence of unique solution, while its disadvantage is the need for an efficient nonlinear algorithm. To overcome this problem, Gradient Algorithm, originally developed by Todini (1979) was presented by Todini & Pilati (1987). Presented methodology is regarded as a bridge between optimization and NR techniques. To prove the existence and uniqueness of solution, minimization of the "*content*" model (Collins et al., 1978) is done first. Afterwards, the NR linearization method is applied on this space of flows and nodal heads resulting in recursive solution of linear system. Compared to other methods, difference is that the system is solved by inversion of the coefficient matrix. Solution size is equal to the number of nodes (N_n nodal heads) with the addition of scalar projection and linear combination of the obtained nodal heads, whose size is equal to the number of links (N_l link flows). In the work of Todini & Pilati (1987), proposed gradient algorithm was compared to Linear Theory and loop method approaches to solve the network hydraulics problem. It was noted that all three methods have the similar convergence rate. To access benefits of the different approaches, methods have been ranked (from 1 to 3) in four different categories: simplicity of input, demand for initial solution, size of the system of linear equations to be solved and efficiency of the solution. Based on this ranking it was concluded that even though on first glance it may seem that loop method would be the most appropriate one, due to the smallest size of the system, gradient algorithm's solution is the most efficient one and it also benefits from the fact that it does not require identification of the loops. Linear theory approach ranked last in this investigation. In the scientific community work of Todini & Pilati (1987) is considered as the key research in the field of steady state WDN hydraulic analysis and symbolically marks the end of the age of advanced methods, as discussed in the introductory part of this section. In the following time period, computational power increased rapidly and newly presented methods and algorithms are essentially variations and upgrades of already available methods summarized in the former discussion.

Gradient algorithm of Todini & Pilati (1987) was adopted in Environmental Protection Agency's (EPA) software for extended period simulations of hydraulic and water quality in water distribution networks – EPANET (Rossman, 2000). From the beginning, EPANET was made freely available as an open source package resulting in its wide acceptance in engineering and scientific communities. Consequently, in the years to come this lead to comprehensive testing and constant improvements of the gradient algorithm to include various upgrades done by many researchers (e.g. pressure-driven analysis and efficiency improvements). Gradient algorithm later became known as the Global Gradient Algorithm – GGA (Todini, 2006). EPANET's source code availability and computational robustness resulted in its implementation in many commercially available WDN analysis packages.

2.3.2.2 Systematization of available methods

The elementary nonlinear system of equations given with equations (2.1) and (2.4) is determined by the two sets of unknowns: set of flows in each link and the set of total pressure heads at each node of the network. For an arbitrary network, made of N_l links and N_n nodes, there are N_l+N_n unknowns in total for which system has to be solved. As there are two sets of unknowns, related to each other, a choice of primary unknown has to be made (i.e. flow or head). Since the system is nonlinear, and its direct solution is not possible, some type of linearization has to be employed (mainly NR method or Linear Theory approach, except for the method of Collins et al. (1978) where optimization approach is used). Only two methods are known to use the Linear Theory approach to solve the system (Isaacs & Mills, 1980; Wood & Charles, 1972), while other methods use the Newton-Raphson method for linearization. Having said that, main systematization of available methods can be made based on the selected primary unknown for which system is solved for, leading to a different solution formulation. Additionally, based on the approach in which equations are solved, methods can be classified as local (equations are solved one by one) or simultaneous (all equations are solved simultaneously). However, only the two methods originally presented by Cross (1936) are local approach methods, while all others fall under the simultaneous approach category. Probably the most comprehensive classification of the available algorithms is presented in Todini & Rossman (2013).

Based on the primary unknown for which system is solved, four different system formulations can be derived:

1. Loop equations system formulation,
2. Pipe flows system formulation,
3. Nodal heads system formulation,
4. Loop-node system formulation.

All these formulations will be discussed in the following text.

Loop equations system formulation

Loop equations representation is originally introduced in the work of Cross (1936) as the "Method of Balancing Heads", as discussed in the Section 2.3.2.1. This formulation can be found in the literature under various synonyms such as loop-flow algorithm (Epp & Fowler, 1970), circuit equations (Kesavan & Chandrashekar, 1972), loop equation (Arsene et al. 2004), loop method (Alvarruiz & Vidal, 2015) and ΔQ method (Ivetić et al. 2016) to name a few. Primary unknown for this formulation are flow corrections that are introduced in each loop in the network, in a manner to satisfy continuity equations in the nodes. Thus, the number of unknowns corresponds to the number of loops in the network (N_L), making this formulation one with the smallest set of equations to solve. After calculation of flow corrections, flow distribution in the network is determined and pressure head distribution is obtained applying the head loss equation, starting from the node with known head.

Pipe flows system formulation

Single formulation of this type is the one presented by Wood & Charles (1972). In this formulation, nonlinear loop head loss equations are transformed into linear equations using approximate flow rate in pipes. In combination with continuity equations, which are linear, this yields the system of N_l linear equations to be solved for unknown pipe flows. Again, when the pipe flows are determined, nodal pressure heads can be easily obtained using the head loss equation.

Nodal heads system formulation

Nodal heads system formulation is obtained by expressing the flow rate in each link of the network in terms of the nodal heads edging the link in consideration. Substitution of flow rates, expressed in this manner, into the continuity equations for nodes will yield the nonlinear system of N_n equations that has to be solved for unknown nodal heads. This formulation is used by Martin & Peters (1963) and Shamir & Howard (1968), both using the NR method to linearize and solve the nonlinear system. After finding the solution, calculated nodal heads can be used to determine the flow rates in the links, as per initial formulation of flow rates.

GGA algorithm of Todini & Pilati (1987) can be classified into this group of methods as well, with the key difference that the nodal heads and pipe flows are calculated simultaneously. NR linearization technique is used here as well, yielding the system of N_n+N_l linear equations. System is solved iteratively, in such manner that first N_n independent linear equations are solved for nodal heads and afterwards, remaining N_l equations, which are linear combination of the calculated nodal heads, are solved for unknown pipe flows.

Loop-node system formulation

Loop-node formulation is also known as hybrid formulation (Hamam & Brameller, 1971; Osiadacz, 1988), as system is solved for unknown loop flow corrections and unknown nodal heads. Loop equations introduce the conservation of energy principle, while node equations incorporate the conservation of mass principle. In this case, system that has to be solved has the size of N_L+N_n equations. Similar to the GGA formulation, set of N_n nodal equations is solved first to calculate the nodal pressures, followed by the solution of the N_L loop equations to calculate loop flow corrections. This procedure is repeated iteratively until target accuracy for the loop flow corrections is obtained.

Systematization of different approaches to the solution of the network flow and pressure distribution is illustrated in the Figure 2.5, highlighting the researchers that introduced each approach for the first time. Variables on which linearization is based are given in the parenthesis.

Out of aforementioned four system formulations, loop equations and nodal heads formulations are the two prevailing in the practice. When these two formulations are compared, it is clear that the main system matrix is smaller in the case of loop formulation (N_L equations) than in the case of nodal heads formulation (N_n equations). Thus, it would be excepted that the loop based formulation would be preferred over the node based. However, this is not the case as in the late 80s and 90s available computer power increased drastically and solving increased number of equations (i.e. the nodal heads formulation) was not much of an issue anymore. Additional requirement of preprocessing tasks in the case of loop formulation, such as identification of network loops, posed another difficulty for its wider success.

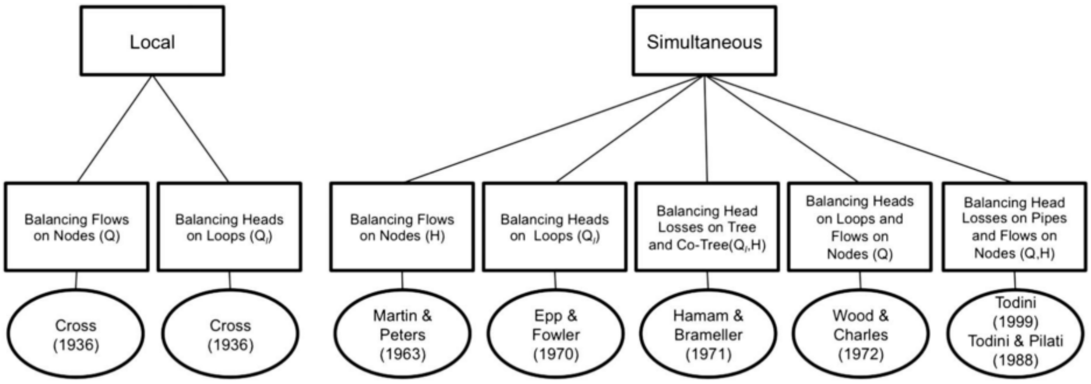

Local
Simultaneous
Balancing Flows on Nodes (Q)
Balancing Heads on Loops (Q$_l$)
Balancing Flows on Nodes (H)
Balancing Heads on Loops (Q$_l$)
Balancing Head Losses on Tree and Co-Tree(Q$_l$,H)
Balancing Heads on Loops and Flows on Nodes (Q)
Balancing Head Losses on Pipes and Flows on Nodes (Q,H)
Cross (1936)
Cross (1936)
Martin & Peters (1963)
Epp & Fowler (1970)
Hamam & Brameller (1971)
Wood & Charles (1972)
Todini (1999) Todini & Pilati (1988)

Figure 2.5 Systematization of different solution methods for network hydraulics problem (adapted from Todini & Rossman (2013))

2.3.2.3 Improvements made in recent years

In recent years researchers presented different methodologies in an attempt to further improve the WDN analysis. Many of the newly presented methods are based on modifications of Todini & Pilati’s GGA method, due to its wide acceptance and success achieved through its implementation in the EPANET software, as discussed in the concluding paragraph of the historical overview section. On the other hand, some researchers revisited other approaches to solve network hydraulics (mostly loop-flow method), for years being left in the shadow of the GGA’s success. In the following text most significant of these researches will be mentioned.

GGA based methods

Simpson & Elhay (2011) presented corrections to the Jacobian matrix formulas used in the GGA method to fully account for the dependence of friction factor on flow, when DW head loss formula is used. The result was preservation of the natural quadratic convergence of the NR method, which is not the case in the original DW head loss formula implementation in the GGA method (where linear convergence rate is achieved).

Elhay et al. (2014) presented the reformulated co-tree method (RCTM), based on the co-tree method (CTM) originally presented by Rahal (1995). In the original CTM method, modifications to the original network are made by introducing the pseudo links connecting each network node with the main source. Then, network spanning tree is determined and a global matrix, corresponding to a certain cut set made of links that are not part of the spanning tree (co-tree set), is obtained and solved using Newton's method. To employ the method, initial set of co-tree flows is needed. In essence, the CTM method is similar to the simultaneous loop-flow method of Epp & Fowler (1970), solving the system matrix of the same size, with the key difference that it does not impose any constraints on the choice of initial set of co-tree flows (such as satisfaction of continuity equation in nodes). The CTM method however did not find much success in practice probably due to complicated steps in its application and global acceptance of the GGA. The RCTM method of Elhay et al. (2014) overcome somewhat complicated implementation of the original CTM by manipulation of the network's incidence matrix, to increase its efficiency and make it competitive with the GGA. Comparison is made with the implementation of the GGA presented by Simpson & Elhay (2011) on eight case studies of different sizes. It is shown that memory storage requirements for solution matrix are reduced drastically when RCTM is used, resulting in significant speedups in calculation (between 15 and 82% for case studies). However, it was noted that the presented results are illustrative for networks with unchanged topology.

Simpson et al. (2014) introduced forest-core partitioning algorithm (FCPA) for speeding up the WDN analysis. This algorithm separates forest (linear) from the looped core part (nonlinear) of the network to enable network solution by appropriate (linear/nonlinear) method. For the solution of the looped part of the network, GGA algorithm is used. Testing of the method was done on the same eight networks used in Elhay et al. (2014) ranging from 932 to 19647 pipes. Employing the FCPA method resulted in time savings between 11 and 31%, when compared to the GGA. Additional benefit reported is avoidance of dealing with zero flows in forest part of the network when HW head loss equation is used, which has to be done in the original GGA formulation.

This investigation was extended in Deuerlein et al. (2016), in which fast graph matrix partitioning algorithm (GMPA) is presented. GMPA improved the FCPA by further separation of linear and nonlinear part of the problem within the network core. Results demonstrated further reduction of the core dimension achieved in aforementioned eight case study networks (5-55%). Approach presented in these two papers is obviously coming from the recognition that real size networks are significantly non-homogeneous in their topology (i.e. there are many tree-like parts).

Loop-flow based methods and comparison with the GGA

Arsene et al. (2004) presented a simulation scheme for on-line monitoring of water networks based on solving the loop equations. Simulation scheme is intended to act as a decision support tool for operational engineers in real-time. Scheme is made out of the 4 modules, first of them being so called co-tree flow simulator essentially based on the CTM method of Rahal (1995). This study proved that hydraulic simulation based on the loop-flow equations can be successfully used for real-time network simulations.

In the paper of Todini & Rossman (2013) different Newton-Raphson (NR) algorithms for solving the steady state WDN hydraulics were compared, giving the advantage to the NR-GA algorithm (GGA) over the NR-LF (Loop Flow) for the following reasons: a) there is no need for definition of network loops, and b) even though the NR-LF have smaller matrix than NR-GA its density is dependent of the choice of network loops which leads to possible higher computational time when using the sparse matrix solvers. However, the above statement was made based on testing on the simple network with only 3 loops, for which computational time was not reported (only the number of iterations).

A more comprehensive comparison of NR-GA and NR-LF algorithms was made in Creaco & Franchini (2014). Comparison is made in terms of computational speed which is based on 16 generic networks made of quadratic and hexagonal loops, as well as one real network of Ferrara. Both algorithms were implemented in matrix form inside of MATLAB 2011b environment. It was concluded that NR-LF algorithm has slightly better performance than the NR-GA in all cases, with this advantage decreasing with increased network topology complexity. Testing is done mainly on the extremely looped generic examples, which favor the NR-GA, and without network specific devices (e.g. valves or pumps).

Alvarruiz & Vidal (2015) presented the research regarding the efficiency improvements of the loop method. This research targeted to overcome the main disadvantage of the loop flow method, as discussed by Todini (2006) and Todini & Rossman (2013) – need for efficient algorithm for identification of the loops. The loop method was implemented in the EPANET's source code using the C programming language to enable fair comparison with EPANET's GGA implementation. Computational time was reported for 3 cases, with networks of different complexities containing up to 4 pumps and non-control valves. It was reported that the 'linsolve' routine, which solves the system of hydraulic equations, is up to 5 times faster for the loop method algorithm than the GGA algorithm in a single iteration. However, the overall speedup factors for the entire simulations were reduced drastically (up to 60%), mainly due to recalculation of the new matrix coefficients that is done after each iteration in loop algorithm. Consequently, additional improvements in this regard are possible and are not covered by this research.

Ivetić et al. (2016) investigated the possibility to speed up network optimization problem by using the ΔQ method for hydraulic calculation inside the evaluation function. In total, four different variants of ΔQ method's implementation were investigated. In the variant in which the exact solution for the flow distribution is searched for, the simplified loop flow equations are solved simultaneously rather than in the matrix form. Comparison is made with the reference GGA solver used in EPANET, in terms of suboptimal solution's objective function value and computational time needed to obtain that solution. Results showed that the use of the ΔQ method in hydraulic computations can accelerate the optimization of a WDN. However, testing was done on two, relatively simple benchmark networks.

As it was implied at the concluding paragraph of the section 2.3.2.2, methods for solving the network hydraulics based on nodal heads system formulation are the most popular nowadays. As a result, almost all popular hydraulic software nowadays uses a node based method, including EPANET (Rossman, 2000), the most popular freely available software package for WDN analysis, which uses the GGA algorithm. When compared to the GGA formulation of the system of equations, as the most prominent node based solver, the loop flow method formulation is often criticized due to the lower sparsity of its Jacobian matrix and the need to identify the network loops in the first place (Todini & Rossman, 2013). However, as mentioned above in recent years some researches proved that there is still room for improvement of the loop flow methods' implementation, and potential use as a viable alternative to node based methods (Alvarruiz & Vidal, 2015; Arsene et al., 2004; Ivetić et al., 2016).

In the recent key papers comparing the node based and loop flow methods, conclusions are made based on testing results conducted on: 1) small number of examples with relatively small number of elements (Todini & Rossman, 2013) or 2) generic extremely looped examples that favor the node base methods (Creaco & Franchini, 2014). One thing that have been overseen is a fact that real life networks have a lot of tree-like parts (branches), with a core that is usually looped. Solving the hydraulics of a network with many tree-like parts is much easier with loop flow method than the node based, since such parts of the network do not require loops identification or any iterative procedures (Stanic et al., 1998). The reason for this is that the initial and final flow distributions in these parts of the network are the same. The FCPA algorithm of Simpson et al. (2014) also supports this statement. Since the real size water distribution networks usually have lot more nodes than loops (e.g. large BWSN2 benchmark network in Ostfeld et al. (2008) has 12,527 nodes and 2,308 loops) and computer algorithms can help in loop identification, the revival of the ΔQ method as an alternative to the node based methods appears very attractive again.

Loops identification procedure remains main disadvantage for application of the ΔQ method, as it may prove to be delicate and time consuming (Todini & Rossman, 2013). In summary, successful implementation of the ΔQ method involves dealing with the two tasks: 1) identification of appropriate set of loops and 2) solving the loops equations. Literature review on available loop identification procedures is given in the following section.

2.3.3 Loop identification procedures

Graph theory algorithms are usually utilized for the purpose of network loops identification. Network is presented in a form of graph which is formed of a set of nodes and a set of connecting links. Graph theory algorithms have been extensively used in the analysis of a WDN in order to perform network decomposition (Deuerlein, 2008) or clustering (Perelman & Ostfeld, 2012). There are many papers dealing with the minimal basis loops detection problem from different aspects, not related to hydraulics. Horton (1987) presented a polynomial algorithm to find the minimum cycle basis (MCB) loops of the graph. The idea behind this approach is to find a super set of MCB loops, and then extract linearly independent ones using the Gaussian elimination. De Pina (1995) used logical framework to look for the geometrical minimal loops starting from the structure without the loops which is known as the spanning tree (ST). There are many other graph theory algorithms that can be used to identify the ST (e.g. algorithm of Kruskal (1956)). Barnat et al., (2002) investigated the possibility of performing a distributed Nested Depth First Search (NDFS) algorithm. Firstly, network graph is decomposed into the maximal strongly connected components, and then the NDFS is applied to each of them. Cerna & Pelanek (2003) presented the distributed explicit fair cycle detection procedure, which is set based and combines advantages of both explicit and symbolic approaches. This procedure is not based on a standard DFS algorithm which, in turn, enables the proposed method to be effectively distributed and parallelized. The standard NDFS technique is relatively fast in finding the loops as it can be run “on the fly” but the algorithm presented by Cerna & Pelanek was significantly faster for the more complicated examples as it can be run in parallel. Work of De Pina was adopted by Kavitha et al. (2004) and Kavitha & Mehlhorn (2005), but the algorithm interpretation was algebraic rather than combinatorial. Accent in both of these works was to find algorithm that will be fast in terms of computational time.

However, despite all this work, not many papers exist regarding the use of the minimal basis loops in combination with the hydraulic calculation of the WDN. In the work of Jha (2007) the Nested Breath First Search (NBFS) algorithm was used to identify the minimal loops in the network, that are later used for the hydraulic simulation of the WDN. The algorithm relies on identification of the signature edges during the first BFS search, and then triggering the second BFS search to find the path between the nodes of the signature edge. Adding the signature edge to this path completes the loop, which is then extracted from the graph. Algorithm is dependent on the selection of the starting node, so all of the nodes are tried in the search process. Still, in order to find the absolute minimal loops heuristic approach is applied based on the identification of bounding edges of the graph representing the WDN. For a real size network, identification of bounding edges can be a hard task, as there is no automated procedure presented for this purpose. In the same research initialization of the pipe flow is done for each loop separately, satisfying node continuity equation. It is noted that the order in which initialization is done can pose a problem, and that the loops might need reordering for the initialization to be successful.

Ivetić et al. (2016) presented an automatic algorithm for the minimal basis loop detection based on the graph theory and relevant heuristics. The algorithm detects the loops that are minimal from the topological point of view (number of links) and it deals only with the network topology without geometry aspect of it. This is done in three steps: 1) the initial set of loops is detected simply based on the graph exploration using BFS; 2) transformation of the ST is performed to obtain a simpler set of loops; 3) decomposition of the set from the second step is performed to obtain the final, minimal set of loops. This algorithm was employed as a pre-processor for the ΔQ hydraulic solver used in the optimization process for the design of WDN.

Creaco & Franchini (2015) extended their previous work (Creaco & Franchini, 2014) and presented the algorithm for automatic identification of minimum loops in a multi-source water network. This algorithm is based on the De Pina framework. It utilizes the Dijkstra (1959) algorithm to search for the shortest path (from the topological viewpoint, meaning that all graph links have the same weight) between the two nodes and is similar to some extent to the methodology presented by Jha (2007). Alvarruiz & Vidal (2015) also presented two additional versions of the algorithm to search for the network loops that will give highly sparse loops matrix. Authors were motivated by the works of Kavitha et al. (2004) and Creaco & Franchini (2014) in which the problem of high computational costs were reported. In the first approach presented (*m3*) loops are simplified by combining them in search for the ones with minimal number of links. The other approach (*m4*) is, in essence, the NBFS algorithm as described in Jha (2007). Algorithm used in this paper is similar to the method *m3*, but adds another criterion to minimize number of shared links between the loops.

In most of previous studies regarding the identification of the minimal basis loops, processing time of algorithms used is investigated and compared. Alvarruiz & Vidal (2015) showed that the method *m4* produced the sparsest loops matrices with the fastest processing time. However, when solving the hydraulics of the WDN the minimal basis loops algorithm should be ran only once in the pre-processing stage if network topology is unchanged. Hence, in problems where multiple runs of the hydraulic solver need to be performed (e.g. optimization), computational burden of the pre-processing stage is not an issue, it is actually the hydraulic calculation time.

2.4 CONCLUSION – IDENTIFICATION OF RESEARCH QUESTIONS

Main research goal of the thesis is development of decision support methodology for sectorization of WDN into DMAs. Section 2.2.3 reviewed methods available in the literature, starting from manual approach (i.e. “trial and error”) to fully automated algorithms coupled with optimization methods. Manual approach is governed by general sectorization criteria of having DMAs of “manageable size” in terms of number of connections, links or network length. It requires significant engineering knowledge of a local expert, familiar with all WDNs specifics, and usually results in an arbitrary sectorization solution that is far from (sub)optimal one.

Algorithms employing optimization methods, while able to search wide solution space in a quest for (sub)optimal solution, suffer from extremely high computational time (e.g. hours/days). So far, computational efficiency has been regarded as something of secondary importance with primary focus on the quality of the obtained solution. Still, even though WDN sectorization is a strategic type decision and hence there is no need to rush things, identifying sectorization solution in reasonable time, (i.e. minutes compared to hours/days/weeks) would allow practicing engineers to analyse lot more different solutions and come up with a better overall solution. Additionally, lists of objective functions used in optimization, as well as constraints and limitations to which optimization is subjected to, have grown too much exhausting all practical aspects important for normal operation of WDN. This opens main research question:

- Is it possible to develop a sectorization algorithm that will, beside general sectorization criteria, implement other practical engineering principles relevant to WDN, and if so,
- Is it possible to achieve better computational efficiency than algorithms employing optimization approach?

Sectorization algorithm's computational efficiency is mainly dependent on number of investigated alternative DMA designs and efficiency of hydraulic solver used to perform hydraulic simulation. For hydraulic simulation purposes, all available sectorization algorithms use EPANET in which node based GGA solver is implemented. As review on available methods for hydraulic simulation (section 2.3.2) indicates, alternative loop-flow based solver can be more efficient than GGA, especially when used inside optimization algorithms. Computational efficiency of loop-flow based solvers is greatly affected by identified set of network loops. This identification is not unique, hence the search for the optimal set of loops makes sense as it leads toward the sparser system of equations, which is then faster to solve. Researchers investigating efficiency of loop-flow based solvers achieved significant speedups per iteration, when compared to the GGA solver. These speedups are then lost in reported overall simulation speedups, indicating methods' implementation problems. This raises following research questions:

- Is it possible to develop new loop identification algorithm able to provide highly sparse solution matrix for loop-flow based method?
- Is it possible to efficiently implement loop-flow based method to preserve achieved speedups per iteration and have them reflect on overall simulation time?

Sectorization can be carried out in different levels of details, and it is better to adopt hierarchical sectorization applicable for different purposes. Hierarchical sectorization is useful in situations where network is naturally hierarchically ordered, and each identified DMA can be further partitioned to obtain finer division while keeping the boundaries of previously established DMAs. Such approach is also very convenient for water companies with limited financial capabilities, primarily focused on improving the system's management with least amount of investment. This concept is fairly uninvestigated, with only one paper addressing the task. Consequently, final research question is:

- Is it possible to implement hierarchical creation of DMAs into the sectorization algorithm?

2.5 Hypothesis

Working hypothesis on which research presented in this thesis is based are:

- Graph Theory algorithms can be successfully implemented in algorithm for sectorization of WDN and algorithm for identification of network loops,
- It is considered that for the purposes of hierarchical DMA planning, basic WDN development plans (e.g. network topology and projected consumption) are known in advance. That is, the research will not consider the uncertainty of these and similar parameters,
- Daily water demand pattern is known for different categories of consumers, meaning that only demand-driven approach will be adopted for hydraulic simulation,
- Loop-flow based method for hydraulic simulation is computationally more efficient than its node based counterparts,
- For the modeling of continuous operation (i.e. transient flow), it is sufficient to use a mathematical model of quasi-steady flow which implies the successive solution of the equations of the steady-state flow in successive time periods. This model is considered to be a sufficient level of approximation for the purposes for which the research is intended.

CHAPTER 3:

IMPROVED LOOP-FLOW METHOD FOR HYDRAULIC SIMULATION

3 IMPROVED LOOP-FLOW METHOD FOR HYDRAULIC SIMULATION

Table of contents:

3.1 INTRODUCTION

Having in mind main research questions identified in the concluding section of the literature review (section 2.4), this chapter presents an improved loop-flow method for efficient hydraulic simulation, which can be beneficial for improving the overall efficiency of WDN sectorization algorithms.

Section 2.3 of the literature review presented an overview of available methods for hydraulic simulation, together with the comparison between them. It was noted that the node based methods are most commonly used today in commercially available software to perform hydraulic calculations. As it was concluded there, loop-flow method (ΔQ method) still has the potential of being computationally faster than the node based methods, since there is usually much smaller system matrix to be solved (especially in the case of the real life networks). However, it requires a loop identification procedure prior to the hydraulic simulation, which proved to be its main disadvantage, as it can be time consuming and inefficient in search for the optimal set of loops.

This chapter presents new efficient method for hydraulic simulation based on the loop-flow method, named TRIBAL-Δ*Q* method. The new method combines the novel TRIangulation BAsed Loops identification algorithm (TRIBAL) with more efficient implementation of the Δ*Q* solver for network hydraulics. In the following text standard loop-flow method will be explained highlighting the need for efficient minimal loop identification procedure (3.2), followed by the description of the TRIBAL-Δ*Q* method (3.3) where thorough presentation of TRIBAL algorithm (3.3.1) and implementation of the Δ*Q* solver (3.3.2) will be given.

3.2 LOOP-FLOW METHOD FOR HYDRAULIC SIMULATION

This section presents the existing loop-flow method, also known as the Δ*Q* method, for hydraulic analysis of looped pressurized networks. Originally presented by Cross (1936), this method is based on the energy conservation principle stating that in every closed WDN loop, the sum of total head losses must be equal to zero:

$$f_{loop} = \sum_{ij \in loop} f_{ij} = \sum_{ij \in loop} R_{ij} Q_{ij}{}^{n} = 0 \tag{3.1}$$

DW or HW equations are used to calculate head losses in all loop pipes. Initial flow distribution, which satisfies the nodal continuity equations, is required to apply this method. Initially assumed flows, denoted as $Q_{ij}^{(o)}$ (Figure 3.1-a), are just an initial guess and most likely do not satisfy the condition for the total head loss in a loops to be zero. When calculating head loss in a loop, clockwise direction of summation is adopted ($f_{loop} = f_{12} + f_{23} - f_{13} \neq 0$). In order to meet this condition, assumed flows are corrected iteratively with flow correction Δ*Q* until the exact flow distribution is obtained ($Q_{ij}^{i+1} = Q_{ij}^{i} + \Delta Q$, *i* being iteration number). Expanding loop head loss equation in a Taylor Series sum and truncating after the first term, which is done under the assumption that flow correction is much smaller than the initial flow, yields:

$$f_{loop}^{i+1} = f_{loop}^{i} + \sum_{ij \in loop} \left(\frac{\partial f_{loop}}{\partial Q_{ij}} \right)^{i} \left(Q_{ij}^{i+1} - Q_{ij}^{i} \right) = 0 \tag{3.2}$$

Derivative of loop head loss equation is:

$$\sum_{ij \in loop} \left(\frac{\partial f_{loop}}{\partial Q_{ij}} \right) = \sum_{ij \in loop} \frac{\partial}{\partial Q_{ij}} \left(R_{ij} Q_{ij}{}^{n} \right) = n \sum_{ij \in loop} R_{ij} \left| Q_{ij} \right|^{n-1} \tag{3.3}$$

Rearranging equation (3.2), and substituting relations (3.1) and (3.3) to solve for loop flow correction, will give iterative solution for the loop flow correction:

$$\Delta Q^{i+1} = -\frac{\sum\limits_{ij \in loop} R_{ij} \left(Q_{ij}^{i} \right)^{n}}{n \sum\limits_{ij \in loop} R_{ij} \left| Q_{ij}^{i} \right|^{n-1}} \tag{3.4}$$

where *i* is the iteration number. This formulation presents the original Hardy-Cross method of balancing heads on loops, which considers each loop in the network independently (i.e. one at the time), instead of simultaneously, different approach which will be discussed later on. After calculation of flow corrections for all loops, pipe flows are updated and the equation (3.4) is used again to calculate new flow corrections. In each step, when estimating loop head loss (numerator in equation (3.4)), flow direction must be accounted for as they can be changed during the calculation. Iterative procedure is repeated until the target accuracy for all flow correction is met. This approach was developed in the pre-computer era and was suitable for hand calculations and relatively simple examples for which loops identification was trivial. For solving more complex examples, approaches that solve loop equations simultaneously are more suitable (e.g. Epp & Fowler (1970)), as it will be explained in the following text.

3.2.1 Loop-flow system of equations

As discussed above, loop flow corrections are introduced to correct the initial flow distribution. Here, it will be explained how the system of equations, that needs to be solved for unknown flow corrections, is formed. First, loop head loss equation (3.1) is rewritten in a manner that will account for a changing pipe flow direction:

$$f_{loop} = \sum_{ij \in loop} f_{ij} = \sum_{ij \in loop} R_{ij} Q_{ij} \left| Q_{ij} \right|^{n-1} = 0 \tag{3.5}$$

Loop flow corrections are introduced in arbitrarily direction, clockwise or counterclockwise. Following the introduction of the flow correction, pipe flows are expressed as sum of initial pipe flows and unknown flow correction flowing through that pipe (e.g. pipe 2-3 in Figure 3.1-b: $Q_{23} = Q_{23}^{(o)} + \Delta Q$). Flow correction is added or subtracted from the initial flow, depending on its orientation.

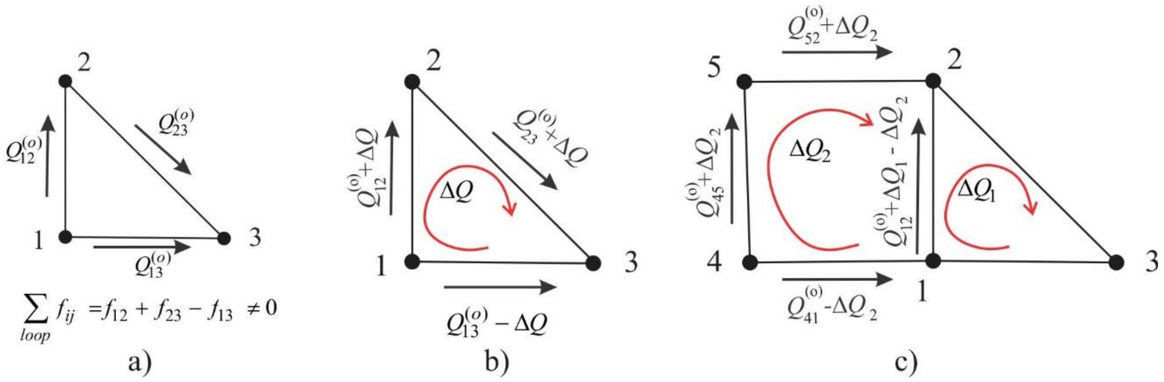

2
$Q_{12}^{(o)}$
$Q_{23}^{(o)}$
1
3
$Q_{13}^{(o)}$
$\sum_{loop} f_{ij} = f_{12} + f_{23} - f_{13} \neq 0$
a)
2
$Q_{12}^{(o)}+\Delta Q$
$Q_{23}^{(o)}+\Delta Q$
ΔQ
1
3
$Q_{13}^{(o)} - \Delta Q$
b)
$Q_{52}^{(o)}+\Delta Q_2$
5
2
$Q_{45}^{(o)}+\Delta Q_2$
ΔQ_2
$Q_{12}^{(o)}+\Delta Q_1 - \Delta Q_2$
ΔQ_1
4
1
3
$Q_{41}^{(o)}-\Delta Q_2$
c)

Figure 3.1 Explanation of the Δ*Q* method

If the introduced flow correction has the same direction as the initial pipe flow, it will be added, and subtracted otherwise. Consider Figure 3.1-c which shows two loops sharing one common pipe (pipe 1-2). This pipe belongs to two loops, so its initial flow is corrected with both flow corrections (ΔQ_1 and ΔQ_2). Initial flow has the same direction as the flow correction ΔQ_1, and opposite direction to the flow correction ΔQ_2. Hence, flow in pipe 1-2 will be $Q_{12} = Q_{12}^{(o)} + \Delta Q_1 - \Delta Q_2$. Since initially assumed flows are constant, head loss equations are now functions of unknown loop flow corrections.

When summing the head losses in a loop (equation(3.5)), head loss for each pipe is accounted for with a sign -1 or +1, depending on the initial flow orientation in that pipe and orientation of the loop flow correction for the loop in consideration, as explained above. Writing head loss equation (3.5) for the second loop in Figure 3.1-c (one with the loop flow correction ΔQ_2) gives:

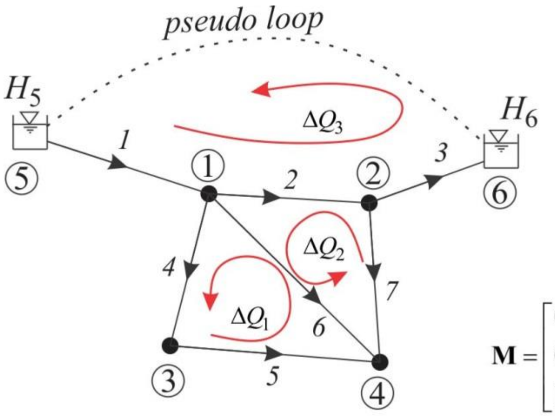
$$\begin{aligned} f_2\left(\Delta Q_1,\Delta Q_2\right) = R_{45}\left(Q_{45}^{(o)}+\Delta Q_2\right)\left|Q_{45}^{(o)}+\Delta Q_2\right|^{n-1}\ldots \\ +R_{52}\left(Q_{52}^{(o)}+\Delta Q_2\right)\left|Q_{52}^{(o)}+\Delta Q_2\right|^{n-1}-\ldots \\ -R_{12}\left(Q_{12}^{(o)}+\Delta Q_1-\Delta Q_2\right)\left|Q_{12}^{(o)}+\Delta Q_1-\Delta Q_2\right|^{n-1}\ldots \\ -R_{41}\left(Q_{41}^{(o)}-\Delta Q_2\right)\left|Q_{41}^{(o)}-\Delta Q_2\right|^{n-1}=0 \end{aligned} \tag{3.6}$$

Equation (3.6) is a nonlinear one, and the number of such equations corresponds to the number of unknown loop flow corrections, which is equal to the number of loops in the network. Assuming the number of nodes (N_n), links (N_l), source nodes (N_r) and number of independently connected components (c) in the arbitrary network, the number of equations to solve is $N = N_L + N_{PL}$, where $N_L = N_l - N_n + c$ is the number of loops and $N_{PL} = N_r - 1$ is the number of pseudo loops. Pseudo loops are formed between the source nodes in the network (i.e. nodes with known head), which is shown in Figure 3.2 illustrating simple example with two reservoirs. According to previous relations, for this simple network (N_n=6, N_l=7, N_r=2 and c=1), number of loops is N_L=7-6+1=2 (loops with flow corrections ΔQ_1 and ΔQ_2), number of pseudo loops is N_{PL}=2-1=1 (loop ΔQ_3), making in total N=2+1=3 loops. In any case, total number of loops in the network N can be expressed as $N = N_l - N_j$, with N_j being the number of junctions (Piller, 1995). In this example $N_j = 4$, thus N=7 – 4=3 again.

The more general form of loop head loss equation (for a random loop k) can be written as:

$$\begin{aligned} & f_k\left(\Delta Q_1,\Delta Q_2,\ldots,\Delta Q_N\right)= \\ & =\sum_{loop=k} sign\cdot R_{ij}\left(Q_{ij}^{(o)}+\sum_{pipe} sign\cdot\Delta Q_p\right)\left|Q_{ij}^{(o)}+\sum_{pipe} sign\cdot\Delta Q_p\right|^{n-1}-\Delta H_{loop=k}=0 \end{aligned} \tag{3.7}$$

where ij is the ij-th pipe in the loop and ΔQ_p is the p-th flow correction (there can be more than one, if pipe is shared between the loops). *Sign* equals one (1) if the direction of the introduced correction ΔQ_p is the same as the direction of the initial flow and minus one (-1) if otherwise. ΔH_{loop} is zero (0) for ordinary loop or equal to the head difference between two reservoirs for the pseudo loop (e.g. for 3rd loop in Figure 3.2 $\Delta H_{loop} = H_5 - H_6$.



$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A_o} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & -1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \Delta\mathbf{Q} = \begin{bmatrix} \Delta Q_1 \\ \Delta Q_2 \\ \Delta Q_3 \end{bmatrix} \quad \mathbf{H_o} = \begin{bmatrix} H_5 \\ H_6 \end{bmatrix}$$

Figure 3.2 Simple network with 2 reservoirs and 3 loops

In total, N equations of the above type (3.7) form the system of nonlinear equations that needs to be solved for the unknown flow corrections for each loop. Nonlinear system written in matrix form is as follows:

$$\mathbf{f}(\Delta\mathbf{Q}) = \mathbf{M}\left[\mathbf{R} \circ \left(\mathbf{Q_o} + \mathbf{M}^T \Delta\mathbf{Q}\right) \circ \left|\mathbf{Q_o} + \mathbf{M}^T \Delta\mathbf{Q}\right|^{\circ(n-1)} - \mathbf{A}_\mathbf{o}^T \mathbf{H_o}\right] \tag{3.8}$$

where **M** is loops incidence matrix of size [N,N_l] relating loops to links in which M_{ij}=1 if the direction of the introduced correction ΔQ for the i-th loop is the same as the direction of the initial flow in j-th link, M_{ij}=-1 if otherwise and M_{ij}=0 if j-th link is not part of the i-th loop; **R** is the link flow resistance vector of size [N_l,1]; $\mathbf{Q_o}$ is the link initial flow vector of size [N_l,1]; $\Delta\mathbf{Q}$ is the loops flow correction vector of size [N,1]; $\mathbf{A_o}$ is the network incidence matrix, based on initial flows direction, reduced to source nodes of size [N_r,N_l] in which A_{ij}=1 if j-th pipe's initial flow inflows the node i, A_{ij}=-1 if j-th pipe's initial flow outflows from the node i and A_{ij}=0 if j-th pipe is not related to the node i; $\mathbf{H_o}$ is the vector of fixed heads at source nodes of size [N_r,1], n is the flow exponent (its value depends on which head loss equation is used – DW or HW) and operator ○ is Hadamard operator used for notation of element wise matrix operations. Aforementioned relevant matrices are illustrated in Figure 3.2.

Nonlinear system has to be linearized in order to be solved. For this purpose, usually NR linearization method is applied (Hoffman, 2001), based on derivation of each loop function f_k into Taylor Series sum and truncation after the first term, yielding iterative solution for the loop flow correction vector in the following matrix form:

$$\mathbf{\Delta Q}_{i+1} = \mathbf{\Delta Q}_i - \mathbf{J}_i^{-1}\mathbf{f}_i \tag{3.9}$$

where i is the iteration number and **J** is the iteration matrix of size $[N,N]$, also known as the Jacobian matrix, containing the derivatives of the head loss functions for each loop, in respect to the introduced loop flow corrections:

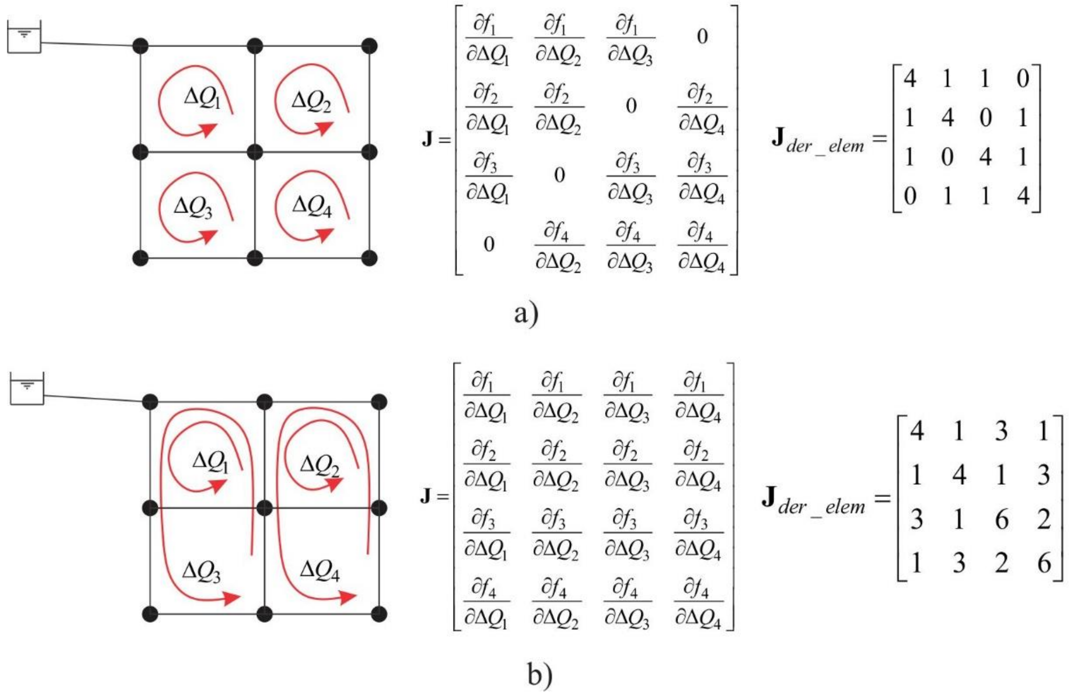
$$\mathbf{J} = \left[\begin{array}{cc:c:c} \frac{\partial f_1}{\partial \Delta Q_1} & \frac{\partial f_1}{\partial \Delta Q_2} & \frac{\partial f_1}{\partial \Delta Q_k} & \frac{\partial f_1}{\partial \Delta Q_N} \\ \frac{\partial f_2}{\partial \Delta Q_1} & \frac{\partial f_2}{\partial \Delta Q_2} & \frac{\partial f_2}{\partial \Delta Q_k} & \frac{\partial f_2}{\partial \Delta Q_N} \\ \hdashline \frac{\partial f_k}{\partial \Delta Q_1} & \frac{\partial f_k}{\partial \Delta Q_2} & \frac{\partial f_k}{\partial \Delta Q_k} & \frac{\partial f_k}{\partial \Delta Q_N} \\ \hdashline \frac{\partial f_N}{\partial \Delta Q_1} & \frac{\partial f_N}{\partial \Delta Q_2} & \frac{\partial f_N}{\partial \Delta Q_k} & \frac{\partial f_N}{\partial \Delta Q_N} \end{array}\right] \tag{3.10}$$

Vector $\mathbf{f}_i$ is residual for the loop equations and it is calculated with $\Delta\mathbf{Q}_i$ according to the equation (3.8).

3.2.2 Effect of identified loops on solution matrix

When compared to the node based formulations of the system of equations, the ΔQ method formulation is often criticized due to the lower sparsity of its Jacobian matrix (e.g. comparison to the GGA algorithm was made in Todini & Rossman (2013)). However, Jacobian matrix sparsity is directly proportional to the identified network loops' structure (Alvarruiz & Vidal, 2015), which is not unique. The simpler the loops' structure is, the simpler the head loss equations will be, meaning that head loss will be a function of a smaller number of flow corrections (ΔQ_p). This means that in the Jacobian matrix more derivative terms of the head loss function f_k will be equal to zero, thus leading to a sparser, more diagonally dominant matrix.

For illustration purposes, the simple network made out of 10 nodes and 13 links, shown in Figure 3.3, is considered. In accordance with previously used notation, example network is made out of one connected component and it has one source node (reservoir) (N_n=10, N_l=13, N_r=1 and c=1). Number of loops then can be determined as $N=N_L+N_{PL}=N_l-N_n+c+N_r-1=4$. Minimal set of loops is shown in Figure 3.3-a, and the set which is not minimal in Figure 3.3-b. Minimal set of loops has 16 links in total (4 loops x 4 links = 16 links) and the other, not minimal set, has 20 links (2 loops x 4 links + 2 loops x 6 links = 20 links). Jacobian matrices (**J**) and matrices containing corresponding number of elements in the sum for each Jacobian derivative ($\mathbf{J}_{der\_elem}$) are also shown in Figure 3.3 for both cases. Each element of the matrix $\mathbf{J}_{der\_elem}(i,j)$ correspond to the number of shared links between loops i and j (e.g. in the second case loops 1 and 3 have 3 common links, thus $\mathbf{J}_{der\_elem}(1,3)=3$). Diagonal elements ($\mathbf{J}_{der\_elem}(i,i)$) correspond to the number of links in a specific loop.



$$\mathbf{J}=\begin{bmatrix} \frac{\partial f_1}{\partial \Delta Q_1} & \frac{\partial f_1}{\partial \Delta Q_2} & \frac{\partial f_1}{\partial \Delta Q_3} & 0 \\ \frac{\partial f_2}{\partial \Delta Q_1} & \frac{\partial f_2}{\partial \Delta Q_2} & 0 & \frac{\partial f_2}{\partial \Delta Q_4} \\ \frac{\partial f_3}{\partial \Delta Q_1} & 0 & \frac{\partial f_3}{\partial \Delta Q_3} & \frac{\partial f_3}{\partial \Delta Q_4} \\ 0 & \frac{\partial f_4}{\partial \Delta Q_2} & \frac{\partial f_4}{\partial \Delta Q_3} & \frac{\partial f_4}{\partial \Delta Q_4} \end{bmatrix} \qquad \mathbf{J}_{der\_elem}=\begin{bmatrix} 4 & 1 & 1 & 0 \\ 1 & 4 & 0 & 1 \\ 1 & 0 & 4 & 1 \\ 0 & 1 & 1 & 4 \end{bmatrix}$$

a)

$$\mathbf{J}=\begin{bmatrix} \frac{\partial f_1}{\partial \Delta Q_1} & \frac{\partial f_1}{\partial \Delta Q_2} & \frac{\partial f_1}{\partial \Delta Q_3} & \frac{\partial f_1}{\partial \Delta Q_4} \\ \frac{\partial f_2}{\partial \Delta Q_1} & \frac{\partial f_2}{\partial \Delta Q_2} & \frac{\partial f_2}{\partial \Delta Q_3} & \frac{\partial f_2}{\partial \Delta Q_4} \\ \frac{\partial f_3}{\partial \Delta Q_1} & \frac{\partial f_3}{\partial \Delta Q_2} & \frac{\partial f_3}{\partial \Delta Q_3} & \frac{\partial f_3}{\partial \Delta Q_4} \\ \frac{\partial f_4}{\partial \Delta Q_1} & \frac{\partial f_4}{\partial \Delta Q_2} & \frac{\partial f_4}{\partial \Delta Q_3} & \frac{\partial f_4}{\partial \Delta Q_4} \end{bmatrix} \qquad \mathbf{J}_{der\_elem}=\begin{bmatrix} 4 & 1 & 3 & 1 \\ 1 & 4 & 1 & 3 \\ 3 & 1 & 6 & 2 \\ 1 & 3 & 2 & 6 \end{bmatrix}$$

b)

Figure 3.3 Comparison of two sets of loops for the simple network: a) minimal loops and b) not minimal loops

Comparing the Jacobian matrices for these two cases it is clear to see that in the case of a) Jacobian is sparser (i.e. it has some elements equal to zero), while in the case of b) matrix is complete (full). This implies that the former Jacobian matrix should be easier to invert, which is necessary to obtain the solution as per equation (3.9). In addition to that, in the case of b) some Jacobian elements have more constituents in their sum due to the increased number of shared links between the loops (e.g. loops 1 and 3 share only one link in the case od a) and three links in the case of b)). Consequently, this will require more time to define Jacobian matrix in the case of b) than in the case of a).

Considering previous discussion, it is of great importance to have an efficient algorithm for identification of network loops that is capable of finding the simplest form of loops (minimal basis loops or near minimal basis loops) in order to save time both for the definition of Jacobian and for its inversion.

3.3 TRIBAL-ΔQ METHOD FOR HYDRAULIC SIMULATION

Combining the new optimal loop identification algorithm with the faster ΔQ solver resulted in new TRIBAL-ΔQ method for the hydraulic analysis of WDN, which is presented in this thesis. The key contributions of the proposed method are:

1. novel method for identification of network loops (TRIBAL) based on graph theory and constrained Delaunay triangulation, the robust and efficient algorithm used in the field of computational geometry, and
2. more efficient implementation of the ΔQ solver with computational load reduction in the calculation of new matrix coefficients.

Flow chart of the TRIBAL- ΔQ method is presented in the following Figure 3.4.

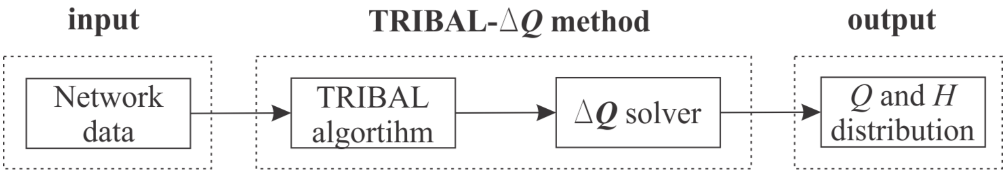


Figure 3.4: Flow chart of the TRIBAL- ΔQ method

TRIBAL-ΔQ method is extensively tested and benchmarked against most popular hydraulic solver nowadays, as it will be discussed in the Chapter 5. In the following sections TRIBAL algorithm is explained, followed by the description of ΔQ solver's implementation and enhancements made.

3.3.1 TRIangulation BAsed Loops identification algorithm (TRIBAL)

TRIBAL algorithm is part of the preprocessing stage of TRIBAL-Δ*Q* hydraulic analysis method, in which network loops are identified. Unlike other available algorithms, that are based on the graph theory and various heuristics (Alvarruiz & Vidal, 2015; Creaco & Franchini, 2015; Ivetić et al., 2016; Jha, 2007), the method proposed here makes use of the graph theory and the Delaunay Triangulation (DT) algorithm (Cheng et al., 2013). For a given planar set of points, DT creates a mesh of triangles in such manner that there are no points inside of the circumcircle of any triangle created. This is the main characteristic of DT and it is known as the Delaunay condition.

In this research, constrained DT (CDT), which predefines some edges of triangulation, is employed, thus resulting in triangulation that may not satisfy Delaunay condition for every triangle. However, this has no negative effect on the TRIBAL algorithm itself. The TRIBAL algorithm’s steps are explained and illustrated on the following simple example. Consider a simple network with 12 nodes and 15 links shown in Figure 3.5-a. Number of loops in the network is equal to 15 – 12 + 1 = 4.

1. **Removing branched parts of the network**

   This is done in order to reduce the size of set of points on which CDT will be performed as in real life networks there are usually significant number of branched parts. For simple network in consideration this step is omitted as there are no branched parts.

2. **Defining the set of constrained edges (CEs) for triangulation and performing the CDT**

   Constrained edges are all links of the network. Result of the CDT is set of triangles defined with two sets: set of nodes and set of edges (Es). Applying this step to the example network yields CDT shown in Figure 3.5-b in which solid lines represent constrained edges (CEs) and dashed lines remaining edges of CDT.

3. **Modify the triangulation if network graph is not planar**

   If graph is not planar it implies having some constrained edges that are crossing each other, which is common in real networks. In that case, triangulation is modified in such manner that some of the crossing edges are excluded and saved for the 8th step of the algorithm. This is easily done by removing only crossing point and does not require running the triangulation again. In the example network, links 4-10 and 9-5 are crossing at one point. Link 9-5 is marked as crossing link, crossing point is removed from CDT which yields in modified CDT (Figure 3.5-c)

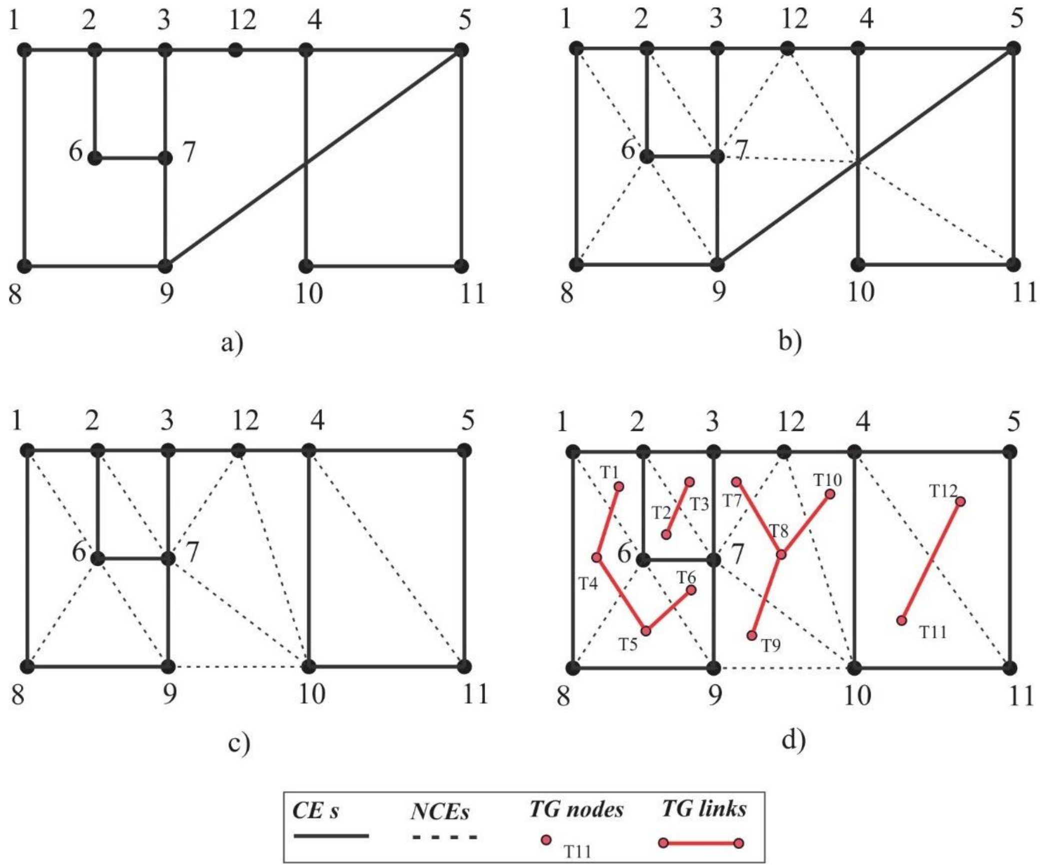


Figure 3.5: TRIangulation Based Loops identification algorithm explained

4. **Create the triangles graph (TG) across NCEs**

Each triangle of CDT is represented as a single node and nodes are connected via links made across the non-constrained edges (NCEs) of triangles. In this manner, new triangles graph (TG) is created, which doesn’t have to be connected but it’s made of number of tree-like triangles subgraphs (graphs without loops). In total, 12 triangles are identified in the CDT, marked as T1 through T12. The TG is formed and it is made of 4 triangles subgraphs – (T1, T4, T5, T6), (T2, T3), (T9, T8, T7, T10) and (T11, T12). This is shown in Figure 3.5-d.

5. **Identification of outer triangle subgraphs and their deletion**

   Outer triangle subgraphs are not bordered from all sides with edges that are in the CEs, thus they are not of interest as their edges do not form network loops and they are deleted from the TG. In the example there is one such subgraph containing triangle T9.

6. **Aggregation of inner triangle subgraphs into loops**

   Simple BFS algorithm is used to propagate through the remaining triangle subgraphs, aggregate triangles in each subgraph and obtain union of their CEs that form the loop around that subgraph. Three remaining subgraphs are aggregated to form three loops: 1st – (triangles: T1, T4, T5, T6) /(links: 6-2, 2-1, 1-8, 8-9, 9-7, 7-6); 2nd – (triangles: T2, T3) / (links: 2-6, 6-7, 7-3, 3-2) and 3rd – (triangles: T11, T12) /(links: 4-10, 10-11, 11-5, 5-4).

7. **Identifying loops created by the crossing links**

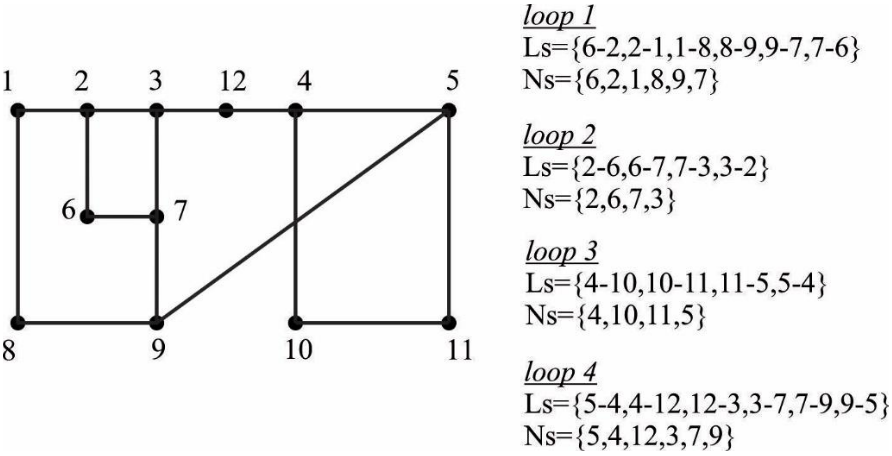
   If crossing links are identified in the step 3 of the algorithm, BFS algorithm is run from one node of the crossing link to find the path to the other node of the link. Identified path, together with the crossing link defines one more loop. This is done for each crossing link in order to identify all such loops. Fourth loop in the example network (5-4, 4-12, 12-3, 3-7, 7-9, 9-5) is found by identifying the path from node 5 to node 9.

8. **Identification of pseudo loops**

   Identification of the pseudo loops is done at the end of this procedure by searching the path between the reservoirs. This is accomplished using the BFS algorithm propagation from one reservoir in the network to all the others. In this manner, it is ensured that the identified pseudo loops will have minimal number of links as this is one of the basic properties of the BFS algorithm itself. There are no pseudo loops in the example network.

It should be noted that the result of TRIBAL algorithm are two arranged sets for each identified loop: 1) Ls – set of links arranged in sequence to close the loop and 2) Ns –

set of nodes also arranged in that same sequence. In this manner the direction of each loop is defined as well (clockwise or counter clockwise). For the simple example used to illustrate application of the algorithm, final result is shown on Figure 3.6.



loop 1
Ls={6-2,2-1,1-8,8-9,9-7,7-6}
Ns={6,2,1,8,9,7}

loop 2
Ls={2-6,6-7,7-3,3-2}
Ns={2,6,7,3}

loop 3
Ls={4-10,10-11,11-5,5-4}
Ns={4,10,11,5}

loop 4
Ls={5-4,4-12,12-3,3-7,7-9,9-5}
Ns={5,4,12,3,7,9}

Figure 3.6: Result of the TRIBAL algorithm for simple example

3.3.2 TRIBAL-ΔQ method implementation

As it was highlighted in the literature review (Chapter 2) and earlier in this chapter, most of the available hydraulic software nowadays use node based methods for performing hydraulic calculations. Most popular of them is probably EPANET (Rossman, 2000) which uses the Global Gradient Algorithm (GGA) (Todini & Pilati, 1987) to solve the network hydraulics for unknown heads and flow distribution. In scientific literature, EPANET with its GGA implementation is considered as etalon, to which all new proposed methods and algorithms are compared to. In order to verify TRIBAL-ΔQ method, presented in this thesis, it is only fair to also compare it to the solver implemented in EPANET. In order to achieve this, the implementation of TRIBAL-ΔQ method was carried on in such manner to enable the use of the original EPANET input files (INP files-basic text files). Same programming language was used for implementation of the hydraulic solver, as it will be explained further in the text. Current implementation of the presented methodology is based on the following key assumptions:

1. only demand driven analysis is available and

2. network topology remains unchanged during the analysis (e.g. closure of the valves or simulation of valves that can change status during simulation are not available (PRVs, PSVs, FCVs) - only pressure breaker valves (PBVs) are implemented).

Both of these assumptions limit the use of the presented methodology for problems that do not result in the topological network changes at this stage.

The implementation of the TRIBAL-Δ*Q* method is divided into two main stages (or blocks), as shown in Figure 3.7. In the first stage (preprocessing stage) TRIBAL algorithm is used to identify network loops with additional preprocessing tasks and in the second stage, improved Δ*Q* solver is used to solve the networks hydraulics.

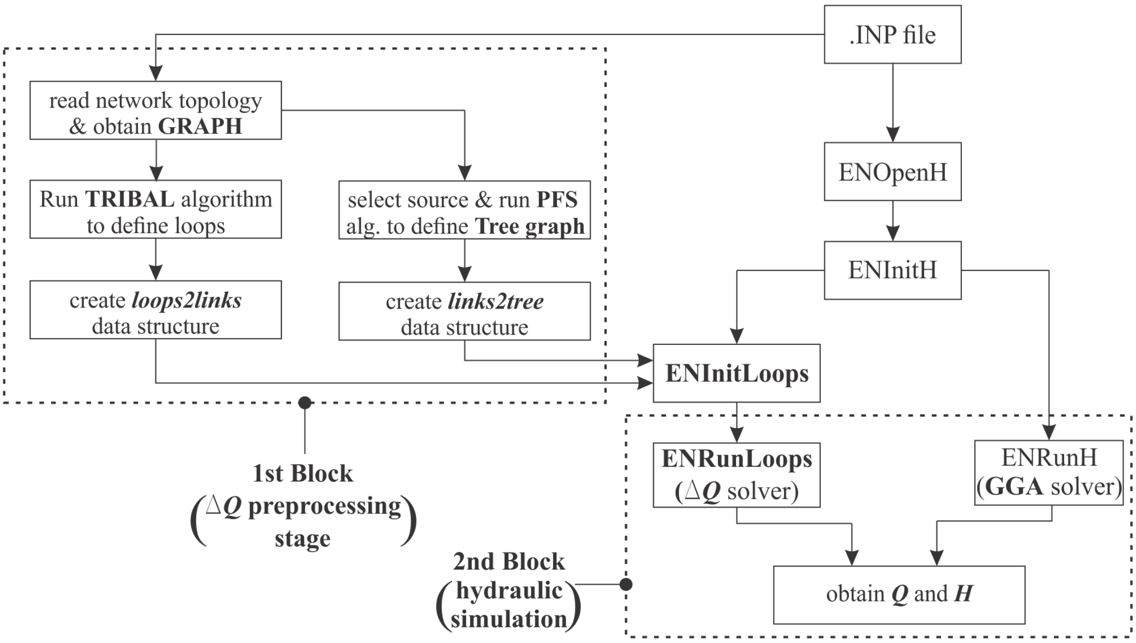

.INP file
read network topology & obtain GRAPH
Run TRIBAL algorithm to define loops
select source & run PFS alg. to define Tree graph
create loops2links data structure
create links2tree data structure
ENOpenH
ENInitH
ENInitLoops
ENRunLoops (ΔQ solver)
ENRunH (GGA solver)
obtain Q and H
1st Block (ΔQ preprocessing stage)
2nd Block (hydraulic simulation)

Figure 3.7: TRIBAL- Δ*Q* method implementation flow chart

3.3.2.1 Preprocessing stage – 1st Block

The preprocessing stage is implemented in the first block where network input data is loaded from the EPANET’s input file (INP) and the processed to prepare it for the follow on hydraulic calculations. The 1st Block’s implementation is done in Matlab2010b (Mathworks, 2010). During this stage, network data is read and used to create the graph representation of the network topology. In this block two data structures are identified, one containing information about network loops structure (***loops2links***) and the other one containing the network spanning tree (***links2tree***), both of which will be used later for the hydraulic simulation.

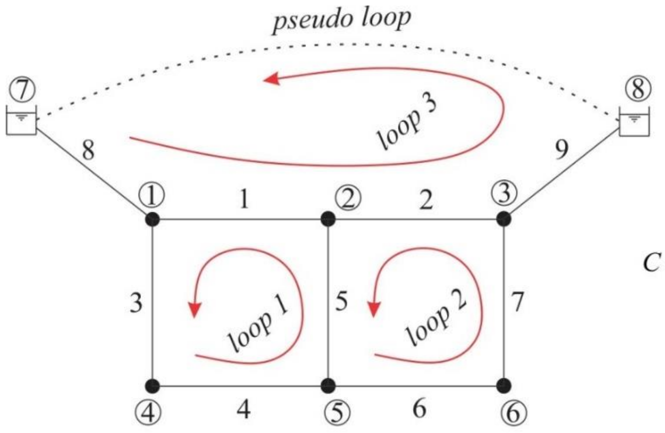
Links2tree structure is obtained by running the Priority First Search (PFS) propagation algorithm from a random selected source node in the network. PFS algorithm is a simple variation of the BFS algorithm, well known from the graph theory (Jungnickel, 2005), in which some type of weight is associated with the links and propagation through the graph is done according to those weights. In implementation used here, pipe's resistance is used as a weight factor in propagation to identify the ST with minimal resistance, as suggested by Alvarruiz & Vidal (2015). Depending on the head loss equation type that is used for the calculation (HW or DW), pipe's resistance is calculated as:

$$R_{ij} = \left\{ \begin{matrix} \dfrac{10.651 L_{ij}}{C_{ij}^{1.852} D_{ij}^{4.871}}; & \text{HW} \\ \dfrac{8\lambda_{ij} L_{ij}}{g D_{ij}^{5} \pi^{2}}; & \text{DW} \end{matrix} \right\} \tag{3.11}$$

Where i and j are end nodes of the pipe, C_{ij} is HW roughness coefficient, D_{ij} is pipe's diameter, L_{ij} is pipe's length, g is gravitational acceleration and λ_{ij} is DW friction factor. Since the value of DW friction factor is flow regime dependent, and considering the fact the pipe flows are still unknown at this point, for the purpose of running the PFS algorithm calculation of DW friction factor is based on equation for turbulent flow regime in rough pipe, which only takes pipe surface's roughness in consideration which is defined in the input file. ***Links2tree*** structure is organized in a manner to flag each link with 1 or -1 depending on its orientation in the ST. This is needed due to the fact that WDN graph is not directional in essence, which means that in the adjacency matrix (graph connectivity matrix - C) any link may be stored as link between nodes *i* and *j* (*ij*) or between nodes *j* and *i* (*ji*). Adjacency matrix C is [N_l,3] size matrix in which first column holds links IDs, second start node (*i*) and third end node (*j*) of the link. Flag for a link is 1 if its orientation in the ST is the same as in matrix C, or -1 if opposite.

Loops2links structure is created using the identified loops, resulting from the TRIBAL algorithm. As it was stated in the previous section (3.3.1), links in the loops are arranged in specific sequence which defines orientation of the loop and its introduced loop flow correction ($\triangle Q$) (clockwise or counter clockwise). Loop links have to be flagged as well (similar to links in the ***Links2Tree*** structure) to allow easier summation of the head loss across the loop in the hydraulic simulation. Loops' links that are at the same time part of the ST (links in ***Links2Tree*** structure) have flag 1 if their orientation in the loop coincides with their orientation in the ST. Otherwise, flag will be -1. For the loops' links that are not in the ***Links2Tree*** structure, flag is determined based on their orientation in the adjacency matrix C. As before, flag is 1 if orientation of the link in loop is the same as its orientation in matrix C, and -1 if else.

Simple example network with 8 nodes and 9 pipes will be used to clarify preprocessing stage explained in the text above. Figure 3.8 shows example network together with its adjacency matrix C and loops (identified using the TRIBAL algorithm).



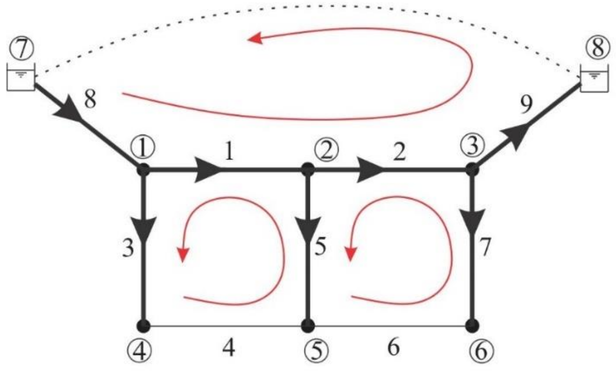
$$C = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 2 & 3 \\ 3 & 4 & 1 \\ 4 & 4 & 5 \\ 5 & 2 & 5 \\ 6 & 5 & 6 \\ 7 & 6 & 3 \\ 9 & 3 & 8 \\ 8 & 7 & 1 \end{bmatrix}$$

loop 1
Ls={4, 5, 1, 3}
Ns={4, 5, 2, 1}

loop 2
Ls={5, 6, 7, 2}
Ns={2, 5, 6, 3}

loop 3
Ls={8, 1, 2, 9}
Ns={7, 1, 2, 3, 8}

Figure 3.8 Example network explaining preprocessing stage of TRIBAL-$\triangle Q$ method



links2tree

$$\begin{bmatrix} 8 & 1 \\ 1 & 1 \\ 3 & -1 \\ 2 & 1 \\ 5 & 1 \\ 7 & -1 \\ 9 & 1 \end{bmatrix}$$

loops2links

$$\begin{array}{c|ccc} \textit{loop 1} & 1 & 4 & 1 \\ & 1 & 5 & -1 \\ & 1 & 1 & -1 \\ & 1 & 3 & 1 \\ \hline \textit{loop 2} & 2 & 5 & 1 \\ & 2 & 6 & 1 \\ & 2 & 7 & -1 \\ & 2 & 2 & -1 \\ \hline \textit{loop 3} & 3 & 8 & 1 \\ & 3 & 1 & 1 \\ & 3 & 2 & 1 \\ & 3 & 9 & 1 \end{array}$$

Figure 3.9 Results of the preprocessing stage for example network

Results of the preprocessing stage, stored in the structures ***links2tree*** and ***loops2links*** are showed in Figure 3.9. Obtained networks' ST is marked with thick arrowhead lines. As it can be seen, in ***links2tree*** structure, link 3 is flagged with -1, since in the ST it is oriented from the node 1 to node 4, and in the adjacency matrix C its orientation is from node 4 to node 1. In ***loops2links*** structure that same link is part of the loop 1, in which it is oriented from node 1 to 4 (same as in the ST), hence its flag is 1 here.

3.3.2.2 Hydraulic simulation – 2nd Block

Hydraulic simulation is performed in the 2nd Block of the algorithm (see Figure 3.7) in which either the GGA solver (already present in EPANET) or the improved ΔQ solver (added to EPANET source code) is run. Numerous steps have been taken in order to implement the ΔQ based solver in EPANET's source code in the most computationally efficient way possible. This includes writing new routines and creating new structures to have easier memory access to obtain faster code execution. Functions **ENOpenH**, **ENInitH** and **ENRunH** are built in EPANET toolkit functions and in order to implement the ΔQ solver in the EPANET's source code, two new functions are added to EPANET toolkit – **ENInitLoops** and **ENRunLoops**. **ENInitLoops** uses data structures previously obtained in the 1st Block to allocate additional memory required for the simulation purposes and **ENRunLoops** function performs hydraulic simulation based on the ΔQ solver. **ENRunLoops** is only an interface function that allows for different subroutines, added for efficient implementation of the ΔQ solver, to be executed. Finally, once the system of WDN equations is solved, network flows (Q) and heads (H) are determined at the end of the 2nd Block.

Iterative solution of the network hydraulics, performed in **ENRunLoops** function, starts with the calculation of initial flow distribution in the network which is done in two steps. First, the ST, contained in the ***links2tree*** structure is used to propagate backwards to the source node applying mass balance equation in the nodes of the network. In the second step, links that are not in the spanning tree are assigned initial flow corresponding to a velocity of 1 ft/s (same as what EPANET uses) and flows from the first step are updated.

For the calculation of the Jacobian matrix elements, EPANET’s ***newcoeff*** routine is used. This routine calculates the inverse head loss derivatives for each link (*ij*) in the network with respect to the link flow as follows:

$$\left(\frac{\partial f_{ij}}{\partial Q_{ij}}\right)^{-1} = \left(\frac{\partial}{\partial Q_{ij}}\left(R_{ij}Q_{ij}\left|Q_{ij}\right|^{n-1}\right)\right)^{-1} = \left(R_{ij}\left(Q_{ij}\left(n-1\right)\left|Q_{ij}\right|^{n-2}\frac{\left|Q_{ij}\right|}{Q_{ij}}+\left|Q_{ij}\right|^{n-1}\right)\right)^{-1} = \frac{1}{nR_{ij}\left|Q_{ij}\right|^{n-1}} \qquad (3.12)$$

For the ΔQ solver, head loss derivatives for each link (f_{ij}) in the loop are calculated with respect to the loop flow correction of the loop in consideration (ΔQ_k):

$$\begin{aligned}
&\frac{\partial f_{ij}}{\partial \Delta Q_k} = \frac{\partial}{\partial \Delta Q_k}\left(R_{ij}\left(Q_{ij}^{(0)}+\Delta Q_k\right)\left|Q_{ij}^{(0)}+\Delta Q_k\right|^{n-1}\right) = \ldots \\
&\qquad R_{ij}\left(\left|Q_{ij}^{(0)}+\Delta Q_k\right|^{n-1}+\left(Q_{ij}^{(0)}+\Delta Q_k\right)\left(n-1\right)\left|Q_{ij}^{(0)}+\Delta Q_k\right|^{n-2}\frac{\left|Q_{ij}^{(0)}+\Delta Q_k\right|}{Q_{ij}^{(0)}+\Delta Q_k}\right) = \ldots \qquad (3.13) \\
&\qquad nR_{ij}\left|Q_{ij}^{(0)}+\Delta Q_k\right|^{n-1}
\end{aligned}$$

In both equations above, R_{ij} has to be recalculated in each iteration as it is not constant (e.g. if DW head loss equation is used). Since pipe flow is $Q_{ij} = Q_{ij}^{(0)} + \Delta Q_k$, it is clear that the second expression, shown in equation (3.13), is only the inverse of the previous one shown in equation (3.12). Hence, EPANET’s ***newcoeff*** routine is used to calculate derivatives which are then inverted only once and stored in the corresponding link structure in order to avoid multiple inversions in further steps of the algorithm. Simple summation of head loss derivatives across the loop’s links is used to form the Jacobian matrix for the ΔQ solver’s system of equations (as per equation (3.10) $\mathbf{J}(m,k)$ = $\frac{\partial f_m}{\partial \Delta Q_k} = \sum_{ij\in m}\frac{\partial f_{ij}}{\partial \Delta Q_k}$). System is solved for the unknown loop flow corrections using the same Cholesky factorization used in the EPANET’s code (***linsolve*** routine) based on node reordering and symbolic decomposition of the matrix (George & Liu, 1981).

After each iteration, link flows are updated with the calculated loop flow corrections according to equation $\mathbf{Q}_i = \mathbf{Q_o} + \mathbf{M}^T\mathbf{\Delta Q}_i$, and link coefficients are recalculated. This is where the most significant action is taken in order to improve computational efficiency of the algorithm. In EPANET, coefficients are recalculated for all links after each iteration, because system of equations is node based. In the ΔQ solver system of equation is loop based hence, coefficients are recalculated only for the links that are in the loops. This proved to be of great importance for networks that have many branched parts (almost all real networks) as computational burden of calculating the new coefficients could mask the real advantages of the ΔQ solver, as reported by other researchers (Alvarruiz & Vidal, 2015). To some extent it is similar to the FCPA algorithm presented by Simpson et al. (2014). This is discussed further in the Chapter 5 presenting results.

Iterative calculations are done until the target accuracy (*eps*) is met. The convergence criterion used here is the same one used in EPANET, i.e. the sum of all absolute pipe flow changes divided by the sum of all pipe flows has to be smaller than some predefined, target value (default value in EPANET is 0.001):

$$eps = \frac{\sum_{k=1}^{N_l} \left| Q_k^{i+1} - Q_k^{i} \right|}{\sum_{k=1}^{Nl} \left| Q_k^{i+1} \right|} \tag{3.14}$$

where *k* is the number of the pipe and *i* is the iteration number. After the pipe flows distribution is determined, head losses for pipes are calculated and the ST is used to calculate the heads in the nodes.

CHAPTER 4

UNIFORMITY AND HEURISTIC BASED ALGORITHM FOR SECTORIZATION OF WDN

4 UNIFORMITY AND HEURISTIC BASED ALGORITHM FOR SECTORIZATION OF WDN

Table of contents:

4.1 INTRODUCTION

It was noted in Chapter 2 that there are number of different approaches and methods that are used for the purpose of identifying and creating the sectors (DMAs) in WDN. Identification of main research questions revealed that, despite all recent advancements made, scope still exist to further improve water network sectorization methodology, especially in terms of usability for practicing engineers. Aspects in which these improvements can be made are: 1) implementation of practical engineering principles, relevant to the WDN, to govern the sectorization process, 2) computational efficiency of the algorithm and 3) implementation of hierarchical sectorization. Computational efficiency of sectorization procedure can be improved by coupling the sectorization algorithm with new TRIBAL-ΔQ method for hydraulic simulation presented in Chapter 3. Possible benefits of such coupling are particularly promising if sectorization procedure involves an optimization method in which multiple hydraulic simulations are required.

Section 4.2 of this chapter presents new algorithm for sectorization of water distribution network, named DeNSE (Distribution Network SEctorization), as a part of decision support methodology for sectorization of WDN proposed in this thesis. At this stage of development DeNSE algorithm is not coupled with any optimization method. Instead, common sense engineering heuristic is implemented and used to search for (sub)optimal sectorization solution, in order to reduce the computational burden generally inherited from the use of optimization. Section 4.3 presents further extensions of DeNSE algorithm, which are the design of hierarchical sectorization in WDN (4.3.1) and coupling with an optimization method (4.3.2).

4.2 DeNSE Sectorization Algorithm

As discussed in Chapter 2, sectorization process should start with the definition of key sectorization objectives and design criteria, followed by the identification of PIs that will be used to assess impact of interventions made in the network. Tracking the water balance in the network is main sectorization objective adopted in DeNSE algorithm. Designing the sectorization solution that requires least investment in the equipment necessary for creation of DMAs (flow meters and isolation valves), while keeping the same level of network's operational efficiency are main design criteria. Such set of design criteria is most appealing to many water utilities, especially in the developing countries, which operate highly inefficient WDNs with significant amount of water and revenue losses. Two PIs are adopted to evaluate the effects of the sectorization on network's operational performance: 1) Resilience Index (*Res*), reflecting post-sectorization reliability of WDN (Todini, 2000) and 2) Water Age (*WA*), surrogate metrics for water quality reflecting water retention rate in the WDN.

Presented DeNSE sectorization algorithm employs newly developed network uniformity index, which drives decomposition into clusters that are not only within predefined size limits, but are also uniform in size as much as possible. Uniformity index also favors sectorization in which cluster's connecting links are ones with smaller diameters, indirectly providing economically more favorable solution as installation of valves and flow meters on smaller diameter links will be less costly. High computational efficiency is achieved using simple and common sense engineering heuristics, rather than optimization tools, to position the valves and flow meters on the connecting links and create DMAs. Furthermore, algorithm presented here does not come up with a single sectorization solution, but with a range of feasible solutions, giving the freedom to the decision makers to select the one best suited for their needs. Following section will cover detailed explanation of DeNSE algorithm. Algorithm is tested on large real-sized benchmark network, used in literature for various modelling tasks. Obtained results are presented in Chapter 5, where thorough comparison with other results previously reported in the literature is also made.

DeNSE algorithm relies on graph theory for identification of Strongly Connected Components (SCCs), which are afterwards aggregated into clusters based on newly presented network uniformity index (*U*). It requires calibrated WDN model as an input and runs through 3 stages to come up with the best sectorization solution, as shown in Figure 4.1. First stage is a pre-processing stage in which all the relevant network data is obtained from the WDN model and prepared for the follow run of the sectorization algorithm. WDN decomposition into clusters is done in the second stage, based on the uniformity index. This stage also involves selecting the best solutions that will be hydraulically analysed in the following stage. Third stage involves heuristic, engineering based positioning of the valves and flowmeters on clusters connecting links in order to create DMAs, extended period hydraulic analysis of the solutions and evaluation of solution’s cost and adopted PIs (aforementioned *Res* and *WA*). Finally, feasible solutions are ranked and preferable solution is selected. Each of the three stages will be explained in details in the following text.

4.2.1 Input Data

The new sectorization algorithm requires the following input data:

1. Calibrated WDN network model in the form of EPANET input file, which contains all relevant data (topology, hydraulic characteristic, demand data, etc.)
2. Minimum ($n_c^{\min}$) and maximum ($n_c^{\max}$) number of property connections per DMA, as well as total number of connections in the network (n_c), since number

of connections per node is usually not available. Recommendations about these values can be found in number of available guidelines for DMA creation, and usually it is considered that number of connections should be in the range of 500-5000 (Farley, 2001; Morrison et al., 2007). It is considered that having DMAs larger than 5000 connections is not practical as it becomes difficult to distinguish leakages from the night flow data, while taking more time to allocate them. It should be noted that the preferable DMA size is network specific, influenced by many factors and should be determined based on a thorough analysis of the specific data relevant to the network in consideration.

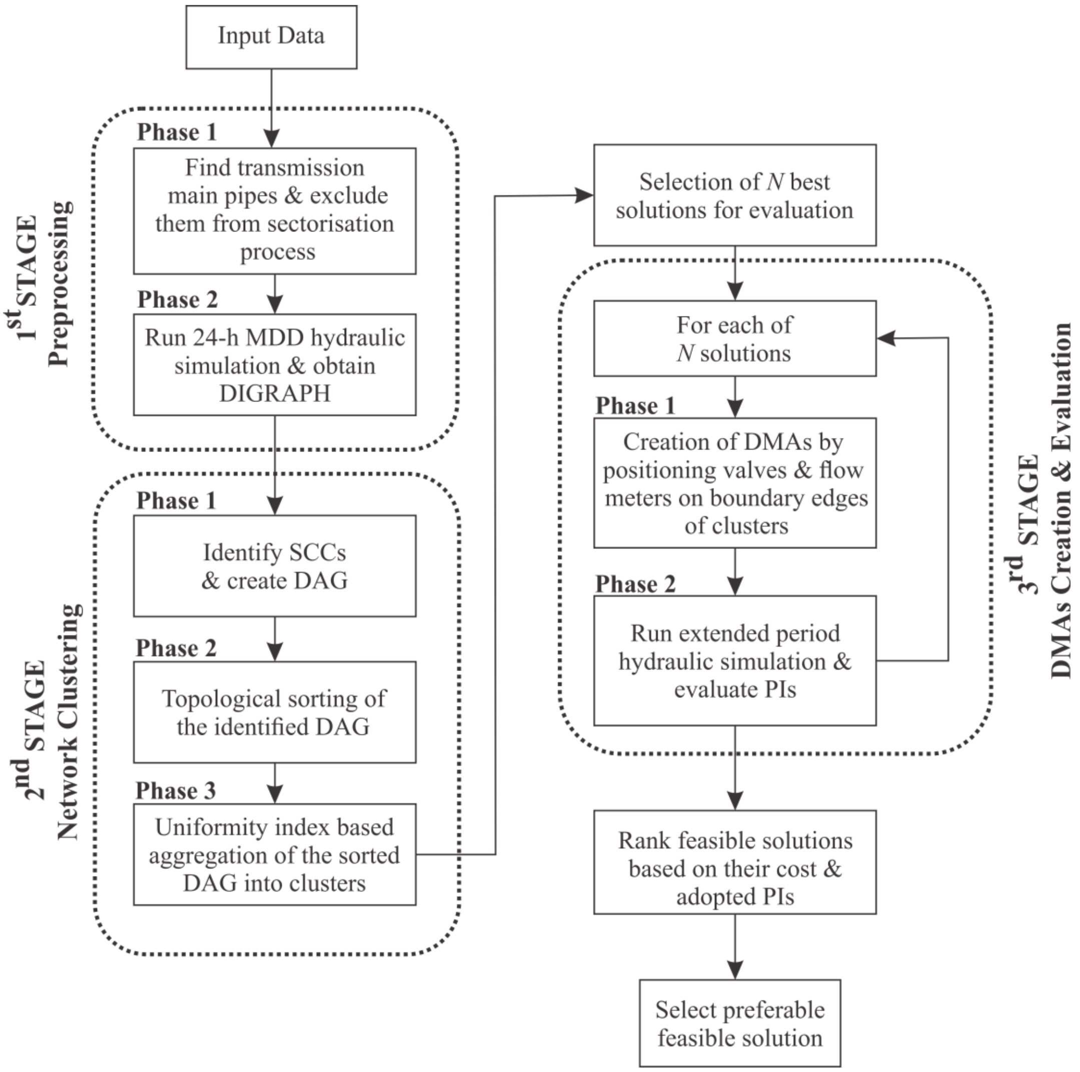

Input Data
1st STAGE
Preprocessing
Phase 1
Find transmission main pipes & exclude them from sectorisation process
Phase 2
Run 24-h MDD hydraulic simulation & obtain DIGRAPH
2nd STAGE
Network Clustering
Phase 1
Identify SCCs & create DAG
Phase 2
Topological sorting of the identified DAG
Phase 3
Uniformity index based aggregation of the sorted DAG into clusters
Selection of N best solutions for evaluation
3rd STAGE
DMAs Creation & Evaluation
For each of N solutions
Phase 1
Creation of DMAs by positioning valves & flow meters on boundary edges of clusters
Phase 2
Run extended period hydraulic simulation & evaluate PIs
Rank feasible solutions based on their cost & adopted PIs
Select preferable feasible solution

Figure 4.1: Flow chart of the DeNSE algorithm

3. Transmission main threshold diameter (D_{main}). Large diameter pipes connected in series, running from the networks main source(s) are considered a transmission main. These are the pipes that convey water between the reservoirs and tanks and serve as a main supply paths in the network. In this methodology they are excluded from any interventions. As with the DMA size, value of D_{main} is network specific, usually being 300-350 mm (Ferrari et al., 2014).
4. Pipe closure threshold diameter (D_{tr}). Pipes having diameter equal or larger than this diameter ($D_{ij} \geq D_{tr}$) will not be considered for possible closure within the heuristic procedure for positioning the valves and flowmeters (part of the 3rd Stage of the algorithm). By default, algorithm uses first class of diameter lower than the D_{tr} (e.g. if D_{main} is 350 mm, D_{tr} will be 300 mm), but user can specify a different value. However, this will affect the number of isolation valves and flowmeters required to create the DMAs and consequently, the solution cost.
5. Minimum required and maximum allowed pressures in the network, p_{min} and p_{max}, as well as the maximum Water Age (WA_{max}) allowed in the network as a water quality indicator.

4.2.2 Preprocessing – 1st Stage

In the first stage, there are two phases (see Figure 4.1).

Phase 1. In the first phase, transmission mains are defined, based on the D_{main} value, and excluded from the sectorization process. For this purpose, network is explored using slightly modified BFS algorithm, simultaneously starting from all main source nodes (reservoirs). BFS algorithm is modified to prioritize propagation through the links with diameters equal or greater than D_{main}.

Phase 2. In the second phase, 24-hour Maximum Day Demand (MDD) hydraulic simulation of the analysed WDN is performed to determine the orientation of pipes (based on water flow directions obtained in the simulation). As a result, directional graph (DIGRAPH) G is defined with two sets $G = <N, C>$, set of network nodes N and set of network links C, where each link is presented with ordered pair of nodes. Network links with changing flow directions are identified as non-oriented (or links that can have both flow directions). Both of these phases are illustrated on a simple example network shown in Figure 4.2.

The example network consists of 17 nodes, two of which are reservoirs, and 21 links. Links connecting reservoirs are identified as transmission mains and are excluded from further analysis. Remaining part of the network, connected to the transmission main with one link in node 9 should be partitioned into DMAs. Illustrated orientations of the remaining links are determined based on the results of the hydraulic analysis. Two of those links are identified as not oriented, and putting that in the context of water networks, those are usually pipes (links) that are connecting tanks with the rest of the network. So in an example network, nodes 8 and 2 could be tanks. In a real size water networks parallel links often exist too. That is why a link should also have an identification number, because it cannot be uniquely defined with ordered pair of nodes.

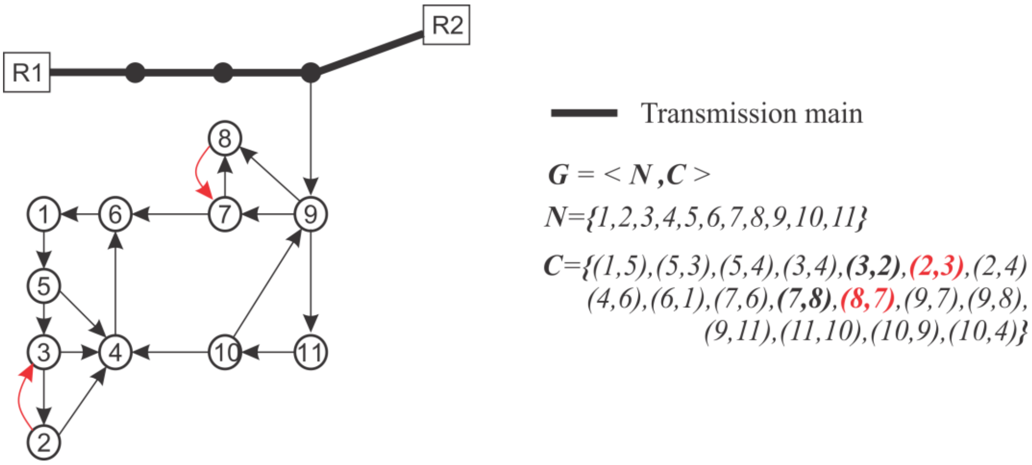


Figure 4.2: Digraph presentation of a simple network with 2 sources and 2 undirected links

4.2.3 Network clustering – 2nd Stage

Partitioning of the WDN into clusters is performed in the second stage of the algorithm. It is done in three phases.

Phase 1. First step is to identify the SCCs within the previously created DIGRAPH. Strongly connected component (SCC) is a term from Graph Theory, and it is defined as a subgraph in which each node can be reached from any other node within that subgraph. Therefore, SCCs are parts of network where water is circulating during the simulation. Due to that fact, control of the water balance and/or water pressure regulation in SCC parts of the network could be difficult to achieve, so the idea is to detect SCCs and treat them as aggregated nodes in further network analysis and clustering. Algorithms for the extraction of SCCs from digraph are well known in the Graph Theory. The Gabow algorithm (Gabow, 2000) is used in the methodology shown here. It is chosen due to its linear computational time, which makes it more efficient compared to the others. This is significant as algorithm has to be able to deal with large networks efficiently. Gabow's algorithm requires only one pass through the network (DIGRAPH) with recursive call of the DFS algorithm with arbitrary selection of the starting node.

For illustration purposes, a simple digraph shown in Figure 4.2 is used.

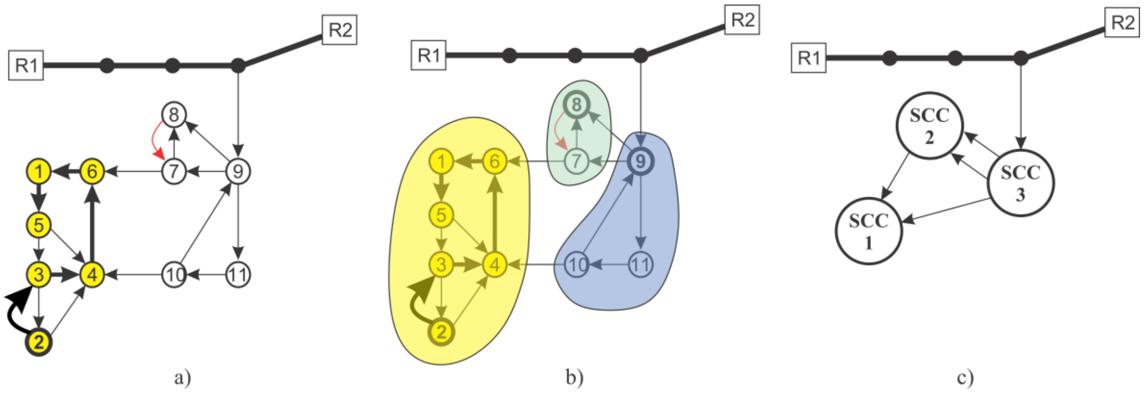


Figure 4.3: DIGRAPH transformation to DAG: a) Start the DFS; b) Detected SCCs; c) Newly formed DAG

Starting the DFS search from the node 2, nodes 3, 4, 6, 1 and 5 are visited (Figure 4.3-a). During the DFS search, a check is made weather the selection of the next node forms a cyclic path or not. If yes, nodes forming the cyclic path are identified as a SCC. The algorithm continues until no further propagation is possible. In example shown in Figure 4.3, the first SCC component identified is composed of nodes 2, 3, 4, 6, 5 and 1. No further propagation is possible, so the DFS starts again from randomly selected node, chosen from the set of nodes that were not visited during the first search. Assuming that the randomly selected node is node 9, and after nodes 11 and 10 are visited, the second SCC composed of these three nodes is identified. DFS search is repeated again starting from node 8, and third SCC composed of nodes 8 and 7 is detected (Figure 4.3-b). At the end, aggregated DIGRAPH is composed of three identified SCCs. The DIGRAPH can also be viewed as set of aggregated nodes and two remaining connected to transmission main with one link (Figure 4.3-c). The most important property of new aggregated DIGRAPH is its acyclicity, indicating it is a DIGRAPH without cycles. Such graph is referred to as Directed Acyclic Graph (DAG), and in terms of water network is very important, because it clearly separates source from the demand nodes and hence, makes the sectorization of network easier.

Phase 2. In the second phase topological sorting of the identified DAG is conducted. DAG nodes, represented with SCCs, are sorted from the downstream end, and this order will be used to drive aggregation of the DAG from the most peripheral SCCs. Again, simple implementation of recursive DFS algorithm, as explained in Sedgewick & Wayne (2011), is used for this purpose. In an example shown in Figure 3c, topological sorting yields following topologically sorted list (TSL): SCCs: SCC1, SCC2 and SCC3.

Phase 3. In this phase aggregation of the sorted DAG, composed of the SCCs connected between each other and connected to the transmission main, is conducted based on the newly presented network uniformity index (*U*). Network uniformity index is defined as follows:

$$U = u_{net} u_v w_{agg} \tag{4.1}$$

where u_{net} is network uniformity in terms of cluster size, u_v is uniformity of the DMAs size vector and w_{agg} is relative weight of aggregated links. Each of these variables are explained in the following paragraphs, followed by the explanation of aggregation algorithm itself.

Each cluster is characterized with its size (d_i), calculated as sum of all nodal demands within that cluster - $d_i = \sum_{j=1}^{N_n^i} q_j$, N_n^i being number of nodes in *i*-th cluster. Network uniformity (u_{net}) measures average deviation of clusters size from the preferred DMA size (d_{pref}). Ideally, all clusters should have size equal to the d_{pref} but, obviously, this is not possible in real networks. Preferred DMA size is calculated based on minimum and maximum DMA size, d_{min} and d_{max}, as $d_{pref} = \frac{d_{\min} + d_{\max}}{2}$. Minimum and maximum

DMA size are calculated based on the daily average total demand in the WDN (Q_{tot}, available from the WDN model), the number of minimum and maximum connections in the zone ($n_c^{\min}$ and $n_c^{\max}$) and total number of connections in the WDN (n_c), given as an input data:

$$d_{\min} = \frac{Q_{tot}}{n_c} n_c^{\min}; \quad d_{\max} = \frac{Q_{tot}}{n_c} n_c^{\max} \tag{4.2}$$

Network uniformity is calculated based on the triangular function *f* that quantifies “quality” of cluster size in the rage [0,1] (Figure 4.4). If a cluster *i* has a size $d_i = d_{pref}$, its value of *f* will be the best, i.e. f_i=1. If a cluster has a different size (i.e. larger or smaller than d_{pref}) it will have the value of f_i <1. Since the function *f* is equilateral, both larger and smaller cluster are equally penalized. Extremely large clusters (larger than 2d_{pref}), are scored with the lowest value of $f_i=0$. Finally, network uniformity is calculated as:

$$u_{net} = \frac{\sum_{i=1}^{N_{cl}} f_i}{N_{cl}} \tag{4.3}$$

where N_{cl} is number of clusters for a given sectorization. Note that maximum value of u_{net} is 1, if all clusters are equal to d_{pref}, and minimum value is zero.

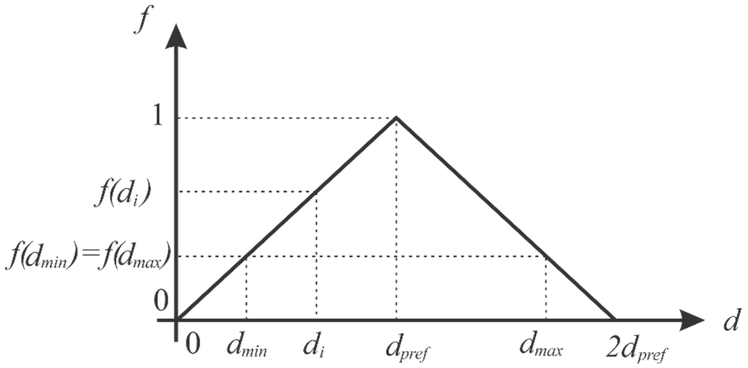


Figure 4.4: Triangular function *f* quantifying cluster size

Sizing clusters in the range d_{min} – d_{max}, and as much as possible close to d_{pref}, is one sectorization objective. Sizing them equally is the other one. Sizes of all clusters form the normalized size vector of a specific sectorization into N_{cl} clusters –

$\boldsymbol{d^n} = \{d_1^n, d_2^n, d_3^n \ldots, d_{N_{cl}}^n\}$, where $d_i^n = \dfrac{d_i}{\sum_{i=1}^{N_{cl}} d_i}$ Uniformity of this vector is calculated as its Euclidean norm (L2 norm):

$$u_v = \sqrt{\sum_{i=1}^{N_{cl}} \left(d_i^n\right)^2} \tag{4.4}$$

If all clusters are equal in size (e.g. d_1=d_2=d_3=…=d_{pref}), which is the most preferable case, uniformity of the size vector is:

$$u_v^{best} = \sqrt{\left(\frac{d_1}{N_{cl} d_{pref}}\right)^2 + \left(\frac{d_2}{N_{cl} d_{pref}}\right)^2 + \ldots} = \sqrt{\frac{N_{cl}\left(d_{pref}\right)^2}{N_{cl}^2\left(d_{pref}\right)^2}} = \sqrt{\frac{1}{N_{cl}}} \tag{4.5}$$

If all nodes are part of the same cluster, meaning worst case scenario in which there is no clustering, uniformity of the demand vector is u_v^{worst} = 1. To be consistent with the ranging values of network uniformity metrics (u_{net}), where 0 is the minimum value and 1 is maximum, uniformity of the size vector is scaled to the same range to yield final form of equation for its calculation:

$$u_v = \begin{Bmatrix} 1 - \dfrac{u_v\sqrt{N_{cl}} - 1}{\sqrt{N_{cl}} - 1}; & N_{cl} > 1 \\ 0 \qquad ; & N_{cl} = 1 \end{Bmatrix} \tag{4.6}$$

Relative weight of aggregated links is calculated as:

$$w_{agg} = \frac{\sum_{i=1}^{n_l^{agg}} D_i}{\sum_{i=1}^{n_l} D_i} \tag{4.7}$$

Where n_l is total number of links, n_l^{agg} is number of links within the clusters, and D_i is links diameter. In case of large number of clusters there will be unaggregated connecting links than in the case of small number of clusters. Hence, the value of w_{agg}

will be smaller in the former than in the latter case. Minimum value of w_{agg} is zero, if no aggregation is done, and 1 if all SCCs are aggregated into one cluster.

Flowchart of the aggregation algorithm (Phase 3), based on uniformity index metrics described above, is given in Figure 4.5. At the initial step of the algorithm, all identified SCCs are considered as individual clusters. Aggregation of SCCs into clusters is done in a step by step manner, propagating through topologically sorted DAG obtained in Phase 2 and aggregating in each step nodes whose aggregation will contribute the most to the network uniformity.

Aggregation algorithm presented here is essentially a Greedy optimization method, in which aggregation direction is determined based on the highest uniformity index gain. As with all similar type algorithms, it is not guaranteed that the global optimum solution will be found. However, the benefit is that generally a good sub-optimal solution can be found with significant computational time savings when compared to other optimization algorithms. Aggregation of identified SCCs into clusters is iteratively carried out through three steps: 1) Identification of candidates for aggregation, based on topologically sorted DAG (Phase 3a); 2) First aggregation – Selection and aggregation of candidate with highest uniformity gain ($\varDelta U$) (Phase 3b); 3) Second aggregation done if predefined conditions, specific to the WDNs, are met (Phase 3c). Algorithm steps will be explained now, followed by the illustrative application on a simple example.

Phase 3a. The aggregation algorithm takes topologically sorted DAG (*TSL*), obtained in the previous step (Phase 2), as an input data. At initial step all SCCs are considered as individual clusters, meaning that initial number of clusters corresponds to the number of identified SCCs. Initial network uniformity index is calculated (*U*), and iterative part of the algorithm starts. Sink nodes in *TSL* (*SNs*) (nodes not having outlet links) are identified and marked as visited during propagation. If there are not such nodes, the algorithm terminates as this means that all nodes are merged into one cluster and there is no more possibilities for aggregation. First step of the algorithm is identification of candidate nodes for aggregation (Phase 3a). For all marked sink nodes (*SNs*) upstream nodes are identified (*UNs*). Aggregation of node *SNs(i)* to its upstream node *UNs(j)* is possible only if all nodes downstream of node *UNs(j)* are marked as visited during propagation. At the end of the Phase 3a list of possible aggregations is created (*AGG*) and algorithm proceeds to the Phase 3b in which first aggregation is done.

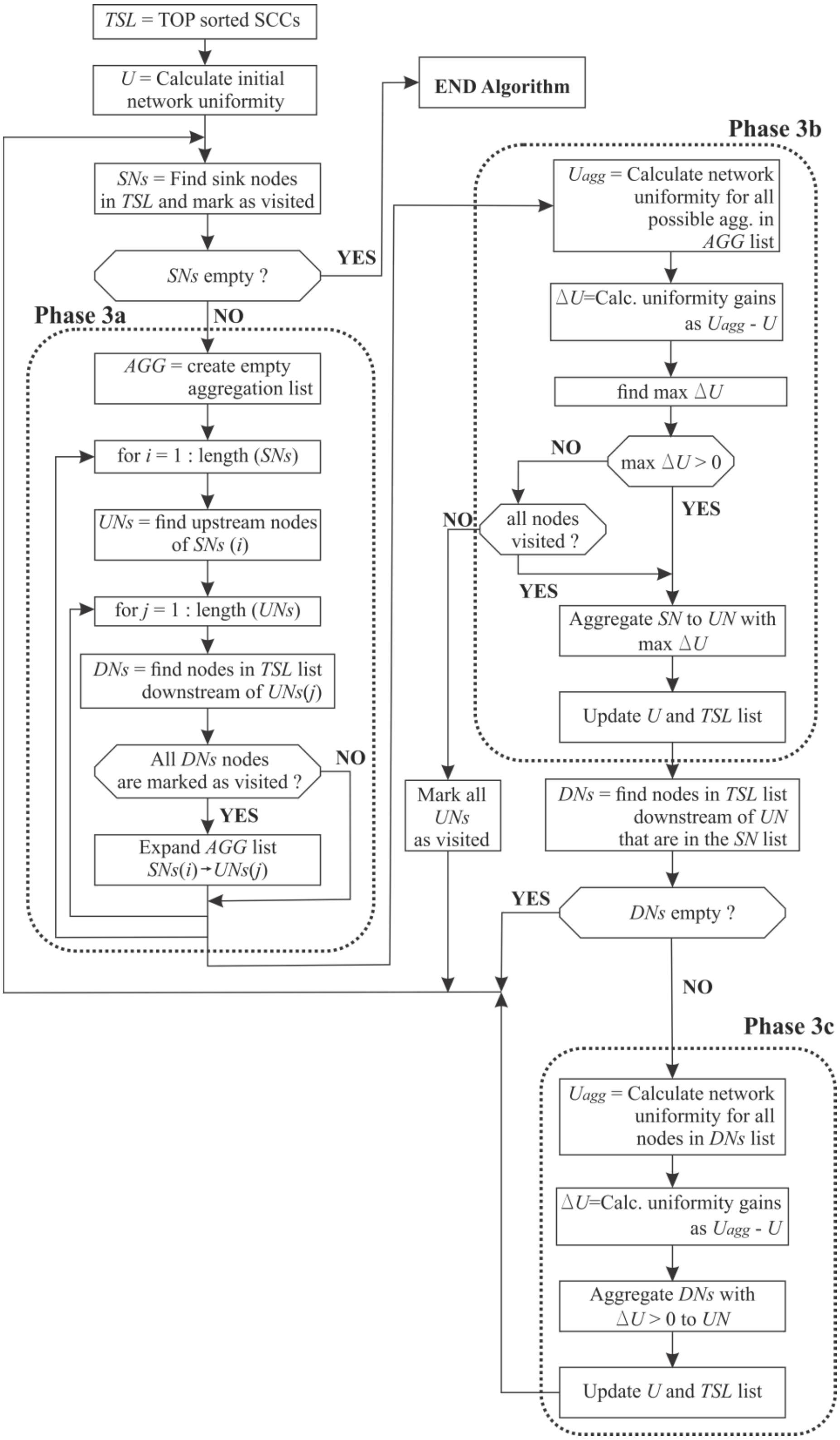

TSL = TOP sorted SCCs
U = Calculate initial network uniformity
END Algorithm
Phase 3b
SNs = Find sink nodes in TSL and mark as visited
SNs empty ?
YES
Phase 3a
NO
AGG = create empty aggregation list
for i = 1 : length (SNs)
UNs = find upstream nodes of SNs (i)
for j = 1 : length (UNs)
DNs = find nodes in TSL list downstream of UNs(j)
All DNs nodes are marked as visited ?
NO
YES
Expand AGG list SNs(i)→UNs(j)
Uagg = Calculate network uniformity for all possible agg. in AGG list
ΔU=Calc. uniformity gains as Uagg - U
find max ΔU
NO
max ΔU > 0
YES
NO
all nodes visited ?
YES
Aggregate SN to UN with max ΔU
Update U and TSL list
Mark all UNs as visited
DNs = find nodes in TSL list downstream of UN that are in the SN list
YES
DNs empty ?
NO
Phase 3c
Uagg = Calculate network uniformity for all nodes in DNs list
ΔU=Calc. uniformity gains as Uagg - U
Aggregate DNs with ΔU > 0 to UN
Update U and TSL list

Figure 4.5: Flowchart of the aggregation algorithm (Phase 3)

Phase 3b. For all aggregations contained in the *AGG* list, new network uniformity indices are calculated (U_{agg}). Uniformity gains (ΔU) for possible aggregations are calculated as the differences in uniformity between new indices and index from the previous step $\Delta U = U_{agg} - U$. Maximum uniformity gain is selected and if it is positive ($\Delta U_{\max}>0$) algorithm continues, aggregates corresponding sink node (*SN*) to its corresponding upstream node (*UN*), updates *U* and *TSL* and terminates Phase 3b. If $\Delta U_{\max} \leq 0$ and there are still nodes that are not visited during propagation, it indicates that none of the current aggregation possibilities contributes to the network uniformity. Hence, aggregation is not done, all upstream nodes are marked as visited and algorithm returns to the beginning of its looped part. On the other hand, if all nodes are already visited (and $\Delta U_{\max} \leq 0$), it means that local optimum has been reached and network uniformity cannot be improved further. Having in mind that the goal is not to find the solution with highest uniformity index, but rather a set of solutions with "good" value of uniformity index that will be hydraulically analysed later in 3rd Stage (Figure 4.1), aggregation will continue until all nodes are aggregated into one cluster. If there are other visited nodes downstream of the upstream node *UN* (*DNs*), in which *SN* has been just aggregated, algorithm proceeds to the Phase 3c.

Phase 3c. Again, array of new network uniformity indices (U_{agg}) corresponding to aggregation of nodes from *DNs* to the *UN* node is calculated, followed by the calculation of uniformity gains (ΔU). All nodes with positive ΔU are aggregated to the *UN*, *U* and *TSL* are updated and algorithm returns to the beginning of its loop. Phase 3c is implemented to avoid the case in which small peripheral nodes remain unaggregated until late stages of aggregation. This could happen as such nodes usually have relatively small uniformity gain and aggregation would continue past them further upstream.

Application of described aggregation algorithm will be illustrated on a simple example shown in Figure 4.6. Example is derived from Figure 4.3-c, adding 6 more SCCs for illustration purposes. For the sake of simplicity, total demand of 20 L/s is assigned to all 9 SCCs. Diameters of the links connecting SCCs are shown in Figure 4.6 in millimeters. Minimum (d_{min}) and maximum (d_{max}) DMA size are set to 40 and 80 L/s respectively, which yields preferred DMA size (d_{pref}) of 60 L/s. Numerical values for all aggregation steps are shown in Table 4.1 to complement graphical illustrations in Figure 4.6.

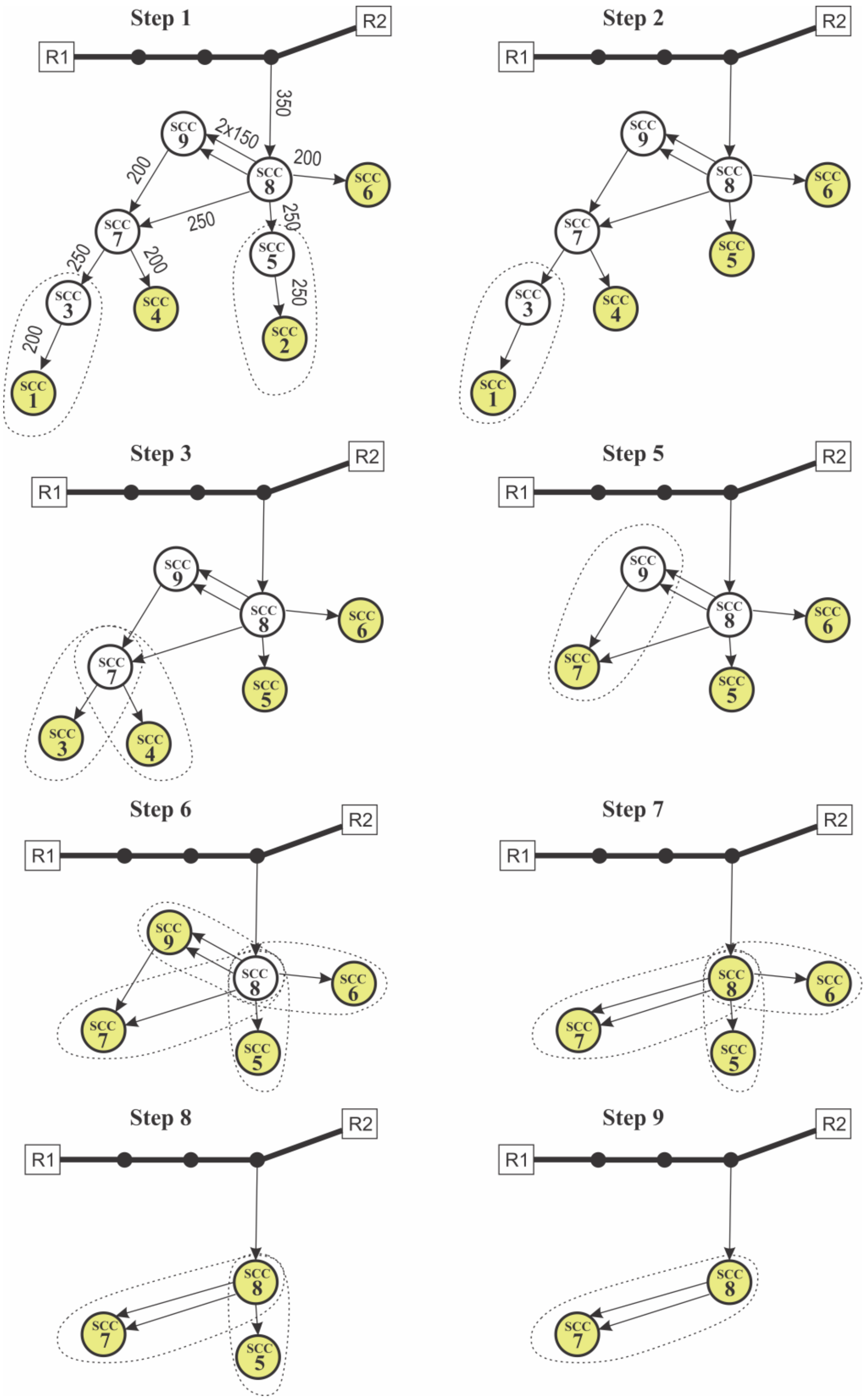


Figure 4.6: Aggregation algorithm illustrated on a simple example

Table 4.1: Numerical values for aggregation steps

step	***AGG***	N_{cl}	u_{net}	u_v	w_{agg}	U_{agg}	ΔU	U
0		9	0.333	1.000	0.000			0.000
1	1-3	8	0.375	0.977	0.095	0.035	0.035	
	2-5	8	0.375	0.977	0.119	0.044	**0.044**	0.044
2	1-3	7	0.429	0.964	0.214	0.088	**0.045**	0.088
3	3-7	6	0.500	0.916	0.333	0.153	**0.064**	0.153
	4-7	6	0.500	0.963	0.310	0.149	0.060	
4	4-7	5	0.467	0.845	0.429	0.169	**0.016**	0.169
5	7-9	4	0.417	0.763	0.524	0.166	-0.003	
6	5-8	4	0.583	0.845	0.548	0.270	0.101	
	6-8	4	0.583	0.889	0.524	0.272	0.103	
	7-8	4	0.417	0.763	0.548	0.174	0.005	
	9-8	4	0.583	0.889	0.571	0.296	**0.127**	0.296
7	7-8	3	0.333	0.683	0.786	0.179	-0.117	
	5-8	3	0.556	0.856	0.690	0.328	0.032	
	6-8	3	0.778	0.950	0.667	0.493	**0.196**	0.493
8	7-8	2	0.333	0.652	0.881	0.192	-0.301	
	5-8	2	0.500	0.985	0.786	0.387	**-0.106**	0.387
9	7-8	1	0.000	0.000	1.000	0.000	**-0.387**	0.000

▨ - highest uniformity gain

Aggregation steps are as follows:

- Step 1: Identified sink nodes are 1, 4, 2 and 6, and they are marked as visited during propagation. Viable candidates for aggregation to corresponding upstream nodes are determined in Phase 3a: node 1 corresponding to node 3 and node 2 corresponding to node 5. In Phase 3b it is concluded that aggregation of node 2 will contribute more to the network uniformity than aggregation of node 1 (as $\Delta U_{2\text{-}5} > \Delta U_{1\text{-}3}$) hence node 2 is aggregated to node 5 and the algorithm proceeds to the next step.
- Step 2: In this step the only viable aggregation is aggregating node 1 to node 3. Since $\Delta U_{1\text{-}3}$ is positive, aggregation is done and the algorithm continues.
- Step 3: Now there are two possible aggregations – node 3 to 7 and node 4 to 7. Node 3 is aggregated as it is a better alternative (see ΔU values in Table 4.1).

- Step 4: In this case the algorithm enters Phase 3c, since node 4 was already visited during the propagation and it is located downstream of node 7 (to which node 3 was just aggregated). Uniformity index gain for this aggregation is positive and hence node 4 is also aggregated into node 7.
- Step 5: Aggregation of node 7 to 9 is the only viable alternative left. As its gain is negative, there will be no aggregation and node 9 is marked as visited.
- Step 6: There are 4 possibilities for aggregation: 5 to 8, 6-8, 7-8 and 9-8. The highest uniformity gain provides aggregation of node 9, hence this node is aggregated to node 8.
- Step 7: Out of the 3 possible aggregations, the best one is aggregation of node 6 to node 8 (see corresponding ΔU values in Table 4.1).
- Step 8: Both aggregation alternatives (7-8 and 5-8) have negative uniformity index gains, meaning that sub-optimal aggregation solution is reached. From this point on, any aggregation will decrease network uniformity index. Since $\Delta U_{5\text{-}8} > \Delta U_{7\text{-}8,}$ node 5 is aggregated.
- Step 9: Finally, node 7 is aggregated to node 8 creating a single cluster which terminates the algorithm.

Evolution of network uniformity index is shown in Figure 4.7, where uniformity is plotted against the number of clusters corresponding to each aggregation step.

Figure 4.7 illustrates that the highest uniformity index value corresponds to network sectorization into 3 clusters with total demands of 40, 60 and 80 L/s. Sizes of all three clusters are within predefined DMA size limits (40 – 80 L/s). Clusters are connected with three links between them. Next aggregation step leads to the solution with 2 clusters, having total demands of 80 and 100 L/s. Obviously this solution does not meet DMA size constraints, as one cluster is larger than d_{max}. However, there are now two links connecting 2 clusters which requires less isolation valves and flow meters to isolate them and create DMAs than in the case with 3 clusters.

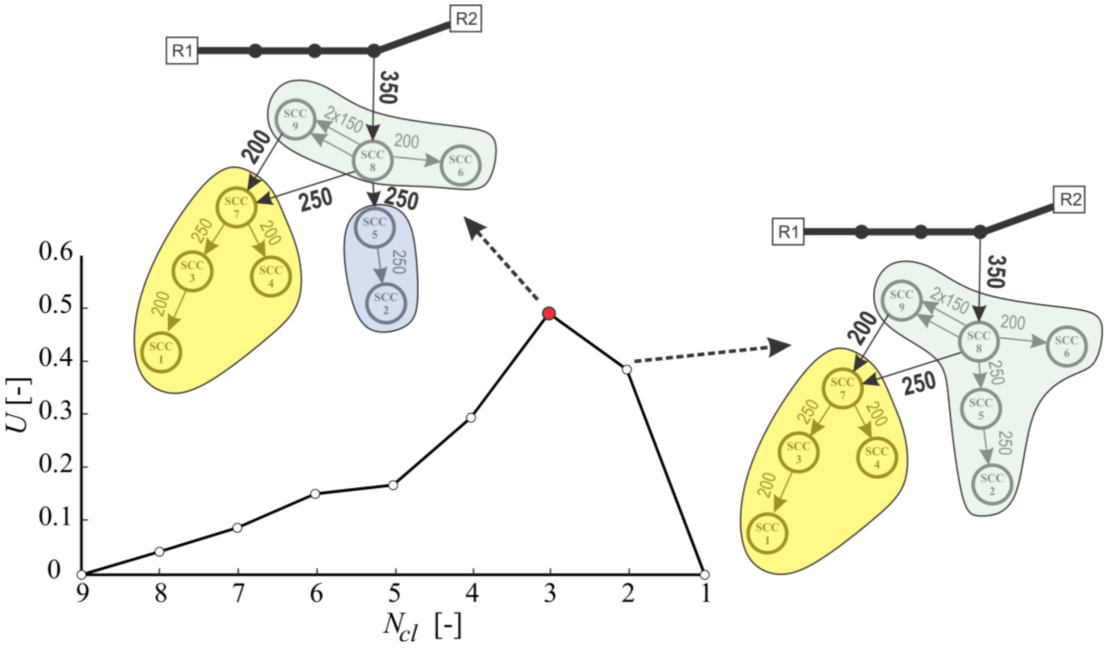


Figure 4.7: Evolution of network uniformity index during aggregation process for simple example

Additional clarification about the evolution of network uniformity index during the aggregation process is made here. As Figure 4.7 illustrates, uniformity index is initially zero when all SCCs are considered as individual clusters and no links are aggregated. Gradually its value increases reaching maximum at some aggregation step, after which it begins to decline. Uniformity index will finally reach the value of zero, since all SCCs are part of a single cluster at the end of aggregation procedure. However, generally this may not be the case depending on the network layout and identified transmission main.

In simple example analyzed above, after removal of the transmission main all SCCs are part of one independent district connected to the main. Figure 4.8 illustrates a different, more complex and general example. In this case there are 11 SCCs that make three independent districts connected to the main. Hence, aggregation will start from 11 clusters and at the end of the procedure there will be three clusters. Further aggregation is not possible as clusters are separated by the removal of transmission main and are not connected to each other. Also, terminal uniformity index value will be different from zero.

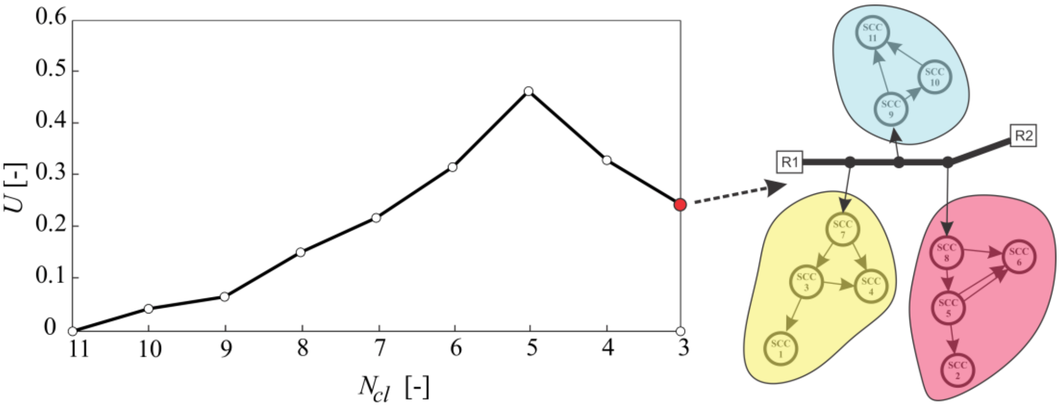


Figure 4.8: Evolution of network uniformity index during aggregation process in general case

4.2.4 Creation of DMAs and evaluation of solutions – 3rd Stage

The clustering of DAG, made out of identified SCCs, based on network uniformity index is finished at the end of the 2nd Stage. As described above, clustering is done in a step by step process, preserving the data about clusters' structure at each aggregation step. Note that the number of aggregation steps corresponds to the number of identified clustering solutions. Obviously, not all of the solutions are of interest, only the ones with high value of network uniformity index are.

Prior to execution of the 3rd Stage itself, selection of solutions that will be hydraulically analyzed and evaluated for satisfaction of selected PIs is made. Default number of solutions (N_{sol}) for the 3rd Stage analysis is set to 15, which is considered to be large enough set of solutions for multi-criteria ranking. Selection of solutions is made based on network uniformity index values obtained at each aggregation step. Solution with the highest uniformity index is selected (best solution), together with additional 14 solutions from succeeding aggregation steps. Additional solutions are on the recession part of uniformity index plot (Figure 4.7) characterized by lower value of uniformity index (than the best solution) but also by smaller number of clusters. Clusters connected only to the transmission main and having size smaller than d_{min} are removed from each solution and excluded from further analysis. Such clusters are below minimum DMA size limit and will not be considered as a DMA.

After the selection of solutions for evaluation has been made, 3rd Stage of the algorithm is evoked. There are two main steps in the 3rd Stage: 1) Conversion of clusters into DMAs (Phase 1) and 2) Evaluation of solutions' cost and adopted PIs (Phase 2).

Phase 1. To convert clusters into DMAs, flow meters and isolation valves have to be positioned on clusters' boundary edges. Positioning of the flow meters and valves is done based on engineering heuristics. Continuing from the simple example used to describe aggregation algorithm (Figure 4.7), let us consider the solution with the highest value of network uniformity index. This solution has 3 clusters and 4 boundary edges to be considered for installation of flow meters/valves. For methodology illustration purposes, another branch of transmission main and 4 boundary edges are added to this solution (Figure 4.9-a).

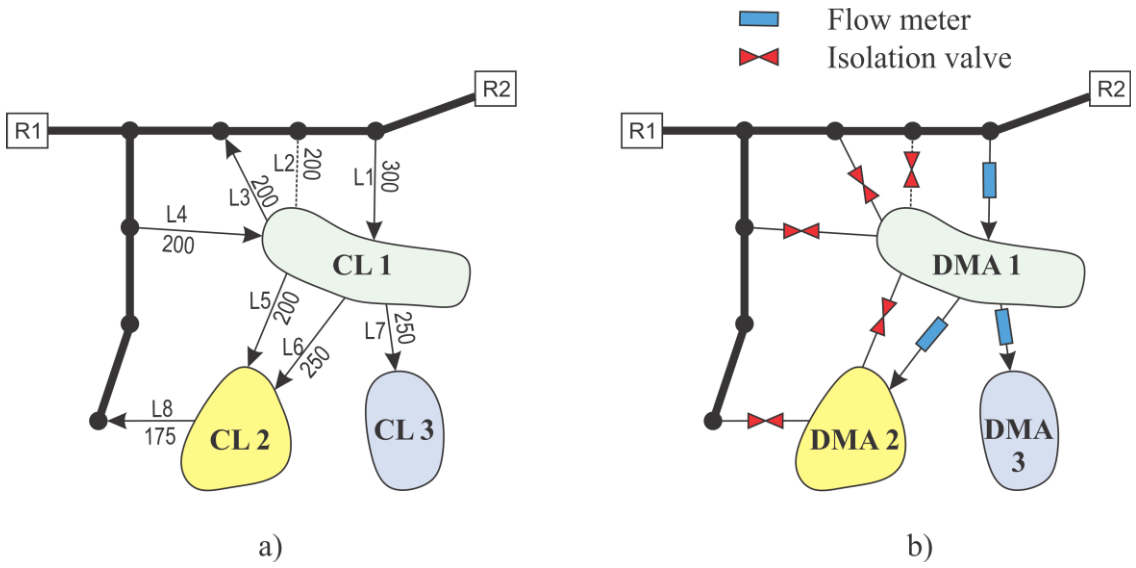


Figure 4.9: Heuristic positioning of flow meters and isolation valves to convert clusters to DMAs (3rd Stage's Step 1)

Boundary edges are labeled as L1 through L8. Flow orientations during 24-hour MDD hydraulic simulation, obtained in Phase 1 of the 1st Stage, are indicated with arrows. Pipes with changing direction are indicated using dashed lines without arrows. Non-oriented pipes are only those connecting clusters with the transmission main, as identified clusters resulted from the DAG analysis. In this case, there is only one such pipe (L2). Heuristic procedure is comprised of the following three steps:

- Non-oriented pipes are identified and the pipes in which absolute difference between maximum and minimum flow rate is less than 0.2 L/s are marked for closure, as this is considered as negligible flow rate (L2).
- All links connecting clusters with the transmission main, oriented from the clusters to the main, are closed (L3 and L8 in the example shown). These are the pipes returning water from the demand nodes into the main, hence it is considered that they are not supply pipes and can be closed without negative effects on systems hydraulics.
- Supply pipes of each cluster (oriented towards cluster) are analyzed independently. It is sufficient to analyze only supply pipes as graph is a DAG and one clusters' output pipes are others' supply pipes. Supply pipes for a cluster are identified and pipe with largest maximum inflow to the cluster (Q_{max}) is considered as main supply pipe, and will not be considered for closure. Maximum capacity of this pipe (C_{max}) is calculated based on maximum allowable velocity of 2.0 m/s, and its remaining capacity is $C = C_{max} - Q_{max}$. All remaining supply pipes having diameter larger than threshold value supplied as an input (D_{tr}) are candidates for closure. Their maximum capacities are calculated in the same manner (c_{max}), and they are analyzed one by one, starting from the link with the lowest maximum flow rate (q_{max}). When a pipe i is considered for closure, resulting residual input capacity is calculated subtracting i-th pipe capacity as $C_{cl} = C + \sum c_{\max} - c_{\max}(i)$. If reduced capacity is still larger than the maximum flow rate carried by the i-th pipe ($C_{cl} \geq q_{max}(i)$), pipe is closed by setting its capacity to zero ($c_{max}(i) = 0$). Iterating through this procedure, candidate pipes are closed until input capacity is fully exhausted. Applying this to the simple example in figure 8 would result in closure of input pipe L4 for cluster CL 1 and pipe L5 for cluster CL 2. Cluster CL 3 has only one input link, so it remains opened.

At the end of the Phase 1, flow meters and isolation valves are positioned on the clusters boundary edges converting them into DMAs (Figure 4.9-b).

Another approach for positioning of the flow meters and valves is the optimization method, which considers each boundary pipe as closed or open. Since it is not uncommon that number of boundary edges exceeds several tens in the case of real WDNs, the optimization method could be very time consuming hence it was not implemented here. In addition to that, most of that time is spent on testing of unfeasible alternatives.

Phase 2. After definition of its DMAs boundaries, each solution is subjected to the extended period hydraulic simulation to investigate the effects of modifications made to the network. Firstly, feasibility of solution is considered through evaluation of pressure constraints in each node:

$$p_{i,t} \geq p_{\min} \; ; \; p_{i,t} \leq p_{\max} \tag{4.8}$$

where $p_{i,t}$ is pressure in i-th node in simulation time step t, and p_{min} and p_{max} are minimum and maximum allowable pressures in the network. If solution does not meet pressure constraints it is considered unfeasible and it is excluded from further analysis.

For each feasible solution, cost and two adopted PIs are calculated, together with a number of other parameters used for evaluation of the solution. Cost of the solution and the adopted PIs are calculated as follows:

1. *Cost* – Cost of the solution is calculated based on the unit cost of devices installed to create the DMAs (flow meters and isolation valves). Unit cost functions are taken from De Paola et al. (2014) and shown in Figure 4.10.

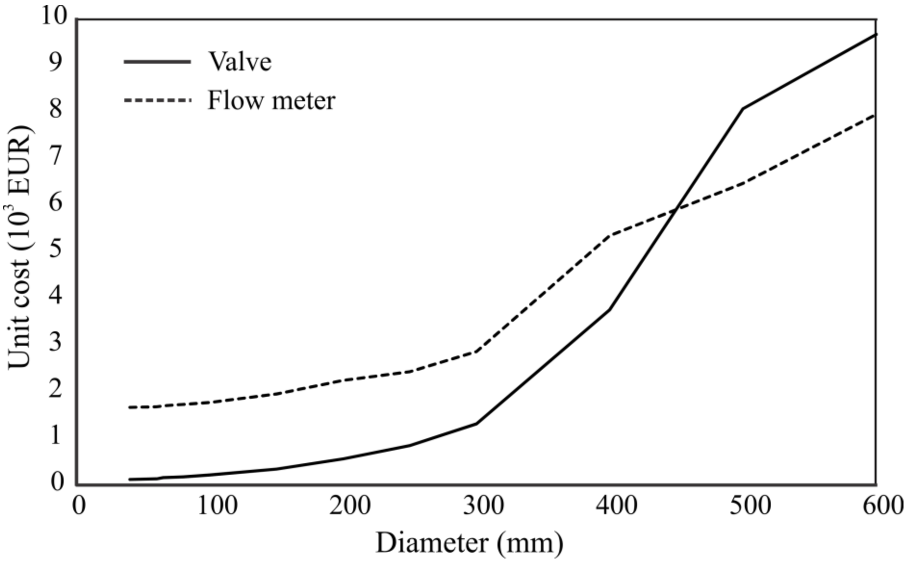


Figure 4.10: Unit costs functions of flow meters and isolation valves

2. Average network resilience index (Todini, 2000), calculated as mean value over the simulation time period (*T*). Resilience index is represented as the ratio of residual amount of power in the network after satisfaction of nodal demands and maximum amount of power that can be dissipated in the network internally, while satisfying the nodal demands and minimal pressure constraints:

$$Res = \underset{T}{mean}\left(\frac{\sum_{i=1}^{n_j} q_i \left(h_i - h_i^* \right)}{\sum_{j=1}^{n_r} Q_j H_j + \sum_{k=1}^{n_p} \frac{P_k}{\gamma} + \sum_{l=1}^{n_t} Q_l H_l - \sum_{i=1}^{n_j} q_i h_i^*} \right) \tag{4.9}$$

where n_j is number of junctions, n_r is number of reservoirs, n_p is number of pumps, n_t is number of tanks, q_i is nodal demand at node *i*, h_i is nodal head at node *i*, h_i* is minimum nodal head at node *i*, Q_j is discharge from the reservoir *j*, H_j is head in reservoir *j*, P_k is the amount of power introduced in the network by pump *k*, *γ* is specific weight of the water, Q_l is demand of tank *l* and H_l is head in tank *l*.

3. Average water age in the network over the last 24 hours of extended period simulation (*WA*):

$$WA = \frac{\sum_{i=1}^{n_j} \sum_{t=T-24}^{T} WA_i^t}{24 n_j} \tag{4.10}$$

Where WA_i^t is water age in junction *i* at time *t*. Water age is also often calculated as demand-weighted water age to give more significance to nodes with larger demands. In this research, equation (4.10) is used for *WA* calculation instead, in order to be comparable with other methodologies available in literature.

Other parameters calculated to aid evaluation of solutions are:

1. Number of DMAs (N_{DMA}), number of meters (N_M) and number of valves (N_V),
2. *NL* – Number of DMAs larger than maximum DMA size (d_{max}),
3. *NS* – Number of DMAs smaller than minimum DMA size (d_{min}),
4. A_{conn} – Average number of connections per DMA.

In addition to cost, PIs and parameters characterizing solution, listed above, for each DMA in a solution three following PIs are calculated:

1. $p_{DMA}^{av}, p_{DMA}^{min}, p_{DMA}^{max}$ – mean pressures over the 24 hours in a DMA, which can be a good indicator of potential leakage reduction benefits - average, minimum and maximum respectively calculated as:

$$p_{DMA} = \frac{\sum_{i=1}^{n_j} p_j}{n_j} \qquad \forall n_j \in DMA,\ \ p_j = p_{av_j}, p_{min_j} \text{ or } p_{max_j} \tag{4.11}$$

2. *Res_{DMA}* – Average resilience index for a DMA, calculated per equation (4.9), only this time accounting for nodes within considered DMA and
3. *WA_{DMA}* – Demand weighted *WA* for a DMA, averaged over entire extended period simulation. Demand weighting is used to account for difference of size between DMAs in terms of demand.

$$WA_{DMA} = \frac{\sum_{i=1}^{n_j}\sum_{t=1}^{T} WA_i^t q_i^t}{\sum_{i=1}^{n_j}\sum_{t=1}^{T} q_i^t} \qquad \forall n_j \in DMA \tag{4.12}$$

After the 3rd stage, sectorization algorithm's run is completed, resulting in set of feasible solutions. This is one of the main advantages of presented methodology, as it gives an array of alternative DMA designs to the decision maker. One can opt for a solution with large number of small DMAs or for a solution with small number of large DMAs, or anything in between. This is especially convenient for the analysis of large WDNs without previously established DMAs, where DMAs strategic planning should be addressed carefully. It is up to a decision maker to select sectorization solution best suitable to his preferences, based on calculated PIs and other parameters listed above. To aid the selection of preferable sectorization solution all feasible solutions can be plotted on two trade-off plots, to investigate how they behave against each other in terms of cost, water age and resilience (Figure 4.11). First plot should relate solution's cost to its resilience index, and the second plot would show a trade-off between solution's cost and water age. Figure 4.11, in which axis arrows indicate direction of increasing preference, illustrates that solutions Sol-2 and Sol-9 have the lowest implementation cost, but have different influence on network's operation as indicated by calculated PIs. Solution Sol-2 outperforms solution Sol-9 in terms of resilience index, but significantly more affects water age in the network.

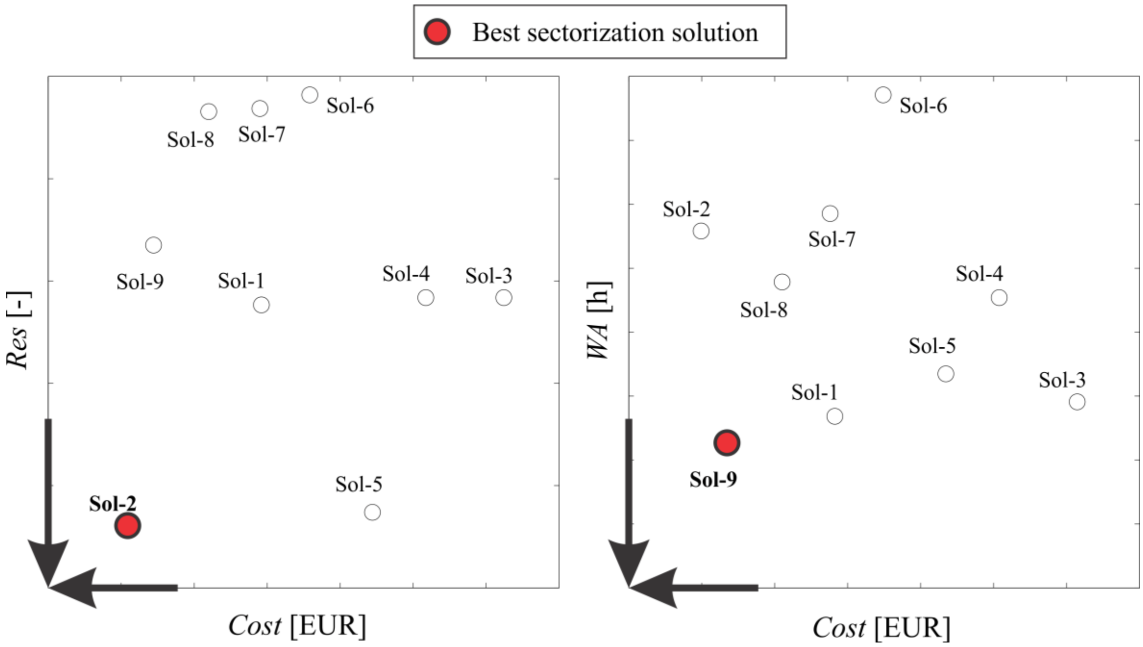


Figure 4.11: Trade-off plots aiding the selection of best feasible sectorization solution

4.2.5 Implementation of DeNSE algorithm

Presented methodology is implemented as per Figure 4.1. The 2nd Stage of the algorithm (Network clustering algorithm) is written in C++ programing language to ensure high computational efficiency. It is compiled as a dynamic link library (DLL) that can be used externally to perform clustering. For hydraulic simulations (in Step 2 of the 1st Stage and Step 2 of the 3rd Stage) EPANET DLL, modified to include TRIBAL-ΔQ method for hydraulic simulation (as shown in Figure 3.7 and described in section 3.3.2.2), is used. Using modified EPANET DLL, hydraulic simulations can be performed either using the GGA solver (already present in EPANET) or the improved ΔQ solver (added to EPANET source code).

4.3 Extensions of DeNSE sectorization algorithm

So far, this Chapter covered presentation of new algorithm for sectorization of WDN into DMAs, DeNSE, which introduced new uniformity index driving the sectorization process and some heuristic engineering criteria. DeNSE algorithm provides a set of feasible solutions, allowing decision maker to select one best suitable to his preferences. Extensive benchmarking results of DeNSE algorithm on a large real sized case study are given in the following chapter (Chapter 5), validating initially assumed working hypotheses. Aim of this section is to give an overview of other possible utilizations of DeNSE algorithm introduced here, and its upgrade in the extension of the work presented in this thesis.

4.3.1 Hierarchical sectorization of WDN with DeNSE algorithm

Chapter 2 covered the literature review on available algorithms for automatic sectorization of WDN into DMAs and benefits they provide over the traditional manual "trial and error" approach. Main advantage being ability to investigate wider specter of feasible solutions. However, none of the presented researches discussed reliability of WDN model data that is used as a main input. This is an important issue as reliability of model, supplied by the local water utility, can vary significantly. In developed countries water companies are usually efficiently managed and well organized, having access to reliable input data about networks' consumption, water losses, infrastructure (e.g. network pipeline and layout) etc. Significant measures have already been taken to tackle the water loss issue resulting in physical losses (e.g. leakages and pipe failures) being dominant in such WDNs. Prerequisite of having the detailed, well calibrated network model, with sufficient measuring data (e.g. pressures and flows), is usually fulfilled. On the other hand, WDNs in developing countries are faced with significant water losses, main portion of them being apparent losses such as systematic data handling errors, customer metering inaccuracies and illegal consumption. In addition to that, there is high level of uncertainty regarding the available network model and consumption data supplied by the local water utility.

Having said all of the above, main purpose of WDN's partitioning into DMAs has to be defined prior to sectorization itself, as it is specific to the system in consideration. For

WDNs with negligible apparent losses, detailed sectorization (i.e. into more smaller DMAs) makes sense, as it will allow easier identification of real losses and can be used additionally for better pressure management purposes. For WDNs with dominant apparent losses and unreliable input data, high resolution sectorization is not justified. Main goal should be adopting the sectorization solution that will enable tracking the water balance in the network and dealing with apparent losses, without endangering network reliability in terms of water supply and network pressures.

Prior discussion points to the need of hierarchical sectorization for WDNs with insufficient reliable input data, characterized with high water losses. Sectorization process should be carried out in phases, starting with a few DMAs that can be larger than size recommended by different guidelines. In the following stages, as the knowledge of the system increases and more reliable data is obtained, originally established DMAs can be partitioned further. With increased resolution of the sectorization, it is usually required that new DMAs keep previously created boundaries of the original DMA layout. In this manner economical aspect is addressed as this implies minimization of costs.

There is a single research that considered hierarchical sectorization of WDN (Scarpa et al., 2016). Methodology presented there is based on progressive union of initially identified elementary DMAs. This can be viewed as bottom-up approach. A top-down approach of sectorization would be closer to engineering perception and more in accordance with the phased creation of DMAs in practical cases explained in paragraph above. Top-down DMAs design approach can easily be carried out with DeNSE algorithm performing its recursive call. For illustration purpose of top-down hierarchical sectorization simple example network shown in Figure 4.12 is used.

Flow chart of the procedure required to create two level hierarchical partition is shown in Figure 4.13. Let’s recall the section 4.2.1 where necessary input data for DeNSE algorithm were given. Among others, listed there are: 1) minimum (n_c^{min}) and maximum (n_c^{max}) number of connections per DMA (e.g. n_c^{min}=500 and n_c^{max}=5000 per some guidelines), 2) total number of connections in the network (n_c) and 2) diameter threshold for transmission main (D_{main}). Additional data for hierarchical clustering would be desired number of DMAs for first hierarchical level solution (N_{zones}). As discussed earlier, first level solution should address the issue of tracking water balance in the network. Staff in charge of operating the WDN have extensive knowledge about the system and usually can provide a good estimate about the number of DMAs necessary for tracking the water balance. Since each WDN is specific, it makes sense to use this as an input parameter for first hierarchical level solution. Pipes connected in series, having diameter larger than 300-350 mm, are usually considered transmission main in the network. However, for first level sectorization solution, larger value for D_{main} should be used (e.g. D^{I}_{main} = 500 or 600 mm), as goal is to partition the network in relatively small number of DMAs that can contain larger number of connections than recommended value of n_c^{max}. Smaller value of D_{main} should be used for second level sectorization ($D^{II}_{main} < D^{I}_{main}$), as this is finer resolution sectorization.

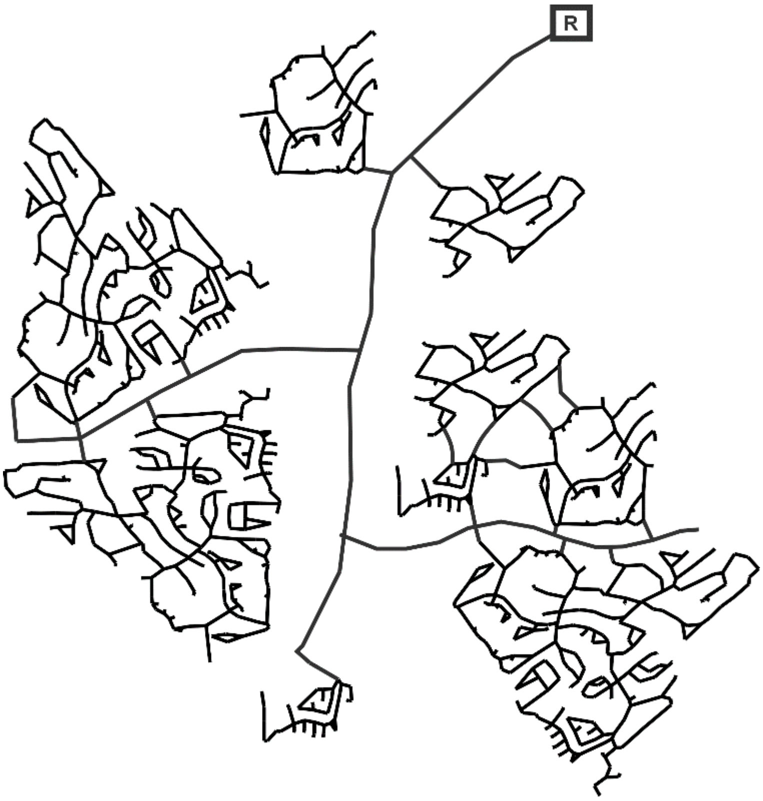


Figure 4.12: Example used to illustrate hierarchical zoning of the WDN

Procedure illustrated in Figure 4.13 starts by acquiring necessary input data for DeNSE algorithm, together with previously elaborated additional data (N_{zones}, D^{I}_{main} and D^{II}_{main}). In following step, expected number of connections in N_{zones} is calculated as:

$$n_c^{N_{zones}} = \frac{n_c}{N_{zones}} \tag{4.13}$$

which will definitely be larger than maximum recommended size for DMA, expressed through the number of connections ($n_c^{N_{zones}} > n_c^{max}$). Within first call of DeNSE algorithm, value $n_c^{N_{zones}}$ is used as upper limit for DMA size and D^{I}_{main} to identify the transmission main. Upper DMA size limit is used in phase 3 of the 2nd Stage of algorithm to define triangular function quantifying cluster size (see Figure 4.4). DeNSE algorithm, with all of its stages (see Figure 4.1), is run on the whole network resulting in a set of feasible solutions, from which user has to choose one. This completes first level sectorization and selected solution is regarded as 1st level hierarchical solution. For example network used here this solution is shown in Figure 4.14-a.

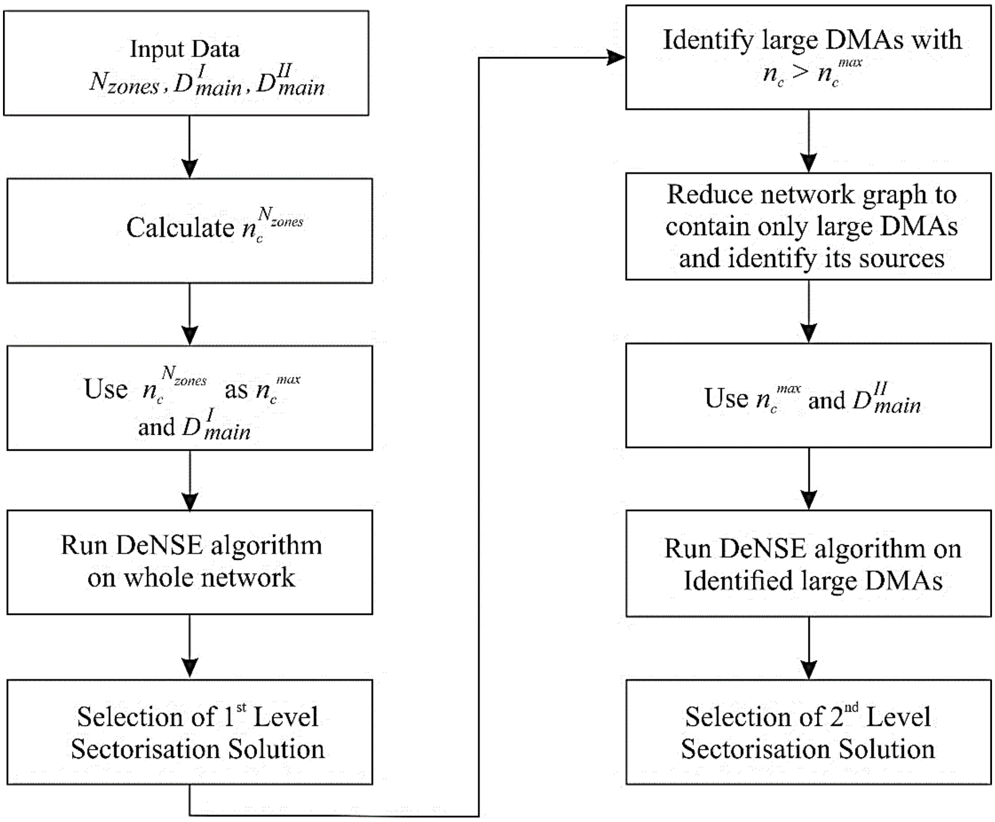
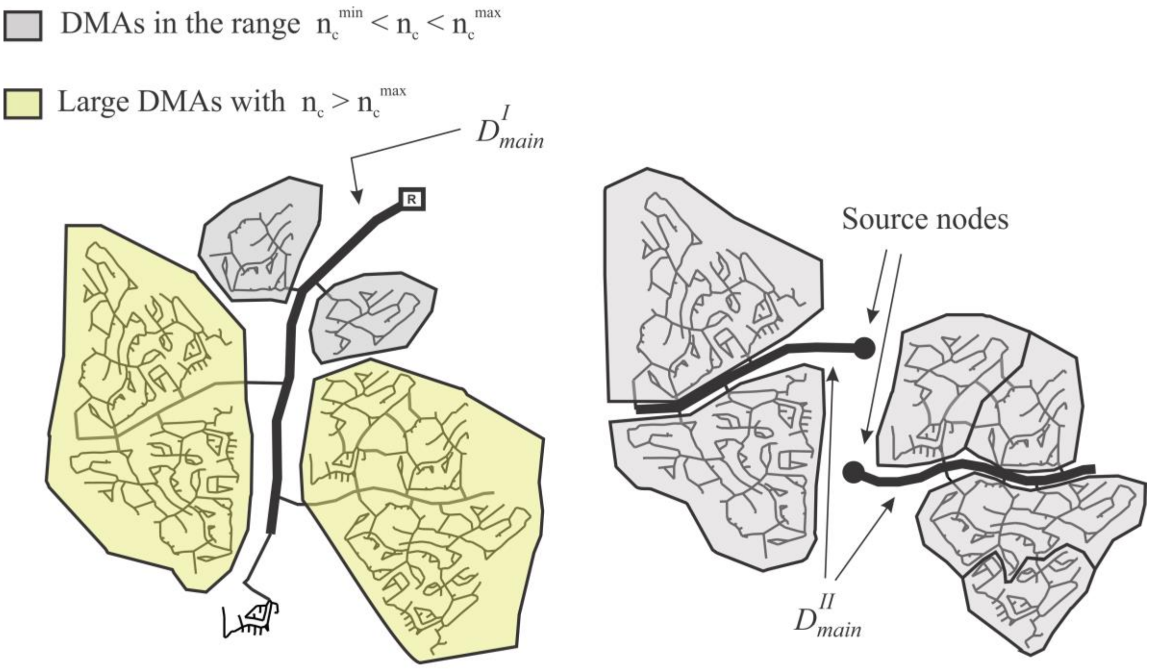


Figure 4.13: Flow chart of 2-level hierarchical sectorization procedure


DMAs in the range $n_c^{min} < n_c < n_c^{max}$
Large DMAs with $n_c > n_c^{max}$
D^{I}_{main}
R
Source nodes
D^{II}_{main}

a) b)

Figure 4.14: Hierarchical sectorization explained: a) first level; b) second level

As illustrated, 1st level solution will have some DMAs that are within DMA size constraints ($n_c^{min} < n_c \leq n_c^{max}$) and there is no need to partition them further. There will be also DMAs that are larger than recommended size ($n_c > n_c^{max}$), which are identified as candidates for second level sectorization. Original network graph is reduced to contain only large DMAs. As source nodes are required to obtain transmission main in network graph, and large DMA does not necessarily contain reservoirs or tanks, connecting points of each large DMA to the transmission main identified in first level are marked as source nodes. In second call to DeNSE algorithm value D^{II}_{main} is used as transmission main threshold and value n_c^{max} as upper limit for DMA size. Again, 2nd level hierarchical solution is selected from a set of feasible solutions. Results of second level sectorization are illustrated in Figure 4.14-b.

Finally, Figure 4.15 shows both sectorization solutions side by side – first and second level. First level solution has 4 DMAs and second level 8 DMAs in total, where 6 of them are derived from two large DMAs identified in first level sectorization while keeping the original two DMAs.

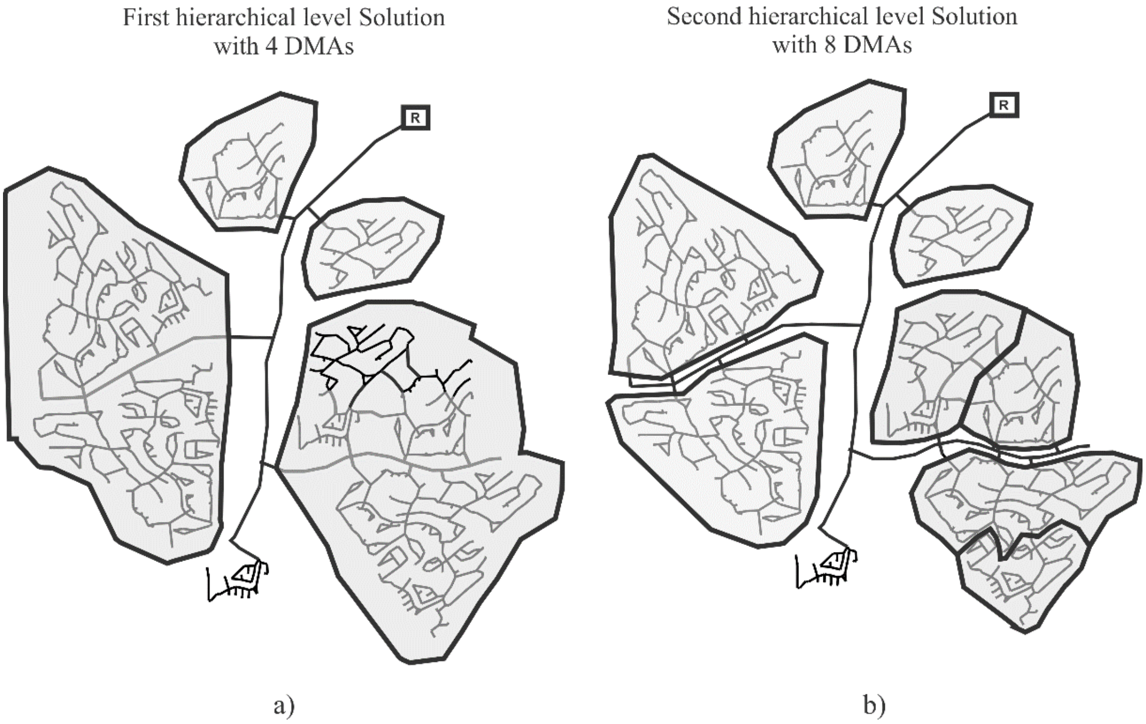

First hierarchical level Solution
with 4 DMAs
Second hierarchical level Solution
with 8 DMAs
R
R
a)
b)

Figure 4.15: Hierarchical sectorization: a) first level; b) second level

4.3.2 Coupling of DeNSE algorithm with an optimization method

Section 4.2.4 described 3rd Stage of the DeNSE algorithm, which converts clusters identified in the 2nd Stage into DMAs by positioning the flow meters and valves on their boundary edges. For this purpose, DeNSE algorithm uses new procedure based on common sense engineering heuristics. Criteria on which procedure is based are pipe diameters, orientation of the flow and maximum flow rates during the 24-h time period. Alternative approach to heuristic procedure, currently employed in DeNSE, is the use of an optimization method to determine the status of clusters' boundary pipes. Inside optimization algorithm each boundary pipe should be considered as opened or closed. Having in mind that for real WDNs there can be several dozens of boundary pipes, optimization method can be significantly time consuming. Reasoning this in particular, heuristic procedure was chosen over the optimization method as more computationally efficient and implemented in DeNSE. Heuristic procedure by no means implies optimal positioning of the flow meters and isolation valves. Main benefit of optimization methods over the heuristic procedure is the ability to investigate broader specter of feasible alternatives. Hence, there is still room for implementation of optimization methods in future development stages of methodology presented here.

Choice of suitable optimization method is not an easy one. Even though there are many information available in the literature regarding the optimization topic, it is easy to get lost among existing techniques and discrepancies in their description in different sources. Very brief overview of available optimization methods is given here, followed by the selection of one to be used in DeNSE algorithm and illustration of its implementation.

4.3.2.1 Available optimization methods

Figure 4.16 shows the most general division of optimization methods (Cavazzuti, 2013). Deterministic optimization assumes that there are no random elements appearing in optimization procedure. Synonym for deterministic optimization is gradient based optimization, as calculation relies on computation of objective function gradient. In the literature it is also referred to as mathematical programing, as this is the only optimization method accepted and used in the field of mathematical science. Stochastic optimization is directly opposite to deterministic, as randomness in the search procedure is allowed. Depending on a manner in which randomness is implemented in the optimization, different methods are available. Simulated Annealing, Particle Swarm Optimization and Game-Theory based Optimization are some of the methods belonging to stochastic optimization methods. Special subset of stochastic methods, probably the most important one and commonly used in the field of applied engineering, are evolutionary optimization methods correlating to Darwin’s evolution theory. Evolutionary optimization starts with a set of samples (population) evolving through combination of best performing individuals to generate an offspring, expected to have better performance. Combination of population’s individuals is done through bio-inspired processes of mutation, cross-over and selection. Genetic optimization is considered a special case of Evolutionary optimization. Input variables are discretized and coded into a binary string referred to as gene. Evolution of the population is influenced mainly by the cross-over process.

Deterministic optimization is by definition a single objective optimization. Stochastic optimization allows multiple objective functions to be defined, so it can be either single objective or multi-objective.

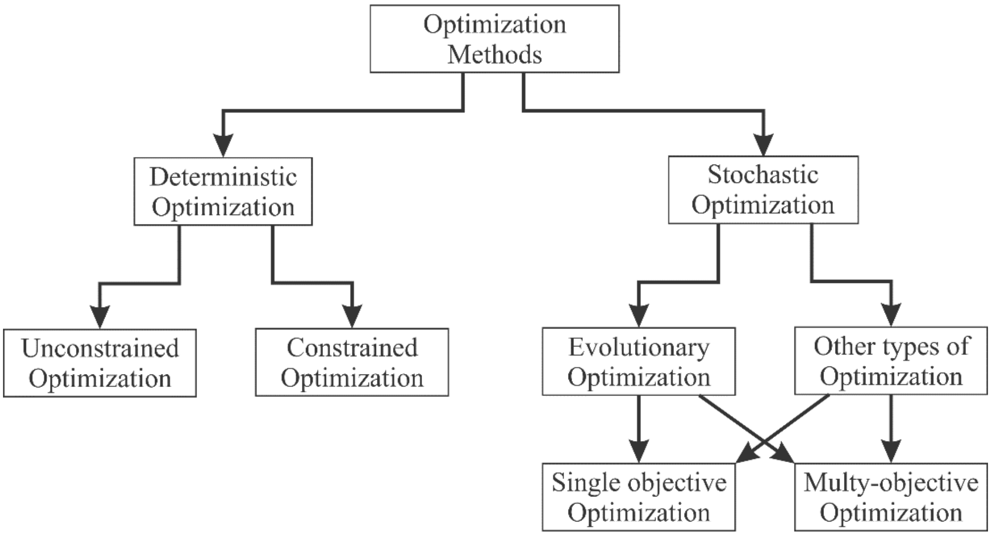

Optimization Methods
Deterministic Optimization
Stochastic Optimization
Unconstrained Optimization
Constrained Optimization
Evolutionary Optimization
Other types of Optimization
Single objective Optimization
Multy-objective Optimization

Figure 4.16: Hierarchical division of optimization methods

Deterministic optimization methods belong to local optimization methods, as they can get stuck in local minimum, coming from the fact that gradient based methods search for the stationary points in the objective function. Local optimization methods are very sensitive to the selection of the starting search point (Figure 4.17).

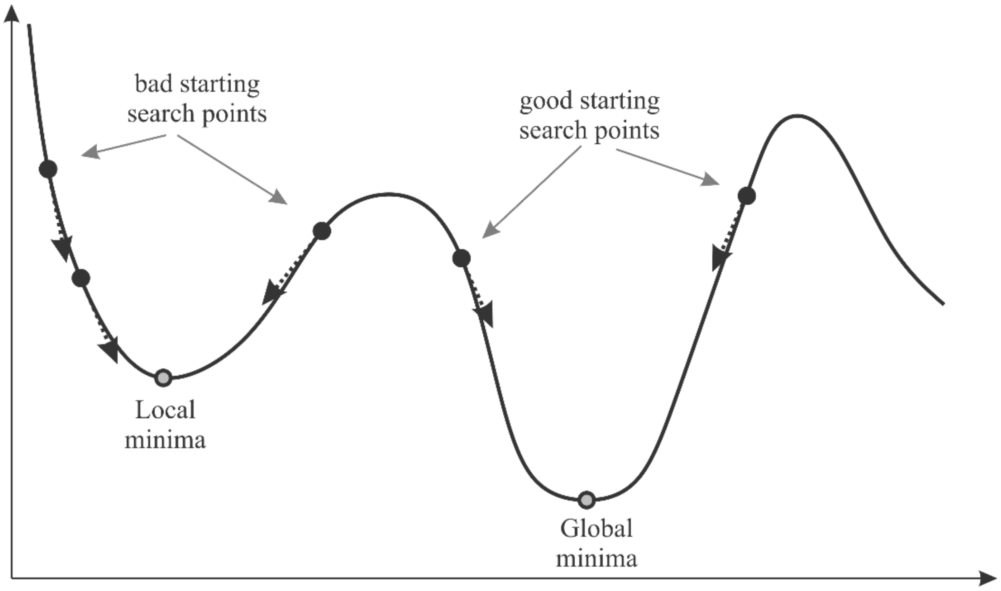


Figure 4.17: Deterministic optimization and local minima problem

Stochastic optimization methods can overcome local minimum problem as they are not based on gradient search. Hence, they are global optimization methods. Global optimization methods work on a set of solutions, and even though the finding of the global optima is more likely than in the case of local methods, it is not guaranteed.

Both deterministic and stochastic optimization methods can be constrained or unconstrained. For deterministic methods this is more important as unconstrained optimization is fairly simple, while taking constraints into consideration problem becomes much more difficult to solve. Some of unconstrained deterministic optimization methods are Linear programming, Quadratic programming and Nonlinear programming.

Comparing deterministic and stochastic optimization methods, both approaches have advantages and drawbacks. Convergence to a solution is generally much faster with deterministic methods. Being based on rigorous mathematical relations without stochastic elements, results are irrefutable and replicable. However, problems of stacking in local minimum and poor convergence in areas with small gradients cannot be neglected. Most importantly, problems involving multiple objective functions must be subjected to stochastic optimization.

4.3.2.2 Implementation of GA in DeNSE algorithm

Based on the discussion made above, genetic algorithm (GA) is chosen as the best suitable optimization method to replace the heuristic procedure for positioning of the flow meters and isolation valves in the 3rd Stage of DeNSE algorithm. Flow charts of both procedures, currently implemented one and proposed one employing GA, are shown next to each other in Figure 4.18 for comparison. Proposed procedure with implementation of GA is now explained, followed by the discussion on its benefits.

Following the 2nd Stage of DeNSE, in which clustering of the network is done, several clustering solutions are selected to enter the 3rd stage which involves placement of the flow meters and valves on boundary edges. Currently employed heuristic procedure is carried out in three steps, explained in detail in section 4.2.4. Proposed, GA based procedure keeps one of these steps – step in which boundary edges that always return water from the clusters to the transmission main are closed. This is executed prior to the

GA itself. These pipes are not on supply paths, and as such can be considered redundant and closed without the effect on system's reliability. Additionally, this reduces the solution search space for the follow on GA. Implementation of GA will be explained using the simple example already used to illustrate heuristic procedure (Figure 4.19).

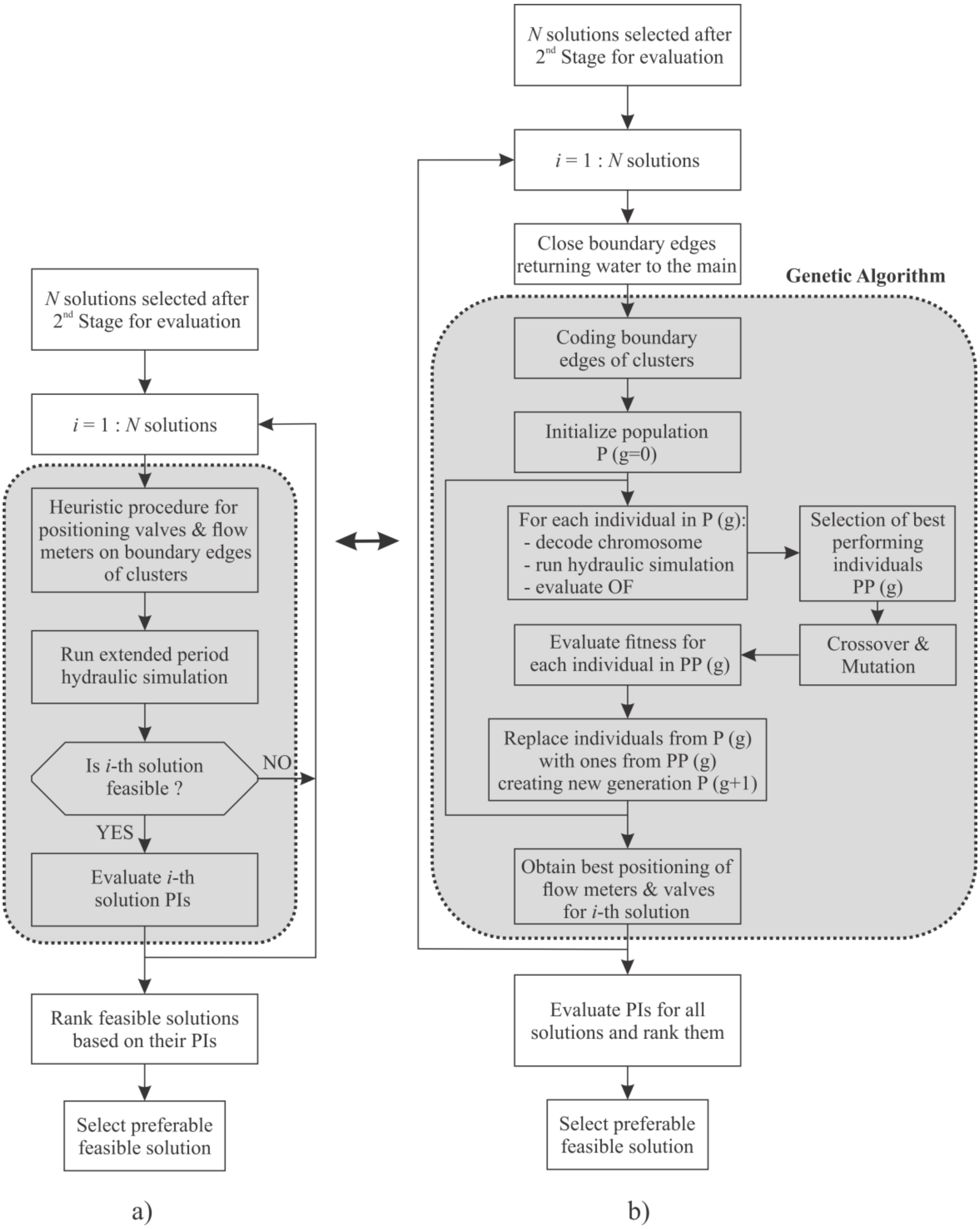

N solutions selected after 2nd Stage for evaluation
i = 1 : N solutions
Heuristic procedure for positioning valves & flow meters on boundary edges of clusters
Run extended period hydraulic simulation
Is i-th solution feasible ?
NO
YES
Evaluate i-th solution PIs
Rank feasible solutions based on their PIs
Select preferable feasible solution
a)
N solutions selected after 2nd Stage for evaluation
i = 1 : N solutions
Close boundary edges returning water to the main
Genetic Algorithm
Coding boundary edges of clusters
Initialize population P (g=0)
For each individual in P (g): - decode chromosome - run hydraulic simulation - evaluate OF
Selection of best performing individuals PP (g)
Crossover & Mutation
Evaluate fitness for each individual in PP (g)
Replace individuals from P (g) with ones from PP (g) creating new generation P (g+1)
Obtain best positioning of flow meters & valves for i-th solution
Evaluate PIs for all solutions and rank them
Select preferable feasible solution
b)

Figure 4.18: 3rd Stage of DeNSE algorithm: a) heuristic based; b) proposed – GA based

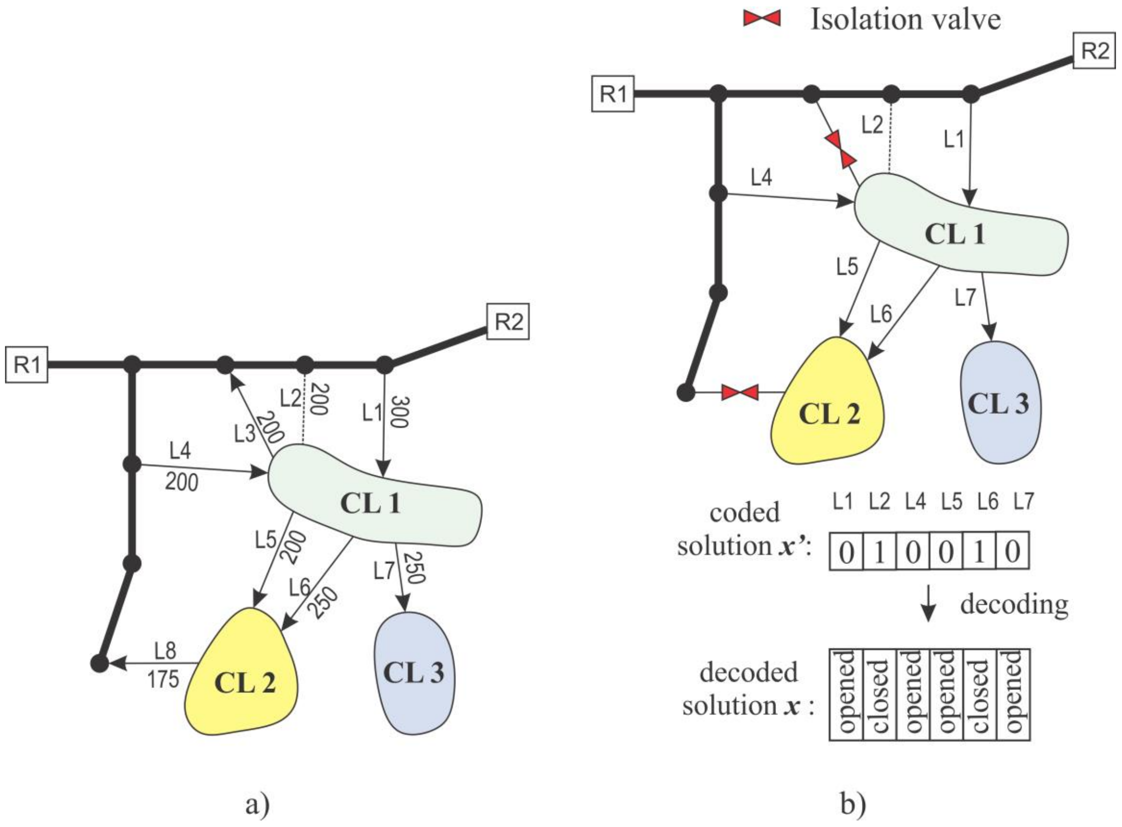


Figure 4.19: Simple example illustrating implementation of GA: a) one clustering solution after the 2nd Stage; b) coding of the remaining boundary edges

Entering the 3rd stage, illustrative clustering solution in Figure 4.19-a has 3 clusters with 8 boundary edges. After the closure of ones always returning water to the main (L3 and L8), there are 6 remaining pipes whose status should be determined. Pipe is either closed by placement of isolation valve, or it remains open and is equipped with flow meter. That being said, pipe status is the only independent variable taking one of two values – opened or closed. In GA, solutions are coded into chromosomes represented with a string of bits (Figure 4.19-b). Parts of that string are coded variables (genes). Number of genes equals the number of pipes with unknown statuses. String of 1 bit is sufficient for representation of each gene, as there are only two possibilities for the status of the pipe (e.g. 1 – closed or 0 – opened). After length of chromosomes is determined, population containing *m* individuals is initialized ($P(g=0)$) and its evolution process through generations begins, employing main GA's operations (Figure 4.20). First step is to decode each solution from generation, run hydraulic simulation and evaluate its objective function (OF). Next step is the selection of best performing individuals – parents to generate offsprings. Selection process is responsible for controlled stochastic behavior of GA and it can follow different rules such as roulette-wheel or tournament selection. It means that randomness of the selection process is biased by the fitness of the objective function in the way that better solutions have better chance to be chosen (selected) and take part in the crossover. After the 2 parents are selected (x_i'' & x_j'' in Figure 4.20), they exchange their coded material at the randomly selected point (also called crossover point) and the new coded solutions are created (x_i'' & x_j''). Crossover is usually implemented with probability which is close to 1.0 (0.8 - 0.9 are preferred values), and it is one of the parameters of the GA. Mutation operator is implemented by altering randomly picked bit in the coded solution from 1 to 0 or vice versa.

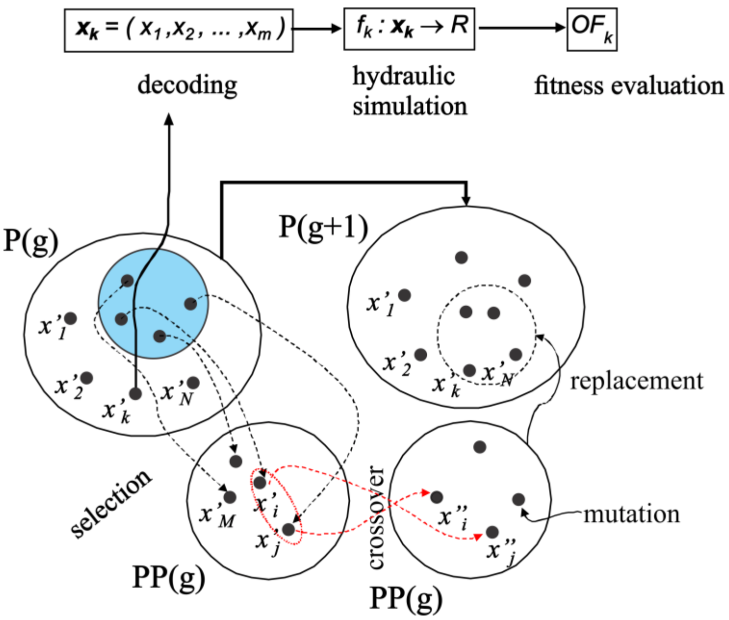


Figure 4.20: GA evolution process

Crossover and mutation operations are illustrated in Figure 4.21. Probability of implementing mutation is also the GA parameter and generally should be quite low (0.01) since the aim of GA is to be driven by crossover rather than mutation. By implementing selection, crossover and mutation, the new set of usually better solutions is created (new generation – P(g+1)), and the whole process is now repeated (decoding, evaluation, selection, crossover, mutation) until the maximum number of generations is reached.

Finally, summarizing described GA method, parameters that have to be defined for its application are listed:

- Population size – number of individual solutions in population,
- Chromosome length – number of bits coding one solution,
- Crossover probability – usually 0.8 – 0.9,
- Mutation probability – usually < 0.05 and
- Number of generations to evolve.

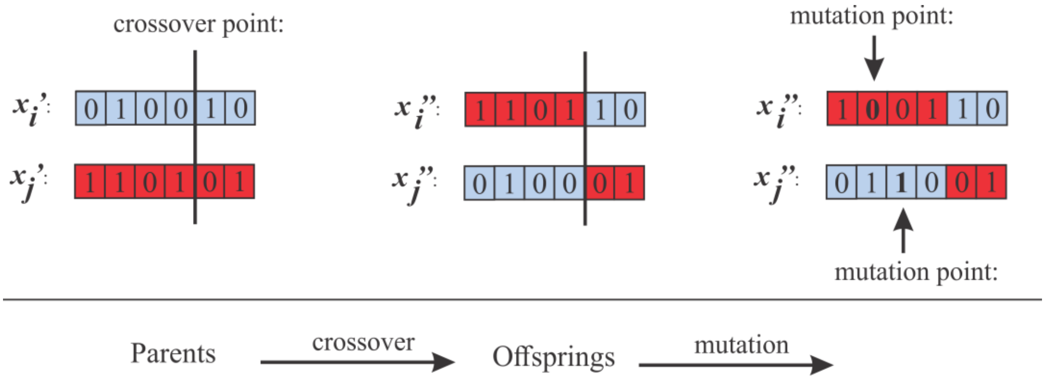


Figure 4.21: Crossover and mutation operations

The efficiency of GA will depend on the adopted values for above listed parameters. Proposed values are just a suggestion, since different objective functions will require different set of values for parameters to achieve the same efficiency. Objective function for proposed implementation of the GA considers only economical aspect – solution cost. The informal definition of GA could be that it is optimization method that searches for optimum solution in discrete multidimensional space without constraints. Network sectorization problem is constrained with the request that any implemented interventions do not endanger network's operating reliability, providing feasible sectorization solution. In DeNSE algorithm's methodology, feasibility of the solution is assessed through evaluation of pressure constraints given by equation (4.8). The only way to impose constraints in the basic form of GA is by using penalty function within OF. Objective function is defined as:

$$\mathrm{OF} = \mathrm{Cos}t + \sum_{i=1}^{n_j} \left(\overbrace{\max\left(0, \Delta p^{*}_{i,\min} - \Delta p_{i,\min}\right)}^{\textit{penalty for lowered pressures}} + \overbrace{\max\left(0, \Delta p^{*}_{i,\max} - \Delta p_{i,\max}\right)}^{\textit{penalty for increased pressures}} \right) C_p \quad (4.14)$$

where $\Delta p^{*}_{i,\min} = \max\left(0, p_{\min} - p^{*}_{i}\right)$ is pressure deficit in *i*-th junction after network interventions and $\Delta p_{i,\min} = \max\left(0, p_{\min} - p_{i}\right)$ is pressure deficit prior to any network interventions. Same goes for pressure surplus in junction: $\Delta p^{*}_{i,\max} = \max\left(0, p^{*}_{i} - p_{\max}\right)$ is pressure surplus in *i*-th junction after network interventions and $\Delta p_{i,\max} = \max\left(0, p_{i} - p_{\max}\right)$ is pressure surplus prior to any network interventions. Definition of OF given with the equation above penalizes only those solutions that worsen the network pressures, i.e. increase them or lower them, compared to the pressures in original state of the network. *Cost* is price of the solution calculated as before, based on unit cost functions for installed devices given in Figure 4.10. C_p is penalty cost.

Comparing the two approaches for positioning of the flow meters and isolation valves, shown in Figure 4.18, following concluding remarks can be made:

1. Heuristic based method, currently implemented in DeNSE algorithm, requires far less hydraulic simulations than GA based approach. To be more precise, hydraulic simulation is performed *N* times (once per clustering solution). In the GA based approach hydraulic simulation is performed multiple times, due to its iterative evolution process. Considering this, heuristic based method is expected to be more computationally efficient.
2. GA based method is global optimization method searching optimal solution within a wide set of possible alternatives for positioning DMA isolation devices. Finding the global optima solution is not guaranteed, but it is expected that at least better local optima solution can be identified with GA based method, compared to the heuristic one.
3. Here proposed GA based method involves one heuristic step borrowed from the original heuristic method. It closes some pipes prior to GA itself, reducing its

solution search space, compared to the conventional GA implementation which would investigate all boundary edges.

It can be concluded that both approaches have their benefits and drawbacks. Computational efficiency of the GA based method, being its main drawback, can be significantly improved if DeNSE sectorization algorithm is coupled with TRIBAL-ΔQ hydraulic solver.

CHAPTER 5:

CASE STUDIES

5 CASE STUDIES

Table of contents:

5.1 INTRODUCTION

Following the Chapters 3 and 4, in which TRIBAL – ΔQ method for hydraulic simulation and DeNSE sectorization algorithm were presented, this chapter presents their benchmarking results on selected case studies. Section 5.2 presents benchmarking results of new TRIBAL – ΔQ method, which is tested on four case study networks of different complexities (in terms of topology and element types e.g. valves, pumps and tanks). Performance of improved ΔQ hydraulic solver is compared with the reference

GGA solver in terms of convergence, efficiency and accuracy. Advantages of the TRIBAL algorithm for identifying loops are investigated through its comparison with other loops identification procedures. Computational efficiency of the hydraulic solver implemented in TRIBAL-ΔQ method is also compared to hydraulic solvers used in other methods available in the literature. Presented results are published in Vasilić et al. (2018),

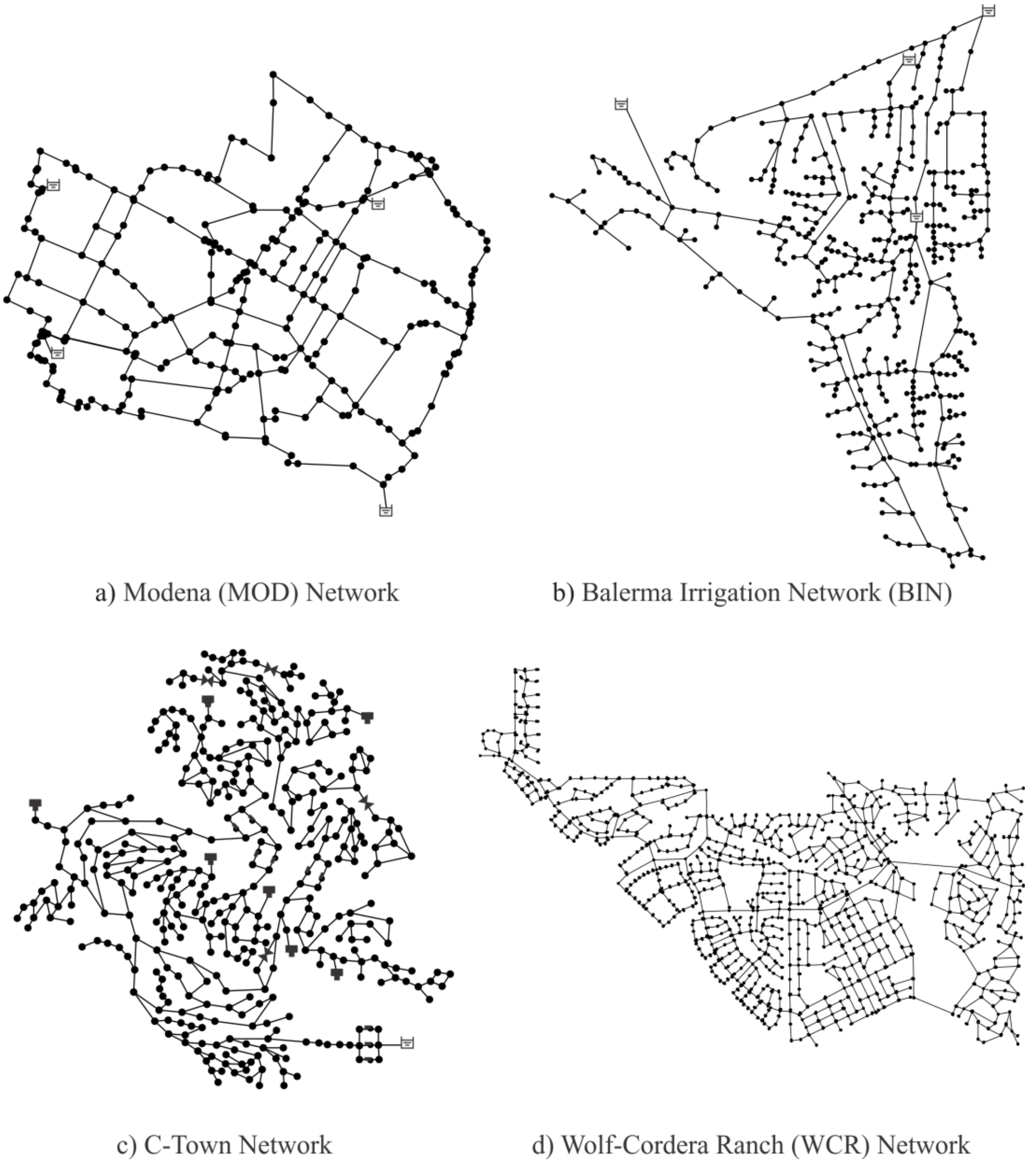
Benchmarking results for the new distribution network sectorization algorithm (DeNSE) are reported in Section 5.3. Large distribution network BWSN2 (Ostfeld et al., 2008), well known and often used in the literature for various modeling tasks, served as a case study network. As discussed in Chapter 4, DeNSE algorithm performs automatic clustering of WDN and provides a set of feasible sectorization solutions. Best solution is not selected by the algorithm itself, as this is a subjective decision, but it is up to a decision maker to select the one best suitable to his preferences. Resulting feasible solutions are discussed and selection of preferable solution is made. Results of DeNSE algorithm are compared to other sectorization algorithms available in the literature that also used BWSN2 network as a case study, in order to assess its performance.

5.2 TRIBAL – ΔQ METHOD RESULTS

5.2.1 Case study networks

Four different example networks are used to test the new TRIBAL-ΔQ method, validate its accuracy and compare it to the EPANET's original solver based on the GGA (see Figure 5.1). EPANET input data for the networks Modena (MOD), Balerma Irrigation Network (BIN) and Wolf Cordera Ranch network (WCR) can be found at http://emps.exeter.ac.uk/engineering/research/cws/resources/benchmarks/, while C – Town Network input data is available at http://www.water-simulation.com/wsp/about/bwcn/.

As it can be seen from Figure 5.1, the example networks used here are very different in terms of topology and number and type of network elements. Networks main characteristics are summarized in Table 5.1.



a) Modena (MOD) Network

b) Balerma Irrigation Network (BIN)

c) C-Town Network

d) Wolf-Cordera Ranch (WCR) Network

Figure 5.1 Case study networks used for TRIBAL-ΔQ method testing

Table 5.1 Characteristics of case study networks

Network	N_n	N_l	N_L	$N_{t/r}$	N_p	N_v	***Links in loops***	L_{factor}
MOD	272	317	49	4	0	0	317	1.00
BIN	447	454	11	4	0	0	162	0.36
C-town	396	444	56	8	11	4	289	0.65
WCR	1786	1995	213	4	6	4	1173	0.59

* N_n-number of nodes; N_l-number of links; N_L-number of loops; $N_{t/r}$-number of tanks and reservoirs; N_p-number of pumps; N_v-number of valves

Parameter L_{factor} is introduced in order to express networks topology in terms of how looped it is. L_{factor} is defined as ratio of number of links that are part of loops and total number of links in the network. L_{factor} value is ranged between 0, if network has no loops, and 1 if all links are part of at least one loop.

$$L_{factor} = \frac{Links\ in\ loops}{N_l} \tag{5.1}$$

Note from Table 5.1 that the MOD network is extremely looped with all links belonging to at least one loop – there are no branched parts.

5.2.2 Comparison criteria

The following criteria are used when comparing the TRIBAL-ΔQ and GGA methods on above four networks:

1. Computational efficiency (i.e. speed). This was assessed by the computational time required to perform hydraulic analysis. The focus was on the comparison of performances of two solvers (ΔQ and GGA) for solving the steady state WDN hydraulics (i.e. 2nd Block in Figure 3.7). Performances were analyzed for several different target accuracies. Computational time required for the TRIBAL algorithm to identify network loops is reported separately.

2. Convergence. This was assessed with the number of iterations required for each algorithm to converge to a stable numerical solution. As above, this was done for several different target accuracies.

3. Prediction accuracy. This was assessed by comparing the average and maximum differences between pressure and flows predicted by the two methods. When comparing the flows, pipes in which velocities are less than 0.05 m/s are excluded from the analysis. This was done to avoid high relative errors for pipes with almost no flow.

5.2.3 Results and discussion

This section presents and discusses the results obtained from comparison of the proposed TRIBAL-ΔQ method and the GGA method, both implemented in EPANET.

Presentation and discussion of the results is made in above listed order of comparison criteria. Presented results are published in Vasilić et al. (2018).

5.2.3.1 Computational efficiency

Computational time for the loops identification algorithm (TRIBAL – i.e. 1st Block in Figure 3.7) is reported first. This time was less than a second for all case studies except for the WCR one, where the time required to complete network preprocessing was approximately 5 seconds. To further compare things, the algorithm for loops identification developed by Creaco & Franchini (2014) was compared to the TRIBAL algorithm proposed here on the highly looped generic network consisting of 120 nodes (Case study 1, Network 4) presented in Creaco & Franchini (2014). This network was used for comparison since both loops identification algorithms are implemented in the same environment (Matlab), and were run on a PC with similar characteristics. Creaco & Franchini (2014) reported time of 1.24 sec and the TRIBAL method presented here took 0.32 sec. This implies that, even in the highly looped/complex networks, the TRIBAL algorithm is reasonably fast and this can only improve if implemented in a more efficient programming environment (e.g. using the C language).

The comparison of the ΔQ and GGA solvers performances was done in terms of computational time required to reach target accuracy (all done as part of 2nd Block calculations). Because computational time of both algorithms for all considered case studies is generally very short, computational time in all cases was estimated for 10,000 cumulative algorithm runs. This series of runs was repeated 10 times and mean time is reported here in all figures and tables shown below.

Total computational time for both solvers and for different target accuracies (*eps*) is shown in the Table 5.2.

For easier comparison of computational time, speedup factors and relative time savings are calculated as follows:

$$\begin{aligned} Speedup\_factor &= \frac{t(\mathrm{GGA})}{t(\Delta Q)} && [-] \\ t\_savings &= \frac{t(\mathrm{GGA}) - t(\Delta Q)}{t(\mathrm{GGA})} 100 && [\%] \end{aligned} \tag{5.2}$$

The obtained values of the *speedup_factor* and *t_savings* are shown in Figure 5.2.

Table 5.2 Computational time in seconds for different target accuracies (*eps*)

eps	**Solver**	**Network**			
		MOD	**BIN**	**C-TOWN**	**WCR**
10^{-3}	GGA	1.91	4.38	2.33	13.43
	ΔQ	1.30	1.22	1.45	6.45
10^{-4}	GGA	1.91	4.40	2.89	not.av.
	ΔQ	1.46	1.33	1.60	7.25
10^{-6}	GGA	2.24	5.18	4.21	not.av.
	ΔQ	1.57	1.47	1.81	8.10

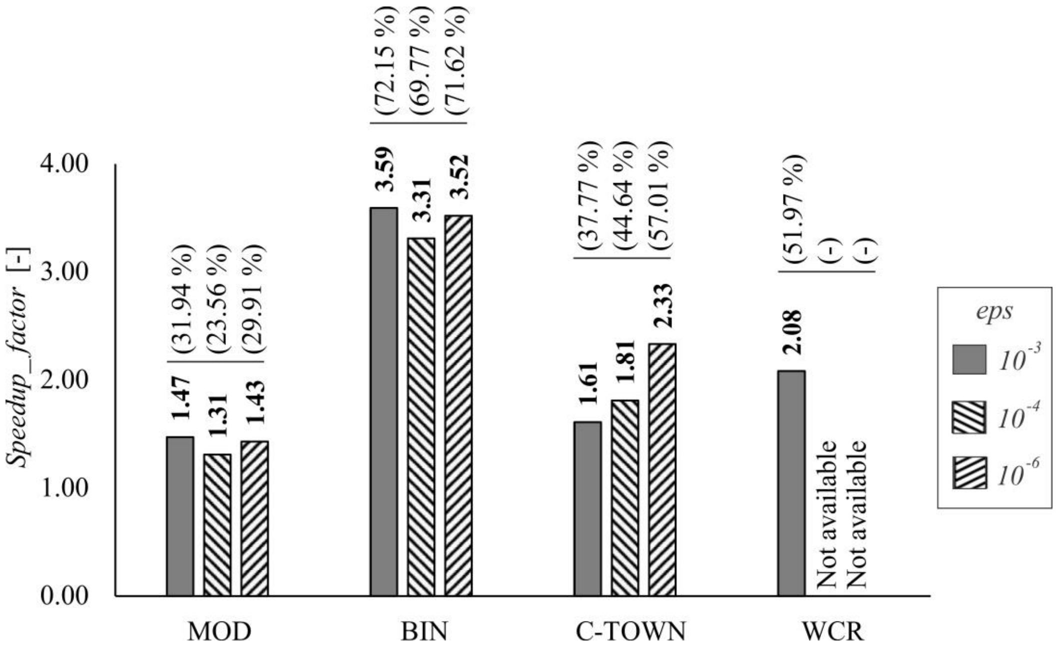


Figure 5.2 Speedup factors and relative time savings obtained with the ΔQ solver compared to the GGA solver

As it can be seen from Figure 5.2, the obtained speedup factors are in the range of 1.30 – 3.59 or expressed in relative time savings the values are 23.56% - 72.15% (values shown in brackets on the graph), all in favour of the ΔQ solver. As expected, the largest speedup was achieved for the BIN network which has only 11 loops and the loop factor value of 0.36. When compared to the speedups achieved in the literature these factors appear very encouraging (Table 5.3). For example, in Simpson et al. (2014) where FCPA method was compared to the GGA, reported speedups ranged from 1.11 to

1.31, while in Alvarruiz & Vidal (2015) the corresponding speedup factors (proposed loop method vs GGA) were in the range of 1.13-1.32 for networks tested there. In that research it was noted that solving the system of equations is actually up to 5 times faster (in favor of proposed loop method), but the calculation weight of estimating new coefficients is up to 60% which significantly influenced overall speedups per iteration. One should have in mind that the speedup factor value is dependent on the networks analyzed (especially in terms of topology and complexity), hence above values are indicative only, i.e. not directly comparable. Having said this, note that in Simpson et al. (2014), the WCR network tested here was also tested there (denoted as *N3*) and the achieved speedup factor, expressed in terms of time savings of FCPA over the GGA method was 14.5%. In this research, the corresponding time saving obtained for the WCR network is 52.0% (or speedup factor value of 2.08, see Figure 5.2) when Δ*Q* and GGA solvers are compared. This, obviously, represents a significant improvement. This comparison also points out advantage of the Δ*Q* solver since the 37.5% savings (= 52.0% – 14.5%) can be accredited to the efficiency of the Δ*Q* solver and not to the fact that the WCR network has substantial amount of branched parts. Finally, it should be noted that the WCR network was also benchmarked in Elhay et al. (2014) by comparing the RCTM and GGA solvers, resulting in a speedup factor of 1.50 (in favor of the RCTM method), while here this factor is 2.08 (in favor of the Δ*Q* solver).

Table 5.3 Computational efficiency of different methods compared to the reference implementation of GGA in EPANET

		Overall reported	**WCR Network**	
published in	**Method**	*Speedup_factor* (-)	*t_savings* (%)	*Speedup_factor* (-)
Simpson et al. (2014)	FCPA	1.11-1.31	14.5	NA
Elhay et al. (2014)	RCTM	1.15-1.84	NA	1.5
Alvarruiz & Vidal (2015)	Loop-flow	1.13-1.31	NA	NA
Vasilić et al. (2018)	TRIBAL-Δ*Q*	1.31-3.59	52	2.08

To further investigate the differences obtained in computational speeds, Table 5.4 illustrates advantages of the TRIBAL algorithm for identifying loops and how it reflects to the sparsity of resulting linear system to be solved. This is done through the comparison of number of non-zero elements (NZE) in the Cholesky factor of the

Jacobian matrix. Comparison is made between the following solvers: 1) GGA based solver, 2) loop solver that uses arbitrary set of loops (ASL-Δ*Q*) and 3) here proposed loop solver based on TRIBAL-Δ*Q* methodology. Arbitrary set of loops and its corresponding loops matrix $\mathbf{M_0}$ (Piller, 1995) is determined by running the BFS algorithm from randomly selected source node to obtain the tree, with remaining links forming the loops. Results show that significantly lower number of non-zero elements is obtained with the TRIBAL-Δ*Q* method when compared to the GGA, thus resulting in improved computational time. Also, TRIBAL-Δ*Q* method based solver yields notably lower number of NZE for all networks in consideration, compared to the ASL-Δ*Q*, highlighting the importance of search for the minimal loops.

Table 5.4 Comparison of the GGA and loop based solvers in terms of linear system sparsity expressed through number of non-zero elements (NZE) in the Cholesky factor of the Jacobian matrix

Network	**Solver**		
	GGA	**ASL-Δ*Q***	**TRIBAL-Δ*Q***
MOD	958	445	280
BIN	1039	29	27
C-TOWN	1034	213	187
WCR	5021	1959	899

Table 5.5 depicts values of speedup factors for two different approaches of updating the network links coefficients (for networks BIN, C-TOWN and WCR and for target accuracy of $eps=10^{-3}$). In case 1, all network links are updated, i.e. as it is implemented in EPANET, while in case 2 only links that are part of loops are updated, i.e. as implemented in this research. This has been done in order to illustrate the effect of updating coefficients only for the pipes that belong to the loops (as opposed to all pipes in the network). The MOD network is excluded from this analysis as all of its pipes belong to at least one loop (L_{factor} = 1), so all the links have to be updated anyway. The effect of updating different pipes is most pronounced for the BIN network for which the relative speed factor increase of 45.34% was obtained (case 2 to case 1). This is expected as this network has most branched parts. For other two networks, which have

higher L_{factor}, the percentages obtained (13.38% and 9.47%) are not as high but certainly are not insignificant. Based on the results presented it can be concluded that, for the four networks tested, the Δ*Q* hydraulic solver is considerably computationally faster than the GGA solver.

Table 5.5 Values of speedup factors for two different approaches of updating links' coefficients

links update approach	**BIN**	**C-TOWN**	**WCR**
Case 1: update all links in network	2.47	1.42	1.90
Case 2: update only links in loops	3.59	1.61	2.08
Relative increase (case 2 to 1, %)	45.34	13.38	9.47

5.2.3.2 Convergence

The comparison of the Δ*Q* and GGA solvers in terms of number of iterations required to converge to target accuracy of *eps* = 0.001, time required (per iteration) to converge for the same target accuracy and the corresponding speedup factors are presented in Table 5.6.

Table 5.6 Number of iterations, calculation time per iteration and speedup factors for *eps*=0.001

		Network			
		MOD	**BIN**	**C-TOWN**	**WCR**
total time	GGA	1.91	4.38	2.33	13.43
(s)	Δ*Q*	1.30	1.22	1.45	6.45
num of it	GGA	5	6	5	6
(-)	Δ*Q*	7	6	7	7
time per it	GGA	0.038	0.073	0.047	0.224
(ms)	Δ*Q*	0.019	0.020	0.021	0.092
speedup per it		2.06	3.59	2.25	2.43

As it can be seen from Table 5.6, both TRIBAL-Δ*Q* and GGA based hydraulic solvers converged to a stable numerical solution in all 4 examples (hence results available in all cases) for target accuracy of 10^{-3}. However, unlike the TRIBAL-Δ*Q* method based solver, in the case of a large WCR network, GGA solver becomes unbalanced for accuracies larger than 10^{-3}.

Table 5.6 also shows that GGA solver usually reaches the target accuracy in smaller number of iterations than ΔQ solver. However, as already noted in the previous section, the increased number of iterations for the ΔQ solver still pays off in term of significant reduction of computational time for all investigated networks. Similar results are obtained for other two values of target accuracy eps (10^{-4} and 10^{-6} not presented here to save space). This difference in number of iterations required to converge is most likely due to the different initial flows used in two methods (Alvarruiz & Vidal, 2015) as the ΔQ method needs to satisfy the mass balance equations at all nodes and the GGA methods does not. However, in some cases (BIN for accuracy of 10^{-3} and C-town for accuracy of 10^{-6}) both solvers require the same number of iterations (BIN=6 iterations; C-TOWN=9 iterations). Comparing the two solvers in terms of calculation time per iteration, it is clear that speedup factors are even higher than for the corresponding total run time factors reported above.

5.2.3.3 Prediction accuracy

The differences in predicted pressures and flows using TRIBAL-ΔQ and GGA based solvers, for target accuracy *eps* = 10^{-4}, are shown in Table 5.7.

Table 5.7 Differences in pressures and flows predicted by TRIBAL-ΔQ and GGA method based solvers for target accuracy *eps* = 10^{-4}

criteria	**MOD**	**BIN**	**C-TOWN**
Max difference in predicted pressures (m)	0.010 (0.017 %)	0.026 (0.022 %)	0.410 (0.283 %)
Average absolute difference in predicted pressures (m)	0.004 (0.006 %)	0.007 (0.007 %)	5 x 10^{-5} (4 x 10^{-5} %)
Max difference in predicted flows (L/s)	5 x 10^{-5} (0.008 %)	0.067 (0.022 %)	0.014 (0.957 %)
Average absolute difference in predicted flows (L/s)	4 x 10^{-6} (0.001 %)	0.002 (0.022 %)	0.001 (0.002 %)

Results are not available for the WCR network as GGA is unstable for considered target accuracy. Values are expressed in absolute and relative terms (shown in %) – relative to the GGA solution. As it can be seen from this table, predictions of the two solvers are virtually identical demonstrating high prediction accuracy.

5.2.4 Summary

The benchmarking of TRIBAL-Δ*Q* and GGA is performed on four large networks of varying topology and different complexity. The comparison criteria used is comprised of computational efficiency, convergence and prediction accuracy of two hydraulic analysis methods and solvers analyzed. The results obtained lead to the following conclusions:

1. The TRIBAL-Δ*Q* method based hydraulic solver is substantially computationally faster than the GGA based hydraulic solver. This significant speedup is a result of: (a) the application of new, computationally efficient TRIBAL algorithm that is able to identify network loops in a way which results in a highly sparse solution matrix (which, in turn, requires less computational time to be inverted and generally numerically manipulated in steady-state hydraulic calculations), (b) the fact that the improved Δ*Q* solver updates relevant coefficients only for the links that are in the loops and (c) efficient implementation of new data structures for networks loops and spanning tree into the EPANET software code.
2. The TRIBAL-Δ*Q* method based hydraulic solver showed stable numerical performance by converging successfully when performing hydraulic analysis in all four pipe networks analyzed and for all three target accuracies used. The GGA based hydraulic solver showed stable numerical convergence in most cases but failed to converge in the case of a large WCR network for two highest target accuracies. The TRIBAL-Δ*Q* method based solver usually requires more iterations to converge than the GGA based solver (for a given target accuracy) but this does not have an impact on the overall computational speed, quite the opposite.

3. With regard to the prediction accuracy, both TRIBAL-ΔQ and GGA based hydraulic solvers demonstrated ability to accurately predict pressures and flows in all four network analyzed.

Given the above conclusions and detailed results presented in the paper, the use of the TRIBAL-ΔQ method based solver for networks seem to be particularly well suited for pipe networks with substantial branched parts. The TRIBAL-ΔQ is preferred method in optimization and similar type problems (e.g. reliability analysis) where multiple, repetitive hydraulic simulation are required without modifying the network configuration (i.e. changing the network topology).

5.3 DENSE ALGORITHM APPLICATION RESULTS

5.3.1 Case study network

New distribution network sectorization algorithm DeNSE has been tested on a large water distribution network. Case study network is well known from the literature and it is frequently used as a benchmark example for different modelling tasks. Network was originally presented as second case study network in the Battle of the Water Sensor Networks competition (BWSN2 – Ostfeld et al. (2008)). It is a real life WDN slightly modified to preserve its anonymity. This network has been used as a case study for number of other DMA design algorithms (Diao et al., 2013; Ferrari et al., 2014; Grayman et al., 2009; Hajebi et al., 2016; Zhang et al., 2017). Network consists of 12 523 nodes, 14 822 pipes, two reservoirs, two tanks, four pumps and five valves. Total demand in the network is Q_{tot} = 1243 L/s and total number of connections in the WDN is n_c = 77 916. Necessary input data for DeNSE algorithm, listed previously in Methodology section, are carefully set to allow proper comparison of the results with aforementioned researches in which the same network was used:

1) network’s EPANET input file is downloaded from Exeter Centre for Water System (http://emps.exeter.ac.uk/engineering/research/cws/downloads/benchmarks/);
2) minimum number of connections per DMA n_c^{min} = 500, maximum number of connections per DMA n_c^{max} = 5000

3) transmission main threshold is D_{main} = 350 mm
4) pipe closure threshold is D_{tr} = 300 mm
5) minimum and maximum operating network pressures are set to p_{min} = 20 m and p_{max} = 75 m, maximum allowable water age is WA_{max} = 48 h.

Based on total demand in the network (Q_{tot}), minimum (n_c^{min}) and maximum (n_c^{max}) number of connections in a DMA, and total number of connections in the network (n_c), minimum and maximum DMA size are calculated – d_{min} = 8 L/s and d_{max} = 80 L/s. The 24-h MDD simulation is used for hydraulic modeling, while for water quality modelling (*WA* calculation) extended period simulation of 192-h is used. Network topology with highlighted transmission main is shown in Figure 5.3, and distribution of pipe diameters in Figure 5.4.

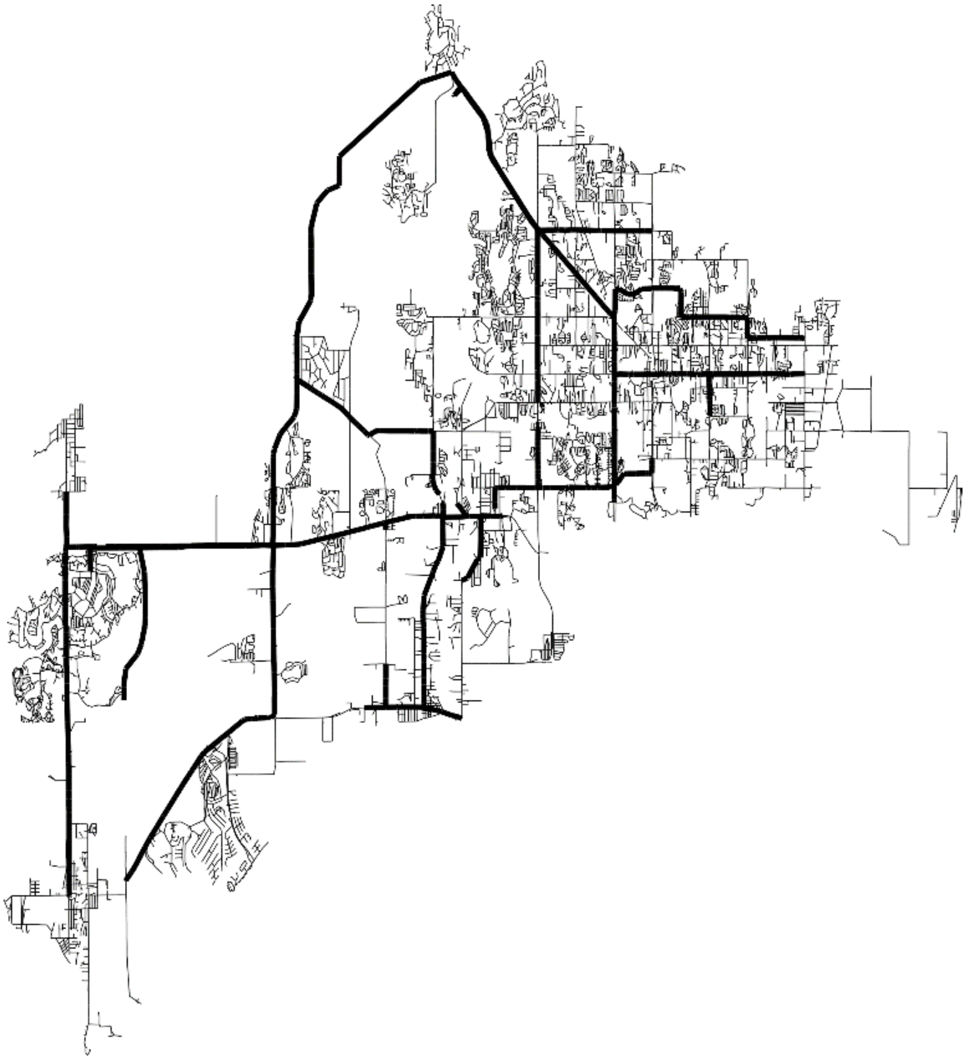


Figure 5.3 BWSN2 network with its transmission main

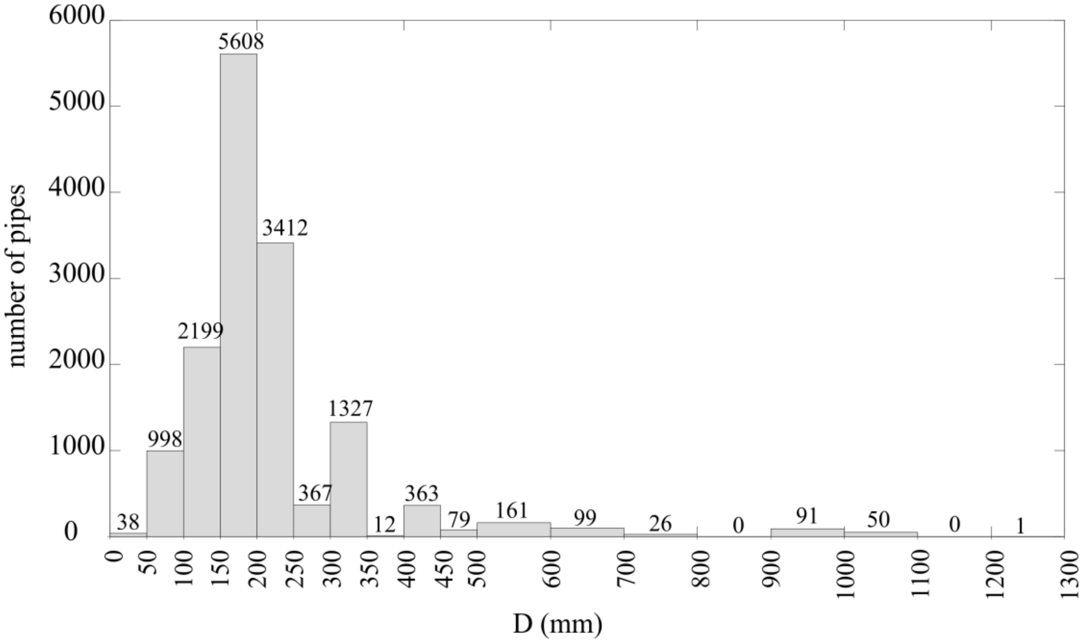


Figure 5.4 Distribution of pipes diameters in BWSN2 network

5.3.2 Results and discussion

5.3.2.1 Network clustering

Figure 5.5 shows evolution of uniformity index through network clustering process done in the 2nd Stage. Maximum uniformity value corresponds to the division in 43 clusters. Minimum number of clusters is 23 which is in accordance with research of Ferrari et al. (2014), in which the same transmission main diameter was used (350 mm) and 23 independent districts, connected to the main, were identified.

After 2nd Stage, 15 solutions are selected for further analysis having between 43 and 29 clusters. In the 3rd Stage flow meters and isolation valves are positioned to create DMAs and each solution is hydraulically analyzed. First solution (Sol-1), with 43 DMAs, does not satisfy pressure constraints and it is excluded as unfeasible. Performance indicators and other evaluation parameters for the remaining 14 solutions are shown in Table 5.8. As it can be seen from the Table 5.8, all solutions have relatively similar values of PIs (*WA* and *Res*). As the number of DMAs in the solution decreases, average number of connections per DMA increases, meaning that DMAs are larger in size. Consequently, for creation of smaller number of DMAs less flow meters and isolation valves is needed, which lowers the solution’s cost. Solution Sol-2 has one DMA which is smaller than minimum size d_{min}. In solutions Sol-3 to Sol-9 all DMAs are within specified d_{min}-d_{max} range, while in the solutions Sol-10 to Sol-15 there are one or two DMAs that are larger than d_{max}.

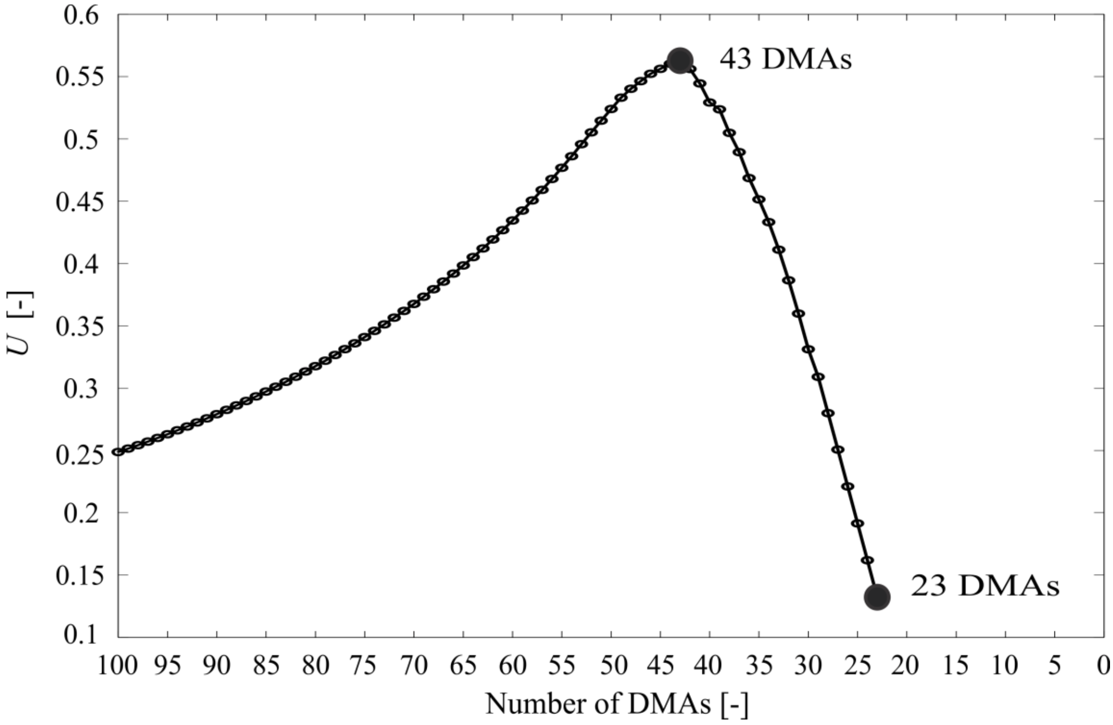


Figure 5.5: Evolution of Uniformity Index during clustering of BWSN2 network

Table 5.8 Evaluation parameters for 14 feasible solutions

Sol ID	N_{DMAs}	***NL***	***NS***	A_{conn}	***WA***	***Res***	***Cost***	N_M	N_V
[-]	[-]	[-]	[-]	[-]	[h]	[-]	[Eur]	[-]	[-]
Sol-2	42	0	1	1655	34.13	0.881	557405	81	178
Sol-3	41	0	0	1696	34.11	0.881	551215	80	177
Sol-4	40	0	0	1738	34.11	0.881	545870	79	177
Sol-5	39	0	0	1783	33.98	0.882	542210	79	176
Sol-6	38	0	0	1830	34.02	0.880	537920	77	176
Sol-7	37	0	0	1879	34.02	0.880	534500	76	175
Sol-8	36	0	0	1931	34.01	0.880	530995	76	169
Sol-9	35	0	0	1987	34.00	0.880	523685	75	166
Sol-10	34	1	0	2045	34.00	0.881	522565	75	164
Sol-11	33	1	0	2107	34.01	0.881	516375	74	163
Sol-12	32	2	0	2173	33.98	0.881	515815	74	162
Sol-13	31	2	0	2243	33.98	0.881	510470	73	162
Sol-14	30	2	0	2318	33.96	0.880	497205	71	153
Sol-15	29	2	0	2398	33.88	0.885	490470	71	138

5.3.2.2 Selection of preferable solution

Preferable solution is searched among solutions that fully satisfy DMA size constraints (solutions Sol-3 to Sol-9). To gain better insight in their advantages and drawbacks, solutions are plotted on two trade-off plots shown in Figure 5.6, with arrows on axis indicating directions of increasing preference.

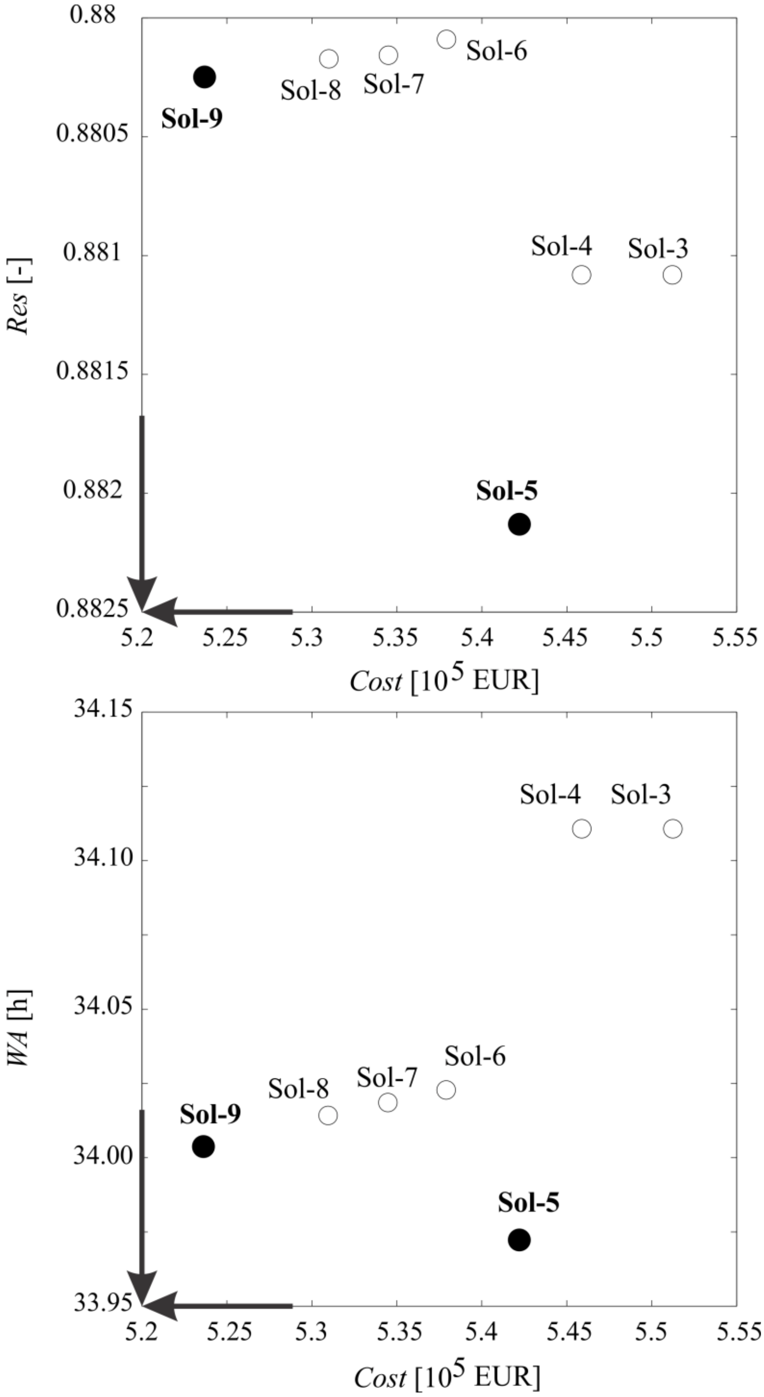


Figure 5.6: Trade-off plots of feasible solutions: *Cost* vs. *Res* and *Cost* vs. *WA*

First plot relates solution's cost to its resilience index, and the second cost to water age. In both cases, solutions Sol-5 and Sol-9 are best positioned on trade-off plots. As noted earlier, all feasible solutions have similar impact on network's resilience (*Res* = 0.880 - 0.885) and water age (*WA* = 33.88 – 34.13 h). Hence, between solutions Sol-9 and Sol-5, Sol-9 is preferred as it is less costly than Sol-5.

Figure 5.7 shows preferable solution Sol-9, which assumes sectorization of WDN into 35 DMAs.

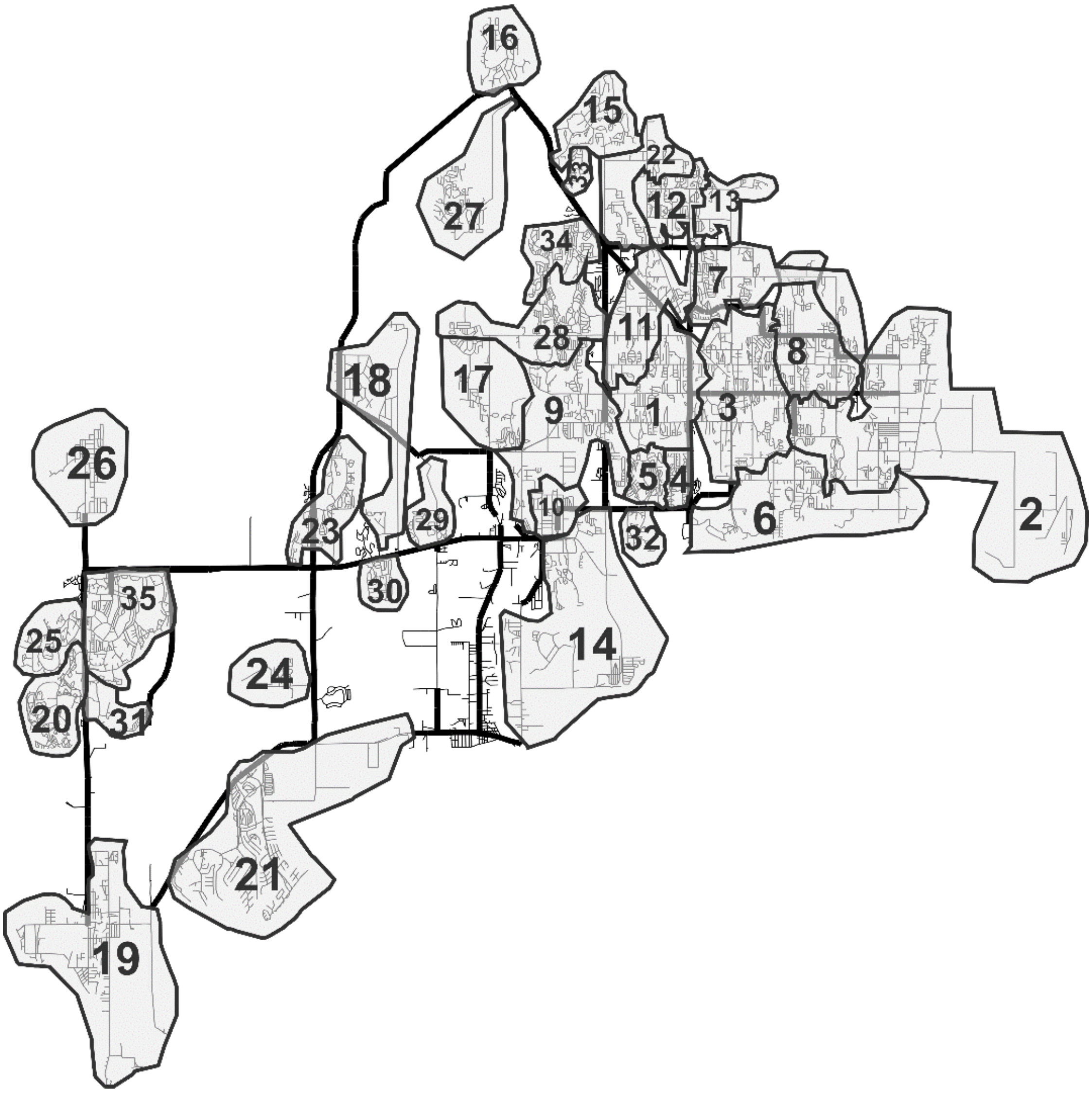


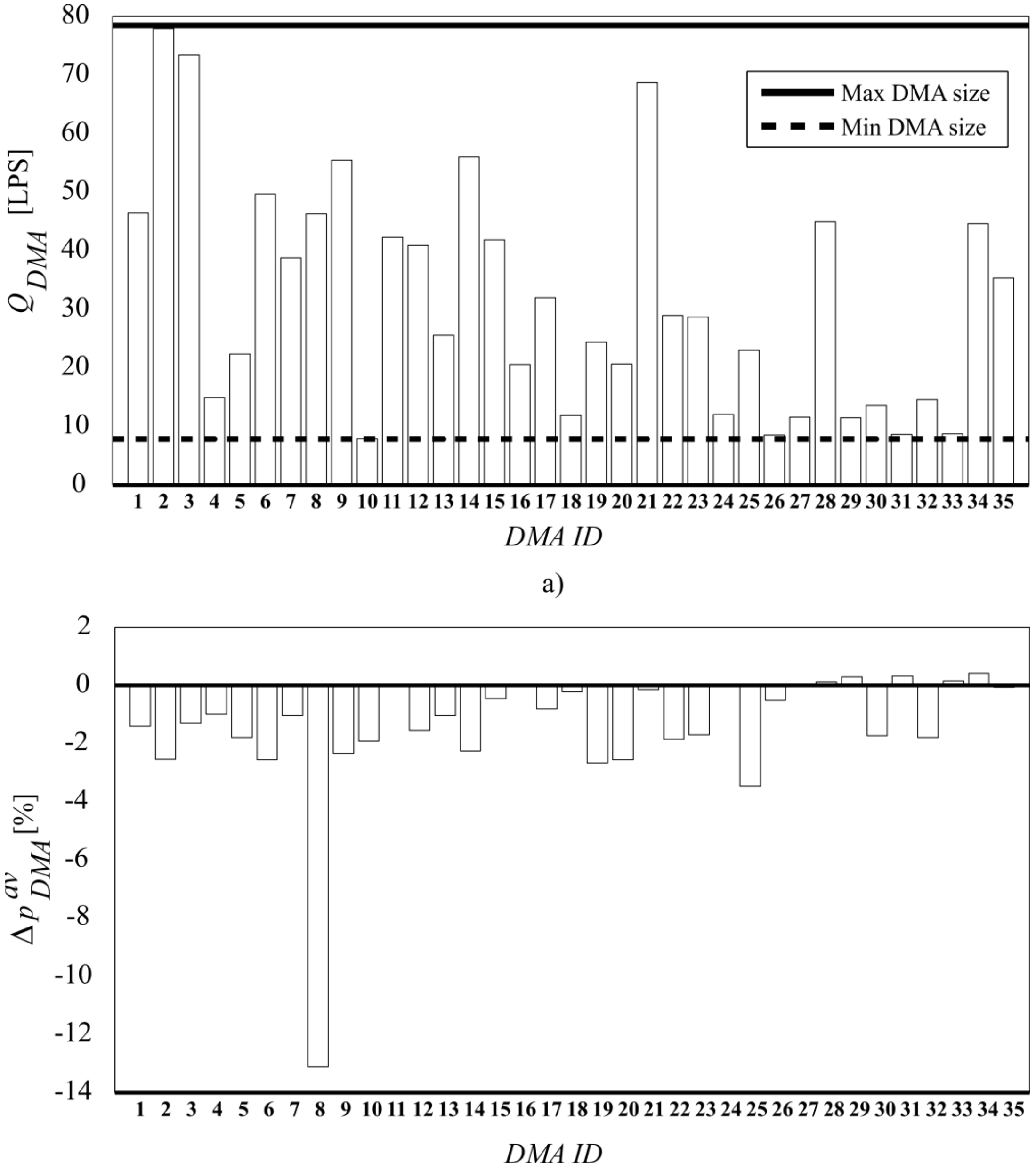
Figure 5.7: Preferable sectorization solution Sol-9 with 35 DMAs

To give further insight into the selected solution and the effects of network interventions required to create DMAs, Figure 5.8 and Figure 5.9 show results for each of 35 created

DMAs in selected solution Sol-9. Figure 5.8-a shows average consumption in DMAs, with highlighted minimum and maximum size constraints. As it can be seen from the graph, identified 35 DMAs vary in size considerably but always within the design limits imposed.

Figure 5.8-b shows relative changes in mean average pressure in DMAs, compared to the original, non-sectorized network. For most DMAs the mean average pressure has slightly decreased (up to 4%), whilst slight increase occurs in six DMAs (up to 1%). Therefore, network sectorization had very limited impact on re-distribution of pressure within the WDN. Significant decrease of pressure is observed in DMA #8 (by 13%), but all pressures are still whithin the required range of $p_{min} - p_{max}$. To support this observation, Figure 5.10 additionaly shows comparison of mean minimum and mean maximum pressures in each DMA before and after sectorization.

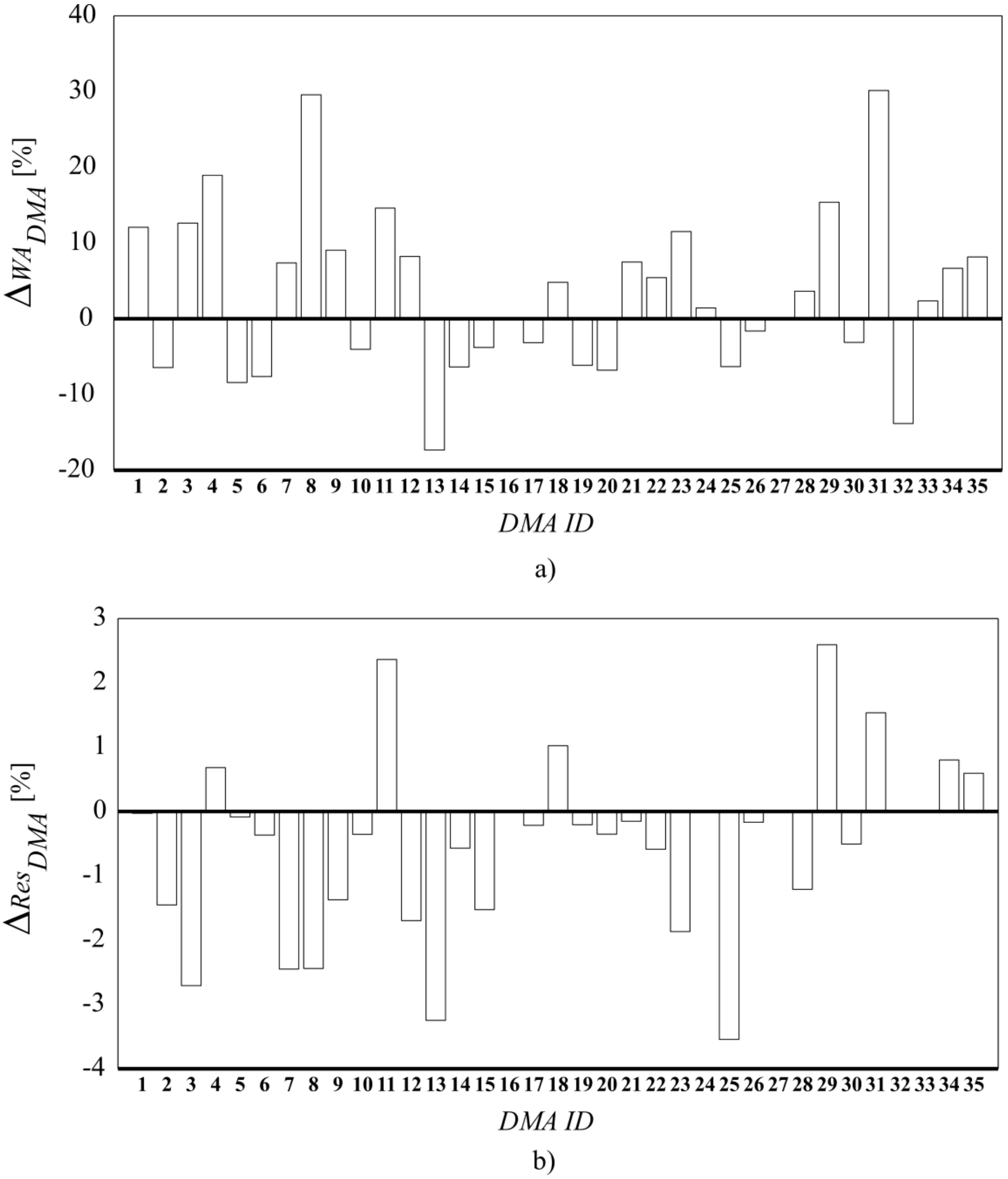
Figure 5.9-a illustrates relative changes in water age in the DMAs, again compared to the original network layout. Maximum decrease of *WA* is 20%, while increase is almost 30%. While decrease of *WA* is desirable, increase of 30% may seem a bit high at first. However, plotting absolute values of *WA* for DMAs in which increase is induced by network interventions (Figure 5.11) it is easy to conclude that *WA* is still well below set maximum WA_{max} of 48 h. Figure 5.9-b shows relative changes in DMAs resilience index. Changes in resilience index range from -3.5% to +2.2%, indicating very limited impact of sectorization on the resilience of the WDN.



a)

b)

Figure 5.8: Results for each DMA in selected preferable solution (Sol-9): a) average DMA consumption; b) relative change of mean average pressure



a)

b)

Figure 5.9 Results for each DMA in selected preferable solution (Sol-9): a) relative change of Water Age; b) relative change of Resilience Index

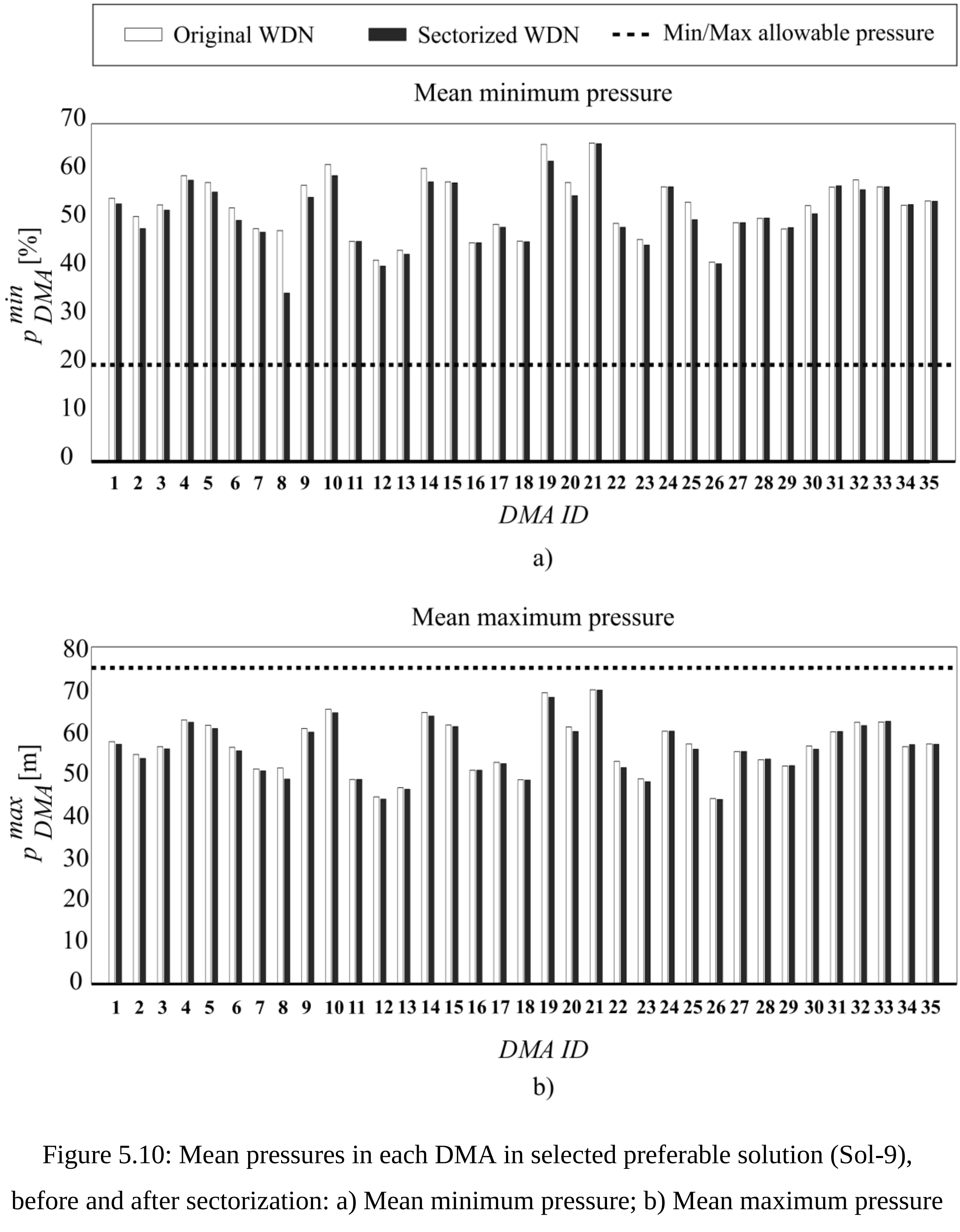


Figure 5.10: Mean pressures in each DMA in selected preferable solution (Sol-9), before and after sectorization: a) Mean minimum pressure; b) Mean maximum pressure

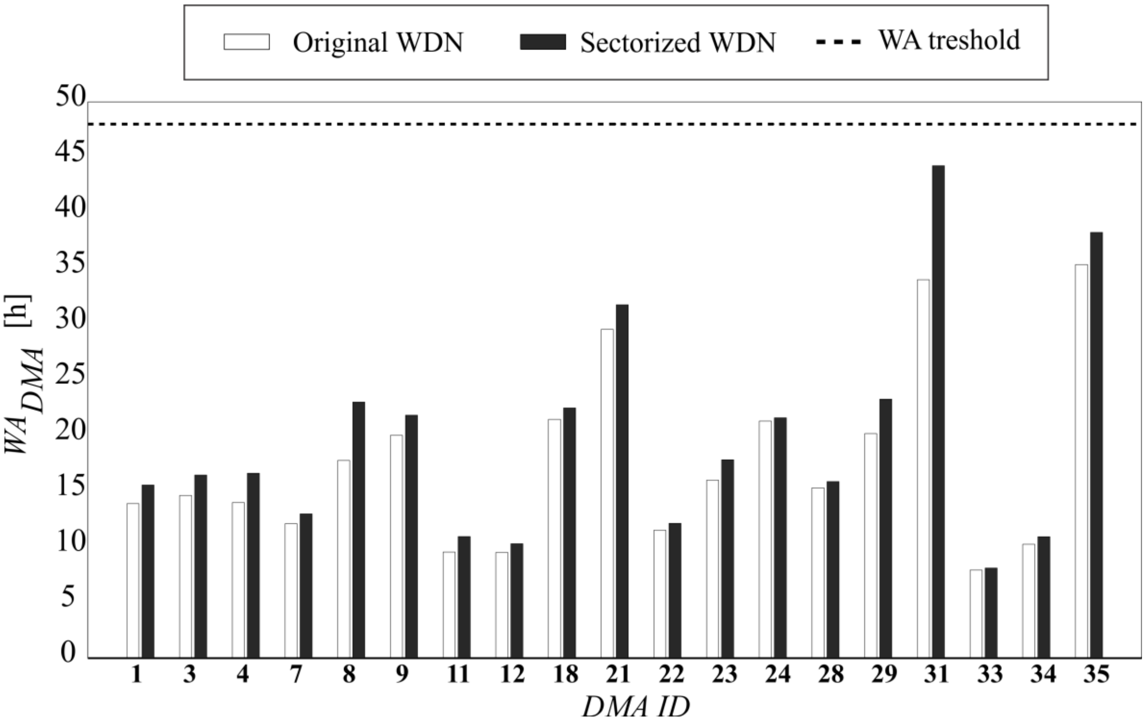


Figure 5.11: Values of water age, before and after sectorization, for DMAs with increased water age

From the results discussed it can be concluded that: 1) all DMAs are within required size limits in terms of consumption, 2) network's hydraulic performance is not endangered as changes in zone pressures are negligible, 3) water quality requirement, expressed through the *WA* parameter is satisfied, as for all DMAs *WA* is still below maximum allowed threshold of 48 h and 4) Network reliability is sustained as changes in resilience index are almost insignificant.

Enlarged DMA #23 is shown in Figure 5.12, to illustrate the network interventions required to create this DMA. Originally, cluster from which this DMA is created had 6 boundary pipes. Three of them are identified as links that always return water to the transmission main, and as such are marked for closure (v1, v2 and v3). Other three boundary pipes are "always-input to the zone" pipes, and using described methodology pipe v4 (*D* = 203.2 mm) is selected for closure, while other two pipes with larger diameters (*D* = 304.8 mm) are left opened and equipped with flow meters (fm1 and fm2).

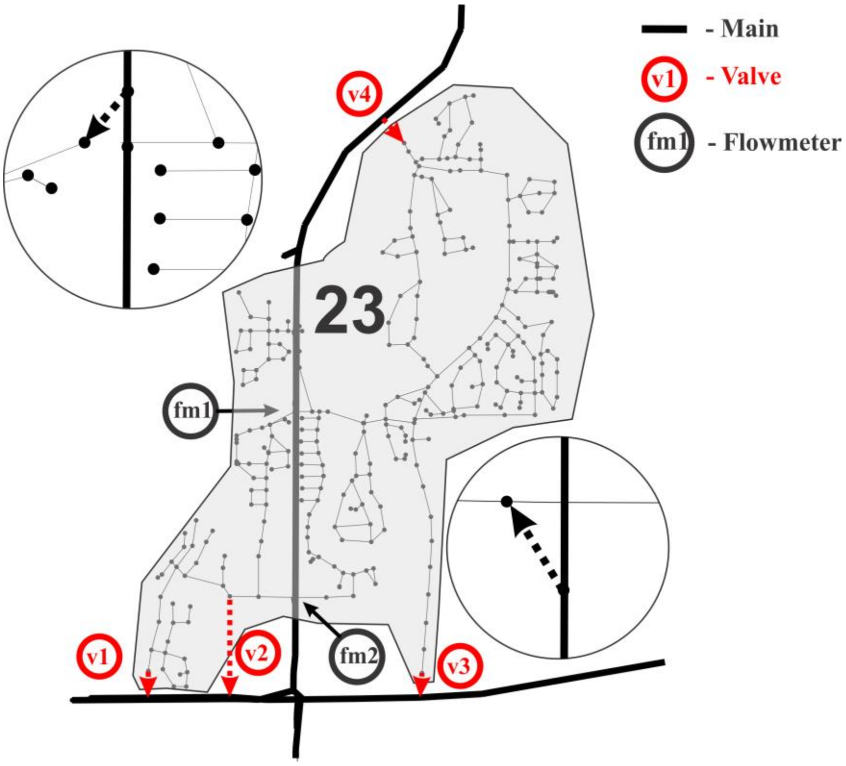


Figure 5.12: Boundary pipes of DMA #23

5.3.2.3 Comparison with other methods

Finally, a comparison of results obtained here is made to the corresponding results obtained using five previously published approaches that addressed the WDN sectorization problem, which also used the same case study (Table 5.9). Comparison is made in terms of number of DMAs (N_{DMAs}), DMAs that are larger (*NL*) and smaller (*NS*) than predefined size constraints, number of flow meters (N_M) and isolation valves (N_V), added pipes (P_{add}), average number of connections per DMA (A_{conn}) and computational time. Computational time is given only as a qualitative metric, to illustrate differences in magnitudes between different methods, and as such will be used in the discussion of the results. Table 5.9 gives an overview of sectorization methods used in each method for: a) partitioning the WDN and b) positioning of the flow meters and isolation valves.

As it can be seen from the Table 5.9, only methodology presented in Hajebi et al. (2016) and DeNSE algorithm, presented in this thesis, produce a set of feasible solutions. A total of 78 feasible solutions are identified in Hajebi et al. (2016) having anything between 28 and 48 DMAs. DeNSE algorithm identified 14 feasible solutions in which number of DMAs ranged between 29 and 42.

Regarding the DMA size constraints, solutions presented by Grayman et al. (2009) and Diao et al. (2013) have DMAs that are both larger and smaller, while in the solution presented by Ferrari et al. (2014) all DMAs fulfill size constraints. In Hajebi et al. (2016) all 78 feasible solutions meet size constraints, while in methodology presented here this is the case for 7 out of 14 feasible solutions.

Table 5.9 Comparison of results with other sectorization methods

Publsh. in	**Method for**		N_{DMAs}	***NL***	***NS***	N_M	N_V	P_{add}	A_{conn}	***Comp. Time***
	WDN partitioning	Device placement	[-]	[-]	[-]	[-]	[-]	[-]	[-]	[min/hrs]
Grayman et al. (2009)	Manual		43	1	3	53	163	11	1996	NA*
Diao et al. (2013)	Comm. detection	2 stage heuristic method	41	2	1	NA	NA	0	2044	20 min
Ferrari et al. (2014)	Graph based recursive bisection algorithm		36	0	0	181	152	0	2317	NA
Hajebi et al. (2016)	Heuristic graph partitioning	MO optimization	28-48	0	0	56-78	66-161	0	1415-2423	15 hrs
Zhang et al. (2017)	Comm. detection	MO optimization	43	NA	NA	103	33	0	NA	278 hrs
DeNSE Alg.	Uniformity based clustering	Engineering based heuristic	29-42	0-2	0-1	71-81	138-185	0	1656-2398	20 min

* NA – not available

Methodologies using MO optimization to position flow meters and isolation valves (Hajebi et al. (2016) and Zhang et al. (2017)) take significant amount of computational time (15 and 278 h respectively). Substantially lower computational time of Hajebi et al.'s method, compared to the method of Zhang et al., can be attributed to the use of shorter extended period simulation time (48 h compared to 192 h). To address this issue specifically, Diao et al. (2013) applied 2 stage heuristic procedure for device placement, resulting in acceptable running time of around 20 min. However only one solution with 41 DMAs, three of them falling out of the required size limits, is reported. Engineering based heuristic procedure used in methodology presented here takes similar amount of time (about 20 min), but produces a set of feasible solutions.

Methodologies of Ferrari et al. (2014) and Hajebi et al. (2016) ensure connectedness of each DMA to the transmission main (direct access to water source) and their isolation from other DMAs (i-DMAs). While methodology presented here does not create i-DMAs, preferable solution presented earlier (Sol-9) fulfills condition of direct access to water source. All 35 DMAs are directly connected to the transmission main: 20 DMAs with 1 pipe, 4 with 2, 6 with 3, 4 with 4 and 1 with 6 pipes.

Table 5.10 gives comparison of main PIs values obtained with different methods – water age (*WA*) and resilience index (*Res*). Presented results show that DeNSE algorithm achieves slightly better value of resilience index and slightly worse value of water age. Reported results are only indicative as different input parameters, affecting the values of compared PIs, are used. For *WA* calculation Grayman et al. (2009), Diao et al. (2013) and methodology presented here use 192-h extended period simulation, while Hajebi et al. (2016) uses 48-h simulation. Furthermore, the *WA* value is highly dependent on the adopted time step for water quality simulation and those papers do not supply this information. Grayman et al. (2009) reported increase of 2.61% in *WA* for the DMA system, when compared to the original network (from 30.71 h to 31.51 h). In the case of DeNSE algorithm, *WA* is increased by 3.31 % for the DMA system (from 32.91 h to 34 h) which is regarded as insignificant increase and same order of magnitude as achieved in Grayman et al. (2009).

Reported *Res* indices are influenced by the adopted minimum allowable pressure in the network and time period over which they are averaged. Grayman et al. (2009) adopted minimum pressure of 30 psi (20 m) and 51-h time period. Hajebi et al. (2016) used 28 m minimum pressure and 48-h time period, while Diao et al. (2013) did not report values of *Res* PI. Grayman et al. (2009) report decrease of *Res* of 4.07 % for the DMA system, when compared to the original network (from 0.836 to 0.802), while the DeNSE algorithm achieves lower decrease of 2.55 % (from 0.903 for the original network to the 0.88 for the DMA system). As noted above, due to the different input parameters, values presented in Table 5.10 are not directly comparable, but illustrative and show that in terms of water age and resilience all methods perform similarly.

Table 5.10: Comparison of main PIs for best solutions

Published in	***WA*** **[h]**	***Res*** **[/]**
Grayman et al. (2009)	31.51	0.802
Diao et al. (2013)	32.01	Not av.
Hajebi et al. (2016)	31.01	0.83
DeNSE Algorithm	34.00	0.88

5.3.3 Summary

DeNSE sectorization algorithm has been tested on a large real sized water distribution network BWSN2, and its results are compared to other available algorithms that dealt with the same case study network. Based on the results presented above, following summary conclusions are drawn:

1. Interventions in the network designed by the algorithm, necessary for creation of DMAs, do not worsen operational status of WDN compared to its baseline condition. Satisfaction of hydraulic constraints (e.g. min and max pressures) is provided with minimum changes in pressures compared to the original WDN layout. Water quality is not endangered, as water retention time in WDN is below maximum threshold value.
2. Cost of a specific sectorization is calculated explicitly, based on unit cost functions for valves and flow meters, opposed to other algorithms where cost is mainly accounted indirectly through number of installed devices or summarized diameters. Having in mind importance of economical aspect for WDN management and the fact that different WDNs have varying topology and distribution of diameters, DeNSE algorithm can provide better assessment of sectorization cost than its alternatives.
3. Algorithm is able to identify a set of good feasible sectorization solutions, even for large networks such as the case study used here. Reported computational efficiency of the algorithm is one of its strong points, as it allows generation of feasible solutions in reasonable time (from user point of view). Consequently, this enables investigation of different sectorization strategies (by changing input parameters) more efficiently. High computational efficiency comes mainly from

the newly presented methodology for positioning the flow meters and isolation valves, based on common sense engineering heuristics. Advantage of this approach is noticeable especially when DeNSE algorithm is compared with algorithms using MO optimization for defining DMAs boundaries.

Given the presented results and above drawn conclusions, DeNSE sectorization algorithm is particularly convenient for application in initial stages of DMAs design process. This relates specifically to WDNs in which prior definition of DMAs has not been implemented. For example, sectorization solutions of different resolutions (e.g. more small size DMAs or less large size DMAs) can be analyzed. Preservation of network hydraulic performance and reliability, provided by DeNSE algorithm, additionally contribute to its potential application for low efficient WDNs (i.e. networks with high water losses). In such networks, initial aim of DMAs design is to track network water balance, and not to control pressures in the network. Hence, main design criterion is to minimize the sectorization cost, while preserving operational performance of WDN.

CHAPTER 6:

CONCLUSIONS AND FUTURE WORK

6 CONCLUSIONS AND FUTURE WORK

Table of contents:

6.1 SUMMARY OF CONCLUSIONS

The aim of the thesis was to develop a decision support methodology for sectorization of WDN, usable for water companies and practicing engineers, especially ones dealing with initial implementation of sectorization strategy. Water utilities managing WDNs in developing countries are usually inefficiently operated and consequently they suffer from significant amount of water (e.g. leakage) and revenue loss. Poor infrastructure and maintenance, shortage of systematical data monitoring and lack of financial means are just some of the factors contributing to the overall bad condition of WDN. Designing an adequate sectorization solution for such WDNs, coping with aforementioned difficulties, pose a challenging task addressed with methodology presented in this thesis. Key contributions made in this research are new method for hydraulic simulation and new WDN sectorization algorithm, developed as a part of proposed decision support methodology.

Low computational efficiency is recognized as one of the downsides of available sectorization methods, limiting their application to large real-sized WDNs. This comes from excessive utilization of optimization methods with many objective functions, often not significant for proper functioning of WDN. New method for hydraulic simulation (TRIBAL-ΔQ) is developed and presented in Chapter 3, specifically to address this issue. TRIBAL-ΔQ is a loop-flow based method for hydraulic simulation which can be beneficial for improving the overall efficiency of sectorization methods, if used inside optimization algorithm to perform multiple hydraulic calculations. New method combines novel loop identification algorithm (TRIBAL) and efficient implementation of loop-flow solver (ΔQ).

Main sectorization objective adopted in methodology presented here is to design layout of DMAs that will allow efficient tracking of water balance in the network. Least investment for field implementation and maintaining the same level of WDN’s operational efficiency are adopted as main design criteria. For the purpose of WDN sectorization new algorithm (DeNSE), utilizing above-named objective and design criteria, is developed and presented in Chapter 4. It utilizes newly developed uniformity index (U) which drives the sectorization process into DMAs that are uniform in size and within predefined limits. New engineering heuristic is developed and used, instead of optimization method, to position the flow-meters and isolation valves on DMAs boundaries. This contributes to better computational efficiency of DeNSE algorithm when compared to other methods using optimization, as discussed in the paragraph above. WDN’s post-sectorization operational efficiency is evaluated using the two adopted PIs – resilience and water age. PIs adopted in this research are not limiting to the presented methodology, as others can be adopted as well.

Benchmarking results provided in Chapter 5 are indeed encouraging. TRIBAL-ΔQ method based hydraulic solver is benchmarked against node based GGA solver, considered as an etalon in scientific community, on four large networks of varying topology and different complexity. Comparison criteria included computational efficiency, convergence and prediction accuracy. TRIBAL-ΔQ solver proved to be significantly computationally faster than the GGA solver. It showed stable numerical performance by converging successfully when performing hydraulic analysis in all test cases, compared to the GGA which failed in one. Prediction accuracy wise, both solvers demonstrated similar performance.

DeNSE sectorization algorithm has been tested on a large real sized water distribution network, and its results are compared to other available algorithms that dealt with the same case study. Obtained results prove that DeNSE algorithm is able to identify a good set of feasible sectorization solutions. Additionally, required interventions in the WDN are minimal and they do not worsen its operational performance, compared to the original network layout. Providing the set of solutions, instead of just one, enables decision makers to select the solution best suitable to their preferences. Reported computational efficiency, better than other comparable algorithms, allows generation of different sets of feasible solutions in reasonable time (i.e. minutes) even for large networks. Advantage is clearly noticeable when DeNSE algorithm is compared to algorithms using MO optimization. Consequently, investigation of different sectorization strategies can be achieved more efficiently. Minimal sectorization cost and preservation of network hydraulic performance and reliability, provided by DeNSE algorithm, make it particularly convenient for initial stages of DMAs design process.

6.2 FUTURE WORK

Some possible extensions of DeNSE sectorization algorithm are presented in section 4.3, but the results of their application are not presented in this thesis, and remain to be confirmed in the future.

Hierarchical sectorization of WDN, i.e. creation of DMAs in phases following increased knowledge of the WDN’s operation, was set as one of the research questions at the end of the Chapter 2. Although practical implementation of this approach on a case study is not presented here, section 4.3.1 reflected on some benefits and explained that hierarchical ordering of DMAs is characteristic already implemented in DeNSE algorithm.

Presented results already proved the dominance of DeNSE algorithm in computational efficiency, especially over the algorithms using optimization methods to position the flow meters and valves on DMAs boundaries. Partially, that dominance is achieved through the use of developed engineering heuristic. Although good feasible solutions are identified, downside of this approach adopted in DeNSE is limited solution search space. Section 4.3.2 proposed coupling of DeNSE with an optimization method in order to search broader specter of sectorization solutions. Given the proven efficiency of the new TRIBAL-ΔQ method for hydraulic simulation, it is hypothesized that with proposed coupling, search of the solution space would be achieved more efficiently than with other available methods employing GGA solver to solve hydraulics. In this manner, achieved computational efficiency of DeNSE algorithm, presented here, would be retained.

Main sectorization criteria used in this research are minimum investment in implementation of the designed solution and preservation of system's operational efficiency. This set of criteria is suitable for initial sectorization of WDNs in which no sectorization measures were implemented in the past and lacking measurement data. For well managed WDNs, main sectorization objective is to improve system monitoring by partitioning already established DMAs into finer resolution. This requires delicate network interventions that could possibly significantly affect water supply, water quality and overall system reliability. DeNSE algorithm can be successfully applied to these networks as well, but should probably include additional sectorization criteria such as design for fire flows, specific water quality parameters (e.g. Chlorine), design for security, etc.

Presented implementation of TRIBAL-ΔQ method also suffers from limitations imposed by adopted key assumptions. Currently, only demand driven analysis is available and network topology during hydraulic simulation has to remain unchanged (e.g. closure of the valves or simulation of valves that can change status during simulation are not available). These assumptions are not limiting for application with the decision support methodology proposed in this thesis, since the pressure control in the network is not set as main sectorization objective. However, if TRIBAL-ΔQ method is to be used for other purposes, contrary to the current assumptions, some modifications would have to be made.

Successful pressure driven hydraulic analysis with loop-flow based method is still not reported in the literature. Preliminary testing done during this research revealed problems with the convergence of the ΔQ solver. Changing the topology of the network will clearly influence the structure of the loops identified with the TRIBAL algorithm, and hence require additional computational time to deal with the factorization of the system matrix (because of its changed sparsity). Introduction of control devices would not require completely new identification of loops each time that device changes its status. These devices would only influence some of the loops and not all of them, depending on the number and location of the introduced control devices. Both pressure driven analysis and network's changing topology remain to be addressed in the future research.

7 REFERENCES

Adams, R. W. (1961). Distribution Analysis by Electronic Computer. *Journal of the Institute of Water Engineers, 15,* 415–428.

Alegre, H., Baptista, J. M., Cabrera, E. J., Cubillo, F., Duarte, P., Hirner, W., … Parena, R. (2006). *Performance Indicators for Water Supply Services* (2nd ed.). London: IWA Publishing.

Alvarruiz, F., & Vidal, A. M. (2015). Improving the Efficiency of the Loop Method for the Simulation of Water Distribution Systems, *141*(10), 1–10. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000539.

Alvisi, S. (2015). A New Procedure for Optimal Design of District Metered Areas Based on the Multilevel Balancing and Refinement Algorithm. *Water Resources Management, 29*(12), 4397–4409. https://doi.org/10.1007/s11269-015-1066-z

Alvisi, S., & Franchini, M. (2014). A heuristic procedure for the automatic creation of district metered areas in water distribution systems. *Urban Water Journal, 11*(2), 137–159. https://doi.org/10.1016/j.proeng.2014.02.006

Arsene, C., Bargiela, A., & Al-Dabass, D. (2004). Modelling and Simulation of Water Systems Based on Loop Equations. *International Journal of Simulation Systems, Science & Technology: Special Issue on Modelling & Simulation of Complex Systems, 5*(1), 61–72.

Babić, B., Đukić, A., & Stanić, M. (2014). Managing water pressure for water savings in developing countries. *Water SA*, *40*(2), 221. https://doi.org/10.4314/wsa.v40i2.4

Baker. (2007). *Municipal Water Distribution System Secutrity Study: Recomendations for Science and Technology Investments.*

Barnat, J., Brim, L., & Cerna, I. (2002). Property Driven Distribution of Nested DFS. In *Proceedings of the 3rd International workshop on verification and computation logic (VCL'2002)*. University of Southampton, UK.

Burrows, R., Crowder, G., & Zhang, J. (2000). Utilisation of network modelling in the operational management of water distribution systems. *Urban Water*, *2*(2), 83–95. https://doi.org/10.1016/S1462-0758(00)00046-7

Butler, D. (2000). *Leakage detection and management*. Cwambran, UK: Palmer Environmental Ld.

Cao, C. (1963). Sulla convergenza del Metodo di Cross. In *VIII Convegno di Idraulica e Construzioni Idrauliche, Piza*.

Cavazzuti, M. (2013). *Optimization methods: from theory to design - Scientific and technological aspects in mechanics*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-31187-1

Cerna, I., & Pelanek, R. (2003). Distributed Explicit Fair Cycle Detection (Set Based Approach).pdf. In *Model Checking Software" from the 10th International SPIN Workshop* (pp. 49–73). Springer Berlin Heidelberg.

Cheng, S. W., Dey, T. K., & Shewchuk, J. R. (2013). *Delaunay mesh generation*. Boca Raton, Florida: CRC Press.

Cherry, E. C. (1951). Some General Theorems for Non-Linear Systems Possessing Reactance. *Philosophical Magazine (Series 7)*, *42*, 1161–1177.

Ciaponi, C., Murari, E., & Todeschini, S. (2016). Modularity-Based Procedure for Partitioning Water Distribution Systems into Independent Districts. *Water Resources Management*, *30*(6), 2021–2036. https://doi.org/10.1007/s11269-016-1266-1

Clauset, A., Newman, M., & Moore, C. (2004). Finding community structure in very large networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, *70*(6 2), 1–6. https://doi.org/10.1103/PhysRevE.70.066111

Collins, M., Cooper, L., Helgason, R., Kennington, J., & LeBlanc, L. (1978). Solving the Pipe Network Analysis Problem Using Optimization Techniques. *Management Science*, *24*(7), 747–760.

Creaco, E., & Franchini, M. (2014). Comparison of Newton-Raphson Global and Loop Algorithms for Water Distribution Network Resolution. *Journal of Hydraulic Engineering*, *140*(3), 313–321. https://doi.org/10.1061/(ASCE)HY.1943-7900.0000825

Creaco, E., & Franchini, M. (2015). The identification of loops in water distribution networks. *Procedia Engineering*, *119*(1), 506–515. https://doi.org/10.1016/j.proeng.2015.08.878

Cross, H. (1936). Analysis of flow in networks of conduits or conductors. *University of Illinois Bulletin: Engineering Experiment Station*, (286).

De Paola, F., Fontana, N., Galdiero, E., Giugni, M., Savic, D., & Sorgenti Degli Uberti, G. (2014). Automatic multi-objective sectorization of a water distribution network. *Procedia Engineering*, *89*, 1200–1207. https://doi.org/10.1016/j.proeng.2014.11.250

De Pina, J. (1995). *Applications of shorthest path methods*. Univeristy of Amsterdam.

Deuerlein, J., Elhay, S., & Simpson, A. (2016). Fast Graph Matrix Partitioning Algorithm for Solving the Water Distribution System Equations. *Journal of Water Resources Planning and Management*, *142*(1). https://doi.org/http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000561

Deuerlein, J. W. (2008). Decomposition Model of a General Water Supply Network Graph. *Journal of Hydraulic Engineering*, *134*(6), 822–832. https://doi.org/10.1061/(ASCE)0733-9429(2008)134:6(822)

Di Nardo, A., & Di Natale, M. (2011). A heuristic design support methodology based on graph theory for district metering of water supply networks. *Engineering Optimization*, *43*(2), 193–211. https://doi.org/10.1080/03052151003789858

Di Nardo, A., Di Natale, M., Guida, M., & Musmarra, D. (2013a). Water Network Protection from Intentional Contamination by Sectorization. *Water Resources Management*, *27*(6), 1837–1850. https://doi.org/10.1007/s11269-012-0133-y

Di Nardo, A., Di Natale, M., Santonastaso, G. F., & Venticinque, S. (2011). Graph partitioning for automatic sectorization of a water distribution system. In D. Butler, D. Savic, & Z. Kapelan (Eds.), *Proceedings of computer and control in water industry CCWI 2011* (pp. 841–846). Exeter: Centre for Water Systems, 2011.

Di Nardo, A., Di Natale, M., Santonastaso, G. F., & Venticinque, S. (2013b). An Automated Tool for Smart Water Network Partitioning. *Water Resources Management*, *27*(13), 4493–4508. https://doi.org/10.1007/s11269-013-0421-1

Di Nardo, A., Di Natale, M., Santonastaso, G., Tzatchkov, V., & Alcocer-Yamanaka, V. (2014). Water network sectorization based on graph theory and energy performance indices. *Journal of Water Resources Planning and Management*, *140*(5), 620–629. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000364

Diao, K., Zhou, Y., & Rauch, W. (2013). Automated Creation of District Metered Area Boundaries in Water Distribution Systems. *Journal of Water Resources Planning and Management*, *139*(2), 184–190. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000247

Dijkstra, E. W. (1959). A note on two problem in connection with graphs. *Numerische Mathematik*, (1), 269–271.

Dillingham, J. H. (1967). Computer Analysis of Water Distribution Systems, Part II. *Water and Sewage Works*, 43–45.

Duan, Q., & Gupta, V. (1992). SAC-SMA, *28*(4), 1015–1031.

Elhay, S., Simpson, A. R., Deuerlein, J., Alexander, B., & Schilders, W. (2014). A Reformulated Co-tree Flows Method competitive with the global Gradient Algorithm for solving the water distribution system equations. *Journal of Water Resources Planning and Management*, *140*(1970), 1–10. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000431.

Epp, R., & Fowler, A. G. (1970). Efficient Code for Steady-State Flows in Networks. *Journal of Hydraulics Division*, *96*(HY1), 43–56.

Farley, M. (2001). Leakage Management and Control : A Best Practice Training Manual, 1–169. Retrieved from http://whqlibdoc.who.int/hq/2001/WHO_SDE_WSH_01.1_pp1-98.pdf

Ferrari, G., & Savic, D. (2015). Economic performance of DMAs in water distribution systems. *Procedia Engineering*, *119*(1), 189–195. https://doi.org/10.1016/j.proeng.2015.08.874

Ferrari, G., Savic, D., & Becciu, G. (2014). A Graph Theoretic Approach and Sound Engineering Principles for Design of District Metered Areas. *Journal of Water Resources Planning and Management*, *140*(12), 1–13. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000424

Gabow, H. N. (2000). Path-Based Depth- rst Search for Strong and Biconnected Components 1 Introduction 2 Strong Components. *Information Processing Letters*, *74*, 107–114.

George, A., & Liu, J. W.-H. (1981). *Computer solution of large sparse positive definite systems*. Englewood Cliffs,NJ: Prentice-Hall.

Giustolisi, O., & Ridolfi, L. (2014a). A novel infrastructuremodularity index for the segmentation of water distribution networks. *Water Resources Research*, *50*, 7648–7661. https://doi.org/10.1002/2014WR016067

Giustolisi, O., & Ridolfi, L. (2014b). New Modularity-Based Approach to Segmentation of Water Distribution Networks. *Journal of Hydraulic Engineering*, *140*(10), 04014049. https://doi.org/10.1061/(ASCE)HY.1943-7900.0000916

Graves, Q. B., & Branscome, D. (1958). Digital Computers for Pipeline Network Analysis. *Journal of the Sanitary Engineering Division*, *84*(SA2), 1608-1-1608–1618.

Grayman, W., Murray, R., & Savic, D. (2009). Effects of Redesign of Water Systems for Security and Water Quality Factors. In *World Environmental and Water Resources Congress 2009* (pp. 504–514). https://doi.org/10.1061/41036(342)49

Hadka, D., & Reed, P. (2013). Borg: An auto-adaptive many-objective evolutionary computing framework. *Evolutionary Computation*, *21*(2), 231–259. https://doi.org/10.1162/EVCO_a_00075

Hajebi, S., Roshani, E., Cardozo, N., Barrett, S., Clarke, A., & Clarke, S. (2016). Water distribution network sectorisation using graph theory and many-objective optimisation. *Journal of Hydroinformatics*, *18*(1), 77–95. https://doi.org/10.2166/hydro.2015.144

Hamam, Y. M., & Brameller, A. (1971). Hybrid method for the solution of piping networks. *Electrical Engineers, Proceedings of the Institution Of*, *118*(11), 1607–1612. https://doi.org/10.1049/piee.1971.0292

Hoag, L. N., & Weinberg, G. (1957). Pipeline Network Analysis by Electronic Digital Computer. *Journal of American Water Works Association*, *49*(5), 517–524.

Hoffman, J. D. (2001). *Numerical methods for engineers and scientists* (Second). New York: Marcel Dekker, Inc.

Horton, J. D. (1987). A polynomial-time algorithm to find a shorthest cycle basis of a graph. *SIAM Journal of Computing*, (16), 359–366.

Isaacs, L. T., & Mills, K. G. (1980). Linear Theory for Pipe Network Analysis. *Journal of Hydraulics Division*, *106*, 1191–1201.

Ivetić, D., Vasilić, Ž., Stanić, M., & Prodanović, D. (2016). Speeding up the water distribution network design optimization using the DQ method. *Journal of Hydroinformatics*, *18*(1), 33–48. https://doi.org/10.2166/hydro.2015.118

Jha, K. (2007). Automatic minimal loop extraction and initialisation for water pipe network analysis. *International Journal of Simulation: Systems, Science and Technology*, *8*(2), 8–19.

Jungnickel, D. (2005). *Graphs, Networks and Algorithms*. (M. Bronstein, A. Cohen, H. Cohen, D. Eisenbud, & B. Sturmfels, Eds.) (Second). Berlin: Springer.

Kanakoudis, V., Tsitsifli, S., Samaras, P., Zouboulis, a, & Demetriou, G. (2011). Developing appropriate performance indicators for urban distribution systems evaluation at Mediterranean countries. *Water Utility Journal*, 31–40.

Karypis, G., & Kumar, V. (1998). METIS* A Software Package for Partitioning

Unstructured Graphs , Partitioning Meshes , and Computing Fill-Reducing Orderings of Sparse Matrices. *Manual*, 1–44.

Kavitha, T., & Mehlhorn, K. (2005). A polynomial time algorithm for minimum cycle basis in directed graphs. In *STACS 2005. 22nd annual symposium on theoretical aspects of computer science, Stuttgart, Germany, February 24--26, 2005. Proceedings.* (Vol. 14186, pp. 654–665). https://doi.org/10.1007/b106485

Kavitha, T., Mehlhorn, K., Michail, D., & Paluch, K. (2004). A faster algorithm for minimum cycle bases of graphs. *Proceedings of ICALP*, *3124*, 846–857. https://doi.org/10.1007/b99859

Kesavan, H. K., & Chandrashekar, M. (1972). Graph-Theoretic Models for Pipe Network Analysis. *Journal of Hydraulics Division*, *98*(HY2), 345–364.

Kingdom, B., Liemberger, R., & Marin, P. (2006). WATER SUPPLY AND SANITATION SECTOR BOARD DISCUSSION PAPER SERIES The Challenge of Reducing Non-Revenue Water (NRW) in Developing Countries How the Private Sector Can Help : A Look at Performance-Based Service Contr, (8).

Kruskal, J. B. (1956). On the Shorthest Spanning Subtree of a Graph and the Traveling Salesman Problem. In *Proceedings of the American Mathematical Society, Vol. 7, No.1* (pp. 48–50).

Laucelli, D. B., Simone, A., Berardi, L., & Giustolisi, O. (2016). Optimal Design of District Metering Areas. *Procedia Engineering*, *162*(2014), 403–410. https://doi.org/10.1016/j.proeng.2016.11.081

Lemieux, P. F. (1972). Efficient Algorithm for Distribution Networks. *Journal of Hydraulics Division*, *98*(HY11), 1911–1920.

Liu, K. T. H. (1969). The Numerical Analysis of Water Supply Networks. In *XIII Congress of the International Association for Hydraulic Research* (pp. 35–42).

Martin, D. W., & Peters, G. (1963). The Application of Newton’s Method to Network Analysis by Digital Computer. *Journal of the Institute of Water Engineers*, *17*, 115–129.

Mathworks. (2010). User’s guide (2010b).

Millar, W. (1951). Some general theorems for non-linear systems possessing resistance. *Philosophical Magazine (Series 7), 42,* 1150–1160. https://doi.org/10.1080/14786445108561361

Morrison, J. (2004). Managing leakage by District Metered Areas: a practical approach. *Water*, *21*(2), 44–46.

Morrison, J., Tooms, S., & Rogers, D. (2007). *DMA management guidance notes*. London: IWA.

Newman, M. E. J. (2004). Analysis of weighted networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, *70*(5 2), 1–9. https://doi.org/10.1103/PhysRevE.70.056131

Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, *69*(2 2), 1–15. https://doi.org/10.1103/PhysRevE.69.026113

Ormsbee, L. E. (2008). The History of Water Distribution Network Analysis: The Computer Age. In *Water Distribution Systems Analysis Symposium 2006* (pp. 1–6). https://doi.org/10.1061/40941(247)3

Osiadacz, a. J. (1988). Comparison of numerical methods for steady-state simulation of gas networks. *Civil Engineering Systems*, *5*(1), 25–30. https://doi.org/10.1080/02630258808970499

Ostfeld, A., Uber, J. G., Salomons, E., Berry, J. W., Hart, W. E., Phillips, C. a, … Walski, T. (2008). The Battle of the Water Sensor Networks (BWSN): A Design Challenge for Engineers and Algorithms. *Journal of Water Resources Planning and Management*, *134*(6), 556–568. https://doi.org/10.1061/(ASCE)0733-9496(2008)134:6(556)

Perelman, L., & Ostfeld, A. (2012). *Water-Distribution Systems Simplifications through*

Clustering. *Journal of Water Resources Planning and Management* (Vol. 138). https://doi.org/10.1061/(ASCE)WR.1943-5452.0000173

Piller, O. (1995). *Modelling the behaviour of a network: Hydraulic analysis and sampling procedures for parameter estimation*. University of Bordeaux, Bordeaux, France.

Pohl, I. S. (1969). *Bi-directional and heuristic search in path problems*. Stanford University, California, Stanford.

Rahal, H. (1995). A co-tree flows formulation for steady state in water distribution networks. *Advances in Engineering Software*, *22*(3), 169–178. https://doi.org/10.1016/0965-9978(95)00020-W

Rossman, L. a. (2000). EPANET 2: users manual. *Cincinnati US Environmental Protection Agency National Risk Management Research Laboratory*, *38*(September), 200. https://doi.org/10.1177/0306312708089715

Salomons, E., Skulovich, O., & Ostfeld, A. (2017). The Battle of Water Networks DMAs – A Multistage Design Approach. *Journal of Water Resourcers Planning and Management*, *143*(10), 1–8. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000830.

Savić, D., & Ferrari, G. (2014). Design and performance of district metering areas in water distribution systems. *Procedia Engineering*, *89*, 1136–1143. https://doi.org/10.1016/j.proeng.2014.11.236

Scarpa, F., Lobba, A., & Becciu, G. (2016). Elementary DMA Design of Looped Water Distribution Networks with Multiple Sources. *Journal of Water Resources Planning and Management*, *142*(6), 04016011. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000639

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, *1*(1), 27–64. https://doi.org/10.1016/j.cosrev.2007.05.001

Sedgewick, R., & Wayne, K. (2011). *Algorithms* (Fourth). Addison-Wesley.

Shamir, U., & Howard, C. (1968). Water Distribution Systems Analysis. *Journal of Hydraulics Division*, *94*(HY1), 219–234.

Simpson, A., & Elhay, S. (2011). Jacobian Matrix for Solving Water Distribution System Equations with the Darcy-Weisbach Head-Loss Model. *Journal of Hydraulic Engineering*, *137*(6), 696–700. https://doi.org/10.1061/(ASCE)HY.1943-7900.0000341.

Simpson, A., Elhay, S., & Alexander, B. (2014). Forest-Core Partitioning Algorithm for Speeding Up Analysis of Water Distribution Systems. *Journal of Water Resources Planning and Managem§ent*, *140*(4), 435–443. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000336.

Stanic, M., Avakumovic, D., & Kapelan, Z. (1998). Evolutionary algorithm for determining optimal tree layout of ater distribution networks. In V. Babovic & L. C. Larsen (Eds.), *Hydroinformatics 98* (pp. 901–910). Rotterdam: Balkema.

Tarjan, R. (1971). Depth-first search and linear graph algorithms. *12th Annual Symposium on Switching and Automata Theory (Swat 1971)*, *1*(2), 146–160. https://doi.org/10.1109/SWAT.1971.10

Todini, E. (1979). Un Metodo del Gradiente per la Verifica delle Reti Idrauliche. *Bolletino Degli Ingegneri Della Toscana*, (11), 11–14.

Todini, E. (2000). Looped water distribution networks design using a resilience index based heuristic approach. *Urban Water*, *2*(2), 115–122. https://doi.org/10.1016/S1462-0758(00)00049-2

Todini, E. (2006). on the Convergence Properties of the Different Pipe, 1–16.

Todini, E., & Pilati, S. (1987). A gradient method for the analysis of pipe network. *International Conference on Computer Applications for Water Supply and Distribution*, (October).

Todini, E., & Rossman, L. (2013). Unified Framework for Deriving Simultaneous Equation Algorithms for Water Distribution Networks. *Journal of Hydraulic Engineering*, *139*(5), 511–526. https://doi.org/10.1061/(ASCE)HY.1943-

7900.0000703.

Tzatchkov, V., Alcocer-Yamanaka, V., & Bourguett Ortíz, V. (2006). Graph Theory Based Algorithms for Water Distribution Network Sectorization Projects. *Water Distribution Systems Analysis Symposium 2006*, 1–15. https://doi.org/10.1061/40941(247)172

Vasilić, Ž., Stanić, M., Kapelan, Z., Ivetić, D., & Prodanović, D. (2018). Improved Loop-Flow Method for Hydraulic Analysis of Water Distribution Systems. *Journal of Water Resources Planning and Management (ASCE)*, *144*(4), 1–11. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000922

WAA, & WRC. (1985). Water Authorities Association and Water Research Centre: Report 26 Leakage Control Policy & Practice.

Walsky, M. (1983). Journal (American Water Works Association),. *American Water Works Association*, *75*(2), 58–63.

Wood, D. J., & Charles, C. O. A. (1972). Hydraulic Network Analysis Using Linear Theory. *Journal of Hydraulics Division*, *98*(HY7), 1157–1170.

Wright, R., Stoianov, I., & Parpas, P. (2014). Dynamic topology in water distribution networks. *Procedia Engineering*, *70*, 1735–1744. https://doi.org/10.1016/j.proeng.2014.02.191

Zhang, Q., Wu, Z., Zhao, M., & Qi, J. (2017). Automatic Partitioning of Water Distribution Networks Using Multiscale Community Detection and Multiobjective Optimization. *Journal of Water Resources Planning and Management*, *143*(9), 1–14. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000819.

БИОГРАФИЈА АУТОРА

Жељко Василић је рођен у Ужицу 27.09.1986. године. Основну школу „Стари Град“ у Београду завршио је 2001. године као носилац дипломе „Вук Караџић“ и носилац звања „Ђак генерације“. Након завршене основне школе уписао је природно-математички смер „Шесте београдске гимназије“ који је завршио 2005. године са одличним успехом.

Грађевински факултет Универзитета у Београду, одсек грађевинарство, уписује 2005. године. Основне академске студије завршава 2009. године на одсеку за хидротехнику и водно еколошко инжењерство са просечном оценом 9,36/10 и стиче звање дипломирани инжењер грађевинарства. За одбрањени дипломски рад добио је награду Привредне коморе града Београда. Након основних студија уписује дипломске академске – мастер студије на Грађевинском факултету на одсеку за хидротехнику и водно еколошко инжењерство. Мастер студије завршава 2010. године са просечном оценом 9,83/10 и стиче звање мастер инжењер грађевинарства. Током школовања био је стипендиста Министарства просвете Републике Србије и града Београда.

Након завршених мастер студија, 2010. године уписао је докторске студије на Грађевинском факултету Универзитета у Београду. На Грађевинском факултету 2011. године заснива радни однос у звању асистента – студента докторских студија за уже научне области Механика нестишљивих флуида и Хидраулика, Хидроинформатика и Хидротехничке мелиорације и уређење сливова. Поред обавеза у настави, активно учествује и на изради техничке документације разних пројеката, студија и експертиза из области хидротехнике, које Грађевински факултет реализује преко Института за хидротехнику и водно еколошко инжењерство. Ангажован је као истраживач на научном пројекту ТР37010 „Системи за одвођење кишних вода као део урбане и саобраћајне инфраструктуре“.

Аутор је и коаутор четири рада на SCI листи, као и већег броја радова у домаћим часописима, на међународним и домаћим научним и стручним скуповима. Течно говори и пише Енглески језик.

Изјава о ауторству

Име и презиме аутора: Жељко Василић

број индекса: 908/10

Изјављујем

да је докторска дисертација под насловом:

DECISION SUPPORT ALGORITHMS FOR SECTORIZATION OF WATER DISTRIBUTION NETWORKS

(АЛГОРИТМИ ЗА ПОДРШКУ ОДЛУЧИВАЊУ ПРИ СЕКТОРИЗАЦИЈИ МРЕЖА ПОД ПРИТИСКОМ)

- резултат сопственог истраживачког рада,
- да предложена дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

Потпис аутора

У Београду, 26. septembar 2018.

____

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора: Жељко А. Василић

Број индекса: 908/10

Студијски програм: Грађевинарство

Наслов рада: DECISION SUPPORT ALGORITHMS FOR SECTORIZATION OF WATER DISTRIBUTION NETWORKS

(АЛГОРИТМИ ЗА ПОДРШКУ ОДЛУЧИВАЊУ ПРИ СЕКТОРИЗАЦИЈИ МРЕЖА ПОД ПРИТИСКОМ)

Ментор: В. проф. др Милош Станић, дипл. грађ. инж.

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла за ради похрањења у **Дигиталном репозиторијуму Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис аутора

У Београду, 26. септембар 2018.

____

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

DECISION SUPPORT ALGORITHMS FOR SECTORIZATION OF WATER DISTRIBUTION NETWORKS

(АЛГОРИТМИ ЗА ПОДРШКУ ОДЛУЧИВАЊУ ПРИ СЕКТОРИЗАЦИЈИ МРЕЖА ПОД ПРИТИСКОМ)

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство (CC BY)

2. Ауторство – некомерцијално (CC BY-NC)

[✓ 3. Ауторство – некомерцијално – без прерада (CC BY-NC-ND)]

4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)

5. Ауторство – без прерада (CC BY-ND)

6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци.
Кратак опис лиценци је саставни део ове изјаве).

Потпис аутора

У Београду, 26. септембар 2018.

____

1. **Ауторство**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. **Ауторство – некомерцијално**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. **Ауторство – некомерцијално – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. **Ауторство – некомерцијално – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. **Ауторство – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. **Ауторство – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.