THESIS ON CIVIL ENGINEERING F69

Real-Time Control Optimization of Water Distribution System with Storage

MARKUS SUNELA

TTÜ PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
School of Engineering
Department of Civil Engineering and Architecture

This dissertation was accepted for the defense of the degree of Doctor of Philosophy (Civil and Environmental Engineering) on October 16th, 2017.

Supervisors: Raido Puust, PhD
School of Engineering, Department of Civil Engineering and Architecture,
Tallinn University of Technology, Tallinn, Estonia

Anatoli Vassiljev, PhD
School of Engineering, Department of Civil Engineering and Architecture,
Tallinn University of Technology, Tallinn, Estonia

Opponents: Zoran Kapelan, PhD
College of Engineering, Mathematics and Physical Sciences,
University of Exeter, Exeter, United Kingdom

Luigi Berardi, PhD
Department of Civil Engineering and Architecture,
Polytechnic University of Bari, Bari, Italy

Defense of the thesis: November 24th, 2017

Declaration:
Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology has not been submitted for doctoral or equivalent academic degree.

Markus Sunela

ISSN 1406-4766
ISBN 978-9949-83-165-4 (publication)
ISBN 978-9949-83-166-1 (PDF)

EHITUS F69

Mahutitega veevarustussüsteemi juhtimise optimeerimine reaalajas

MARKUS SUNELA

TTÜ
KIRJASTUS

In memoriam

Mikael Juha Erkki Niemistö
★ 3rd April 1983 — †11th November 2013

My dear big brother, who died too young.
Inspirer.

CONTENTS

SUMMARY

A NOVEL, general framework for performing whole-cost optimization of water production and distribution in real-time was developed in this dissertation. Optimization enables significant savings in energy and chemical costs.

Optimization resulted in near optimal settings for all pump, valve and source stations, and optimal frequencies for all pumps in the water supply system as a function of time for the next 24 hours in near real-time.

This dissertation developed a novel way to formulate the design variables of the optimization problem in order to minimize the size of the search space, a novel way to preoptimize operation of pump batteries, a novel way to model pressure or flow controlled variable-speed driven pumping and a novel method to model complex control strategies in the hydraulic simulator.

The optimization algorithm used is a modified version of greedy, meta-heuristic, single-solution Hybrid Discrete Dynamically Dimensioned Search (HD-DDS), that has not been applied in operational optimization of water supply systems before.

According to the author's review of previous studies, this research is the first where real-time operative optimization of a large-scale water supply system (WSS) is performed using a non-simplified and non-surrogate model covering all pipes in the system, and where the raw water production, conveyance and treatment are also included in the model and optimization.

In the case study (Tampere Water) the proposed optimization framework resulted in 20 % savings in the production and distribution costs, while ensuring better quality of service than before. Real-time aspect is ensured by the optimization run taking about two hours of computation time.

KOKKUVÕTE

Käesolev doktoritöö esitleb uuenduslikku, üldistatud raamistikku, tervikliku veevõrgu maksumuse reaalajas optimeerimist, mis kaasab nii vee tootmistsüklit kui ka selle juhtimist tarbijani.

Optimeerimine võimaldab märkimisväärset kokkuhoidu nii energiakulu kui kemikaalide maksumuse seisukohalt.

Optimeerimise käigus leitakse optimumile lähedased kõikide pumpade, klappide, lähteallikate seaded ning kõikide veevõrgu süsteemi pumpade optimaalsed pöörete arvud järgnevaks 24 tunni perioodiks reaalajale lähedase aja jooksul.

Käesolev doktoritöö esitleb unikaalset, optimeerimiseks vajalike disaini parameetrite defineerimise võimalust, et minimeerida lahendite ruumi; uudset pumplate eeloptimeerimist; uuenduslikku pöörete-arvu reguleerimisega pumba elementide modelleerimist lähtuvalt fikseeritud surve või vooluhulga tagamisest ning uudset lähenemisviisi kontrollimaks hüdraulilise simulaatori keerukamaid juhtimise strateegiaid.

Optimeerimisalgoritm on tuletatud *hybrid discrete dynamically dimensioned search (HD-DDS)* baasil, mida pole varasemalt veevõrgu süsteemi opereerimise optimeerimise juures kasutatud.

Lähtuvalt autori poolt teostatud kirjanduse ülevaatest hõlmab käesolev uurimustöö esmakordselt suuremahulise veevõrgu reaalajas juhtimise optimeerimist, kus mudel kaasab kõiki süsteemi torusid ning toorvee tootmine, transport ning töötlus on samaaegselt kaasatud mudelisse ning ka selle optimeerimisse.

Doktoritöö raames rakendati Tampere linna veevõrgule väljatöötatud lahendust, kus optimeerimine andis 20 % kokkuhoiu vee tootmise- ja vee transpordi kuludelt, tagades samal ajal varasemast parema teenusekvaliteedi. Reaalaja terminit kaasatakse optimeerimises lähtuvalt asjaolust, et optimeeritud lahendi otsing võtab ligikaudu kaks tundi aega.

YHTEENVETO

TÄSSÄ väitöskirjassa kehitettiin uusi, yleinen ratkaisu vedentuotannon ja -jakelun kokonaiskustannusten optimoimiseksi reaaliaikaisesti. Optimoinnin avulla on mahdollista saada aikaan merkittäviä säästöjä energia- ja kemikaalikuluissa.

Optimoinnilla haetaan asetusarvot kaikille asemille ja optimitaajuudet kaikille vedenjakelujärjestelmän pumpuille ajan funktiona aina 24 tuntia optimointihetkestä eteenpäin lähes reaaliajassa.

Väitöstutkimuksessa on kehitetty uusi tapa muotoilla optimoinnin suunnittelumuuttujat hakuavaruuden minimoimiseksi, uusi tapa esioptimoida pumppupattereiden toiminta, uusi tapa paine- tai virtaussäädetyn, taajuusmuuttajaohjatun pumppauksen mallintamiseksi sekä menetelmä monimutkaisten säätötapojen mallintamiseksi verkostosimulaattorissa.

Työssä käytetään optimointialgoritmina muokattua versiota ahneesta, metaheuristisesta, yhtä ratkaisua käsittelevästä hybrid discrete dynamically dimensioned search (HD-DDS) -optimointialgoritmista, jota ei ole aiemmin sovellettu vedenjakelujärjestelmän operatiivisissa optimoinnissa.

Tekijän kirjallisuusselvityksen perusteella tämä tutkimus on ensimmäinen, jossa reaaliaikaista operatiivista optimointia tehdään yksinkertaistamattomalla koko verkoston kattavalla mallilla, jossa on mukana myös raakavedentuotanto ja vedenpuhdistusprosessit.

Tapausesimerkissä (Tampereen Vesi) optimoinnilla saatiin aikaan 20 %:n säästö tuotanto- ja jakelukustannuksissa samalla, kun palvelutaso parani. Vuorokauden aikajakson optimointi vaati tapausesimerkissä noin kaksi tuntia laskenta-aikaa.

LIST OF PUBLICATIONS

I Sunela M I, Puust R ; **Simple visual tool to analyse pump battery efficiencies for various pump combinations.** *Procedia Engineering*, 89, 2014, pp. 525–532

II Sunela M I, Puust R ; **A visual tool to calculate optimal control strategy for non-identical pumps working in parallel, taking motor and VSD efficiencies into account.** *Water Science and Technology: Water Supply*, 15(5), 2015, pp. 1115–1122

III Sunela M I, Puust R ; **Modeling water supply system control system algorithms.** *Procedia Engineering*, 119, 2015, pp. 734–743

IV Sunela M I, Puust R ; **Real Time Water Supply System Hydraulic and Quality Modeling – A Case Study.** *Procedia Engineering*, 119, 2015, pp. 744–752

V Sunela M I ; **Real-time whole-cost operational optimization of water production and distribution (in Finnish).** *Vesitalous*, (3), 2016, pp. 13–20

VI Sunela M I, Puust R ; **Real-time whole-cost operational optimization of water production and distribution.** *CCWI 2017*, 2017

Summary of author's contributions

I Organizer and main writer for Publication I.

II Organizer and main writer for Publication II.

III Organizer and main writer for Publication III.

IV Organizer and main writer for Publication IV.

V Single author for Publication V.

VI Organizer and main writer for Publication VI.

LIST OF CONFERENCE PRESENTATIONS

I Sunela M I, Puust R; **Simple visual tool to analyse pump battery effciencies for various pump combinations**; 16^{th} Conference on Water Distribution System Analysis, Bari, Italy, July 14–17, 2014

II Sunela M I, Puust R; **Modeling water supply system control system algorithms**; 13^{th} International Computing and Control for the Water Industry Conference, Leicester, United Kingdom, September 2–4, 2015

III Sunela M I, Puust R; **Real Time Water Supply System Hydraulic and Quality Modeling – A Case Study**; 13^{th} International Computing and Control for the Water Industry Conference, Leicester, United Kingdom, September 2–4, 2015

IV Sunela M I; **Real-time whole-cost operational optimization of water production and distribution**; 60^{th} National Water Services Days, Hämeenlinna, Finland, June 8–9, 2016

V Sunela M I, Puust R; **Real-time whole-cost operational optimization of water production and distribution**; 15^{th} International Computing and Control for the Water Industry Conference, Sheffield, United Kingdom, September 5–7, 2017

RELEASED SOFTWARE

THIS research resulted in the following software tools that are also used in other similar projects:

Water Supply & Distribution Optimizer – a general real-time operational optimization solution for water supply systems

Parallel Pumping Optimizer – a multi-platform tool for optimizing parallel pumping. The software was used for optimizing pumping in the case studies of Publication I and Publication II, and was integrated into EPANET[255]. The optimizer is also used as a part of the optimization process presented in this dissertation.

Enhanced EPANET – a multi-platform hydraulic modeling software package and library based on EPANET hydraulic simulator. Enhancements include pump battery component[256], control system modeling[252, 256], parallel pump optimizer, support for multi-threading and computation speed optimizations.

PREFACE

BEFORE you lies the culmination of several years' worth of labor and research into pumping, electrical motors, and water supply systems (WSS), their performance characteristics and optimization.

This PhD thesis is part of my doctoral studies at Tallinn University of Technology, Estonia. The research was conducted at the request from the Tampere Water Utility.

Main motivation for the research was that no such a study had earlier been applied in Finland, and that the earlier work done elsewhere neglected some of the complexeties of the optimization problem of the water supply system (WSS).

This work is expected to show that it is possible to optimize large-scale WSSs using full-scale hydraulic models and to include all components affecting energy usage and efficiency, while still achieving near real-time performance.

Solving the complex multi-part problem resulted in multiple new tools and products, new business, and proved to be very rewarding.

I hope this work is of interest and will be improved upon by other researchers.

ACKNOWLEDGEMENTS

First of all I want to thank the Tampere Water Utility and its personnel Heikki Syrjälä, Pekka Laakkonen, Tapio Soini, Heidi Rauhamäki, Sini Vuorinen, Pekka Pesonen, Veli-Ville Vihersalo, and Matti Virkamäki, for financing the thesis in a large part, and providing me with all the data, information, projects and support during the years. Lasse Hietala from Insta Automation has also been very helpful during the project.

Jukka Meriluoto recruited me, and gave green light to my development ambitions at the beginning of my professional career. Perttu Hyöty and later Jouni Hyypiä, together with Jani Sillanpää have supported, defended and marketed both my research and my development activities tirelessly.

The initial push and support was given by my mentors and early clients: Hanna Riihinen (everything I know I learned from you), Antti Smolander, Jarmo Antikainen, Matti Heikkinen, Timo Ranta-Pere and Jukka Tyrväinen. Some of my closest colleagues requiring special mentions include Kalervo Aho, Jani Levonen, Minna Maasilta, Jukka Sandelin, Mika Kuronen, Janne Väyrynen, and Hannu Sippola.

Saila Kallioinen kindled interest in hydromechanics in me. Without Professor Tuula Tuhkanen I probably would not have done my exchange studies at Tallinn University of Technology. Professors Dragan Savic and Angus Simpson introduced the wonderful world of water supply system optimization to me.

Research and guiding of late Professor Tiit Koppel and my supervisors Raido Puust and Anatoli Vassiljev have been most helpful. The whole staff and all students at the Department of Civil Engineering and Architecture, especially Ene Pähn, have made my studies and research not only pleasant but also possible altogether.

My wife, Ulla, and children, Sohvi and Iipo, have endured the long hours I have been working, being always there for me. Thank you.

Ulla's sisters, and my mother-in-law, Merja Sunela, have encouraged me and supported my family at the times of need, for which I am grateful.

Without the endless encouragement to study, explore and experiment — and push to learn computers and programming — coming from my parents, Riitta and Matti Niemistö, I would have never pursued academics. They have been my early educators and guardians. The first half of my life was spent experimenting and learning together with my brothers, Tuomo and Mikael, who passed away during this project.

ACRONYMS

ACO	ant colony optimization
AMALGAM	a multialgorithm, genetically adaptive multiobjective method
ANN	artificial neural network
API	application programming interface
AR	auto-regressive
ARIMA	auto-regressive integrated moving average
BB	branch and bound
BEP	best efficiency point
BONMIN	basic open-source non-linear mixed integer programming
CLR	common language runtime
COIN-OR	computational infrastructure for operations research
CPU	central processing unit, processor
DAN2	dynamic neural network
DDS	dynamically dimensioned search
DE	differential evolution
DE	dynamic evolution
DMA	district metering area
DP	dynamic programming
EA	evolutionary algorithm
EGGA	enhanced global gradient algorithm
EPA	U.S. Envinronmental Protection Agency
FCV	flow control valve
FSP	fixed speed pump
GA	genetic algorithm
GAMS	general algebraic modeling system
GCC	GNU C Compiler
GEP	genetic expression programming
GGA	global gradient algorithm
GHEST	genetic heritage evolution by stochastic transmission

GMPA	graph matrix partitioning algorithm
GNU	GNU's not Unix
GPU	graphics processing unit
GRG	generalized reduced gradient algorithm
HBMOA	honey bee mating optimization algorithm
HD-DDS	hybrid discrete dynamically dimensioned search
HS	harmony search
HW	high water level
JAX-RS	Java API for RESTful web services
JDBC	Java database connectivity
JNI	Java native interface
JSF	Java server faces
JVM	Java virtual machine
LEM	learnable evolution model
LEMMO	multi-objective learnable evolution model
LIMS	laboratory information system
LMA	Levenberg–Marquardt algorithm
LP	linear programming
LT-GA	linear theory global algorithm
MARS	multivariate adaptive regression splines
MA	memetic algorithm
MA	moving-average
MHD-DDS	modified hybrid discrete dynamically dimensioned search
MINLP	mixed integer non-linear programming
MOGA	multi-objective genetic algorithm
MPC	model predictive control
MPI	message passing interface
MW	mean water level
NLP	non-linear programming
NRW	non-revenue water
NSGA-II	non-dominated sorting genetic algorithm
NW	low water level
ODBC	open database connectivity

OLS	ordinary least squares
OPC	open process contro
OpenCL	open computing language
OpenMP	open multi-processing
OPUS	optimal power use surface
ParEGO	extended efficient global optimization
PID	proportional integral derivative
PLC	programmable logic controller
PPR	projection pursuit regression
PRV	pressure reducing valve
PSO	particle swarm optimization
QoS	quality of service
REST	representational state transfer
RMSE	root-mean-square error
RTX	(EPANET) real-time extension
SA	simulated annealing
SBB	simple branch and bound
SCADA	supervisory control and data access
SIMD	single instruction multiple data
SLFA	shuffled leaping frog algorithm
SLGA	self-learning genetic algorithm
SOM	self-organizing map
SPEA2	strength pareto evolutionary algorithm
SSP	single speed pump
STA	state transition algorithm
SVR	support vector regression
TLS	thread local storage
TS	tabu search
UML	unified modeling language
UUID	universally unique identification
VS-SVR	variable-structure support vector regression
VSD	variable-speed drive
VSP	variable-speed pump

WDN water distribution network

WDS water distribution system

WSS water supply system

XML extensible markup language

1 INTRODUCTION

1.1 General

During the last years growing consideration is attached to the so-called water-energy nexus. Producing potable water and supplying it to the users requires a considerable amount of energy. No matter how energy is produced, the production always requires water at some stage. Typically, this water is not available for further consumption.

Energy is one of the largest expenses for water utilities, usually the second right after wages. Water supply uses 2 % to 3 % of the total energy consumption globally. Pumping water is the main energy consumer, using up to 80 % of the energy used in water supply systems. [46]

A more complete picture of the energy use of a water supply system can be painted by calculating energy balance for the system. [58, 57] The somewhat extended version of energy balance methodology is presented by the author in [255]. Hydraulic model is used for calculating energy use components as shown in Figure 1. The balance can be calculated for the whole supply system or any part of it. The energy balance, however, does not include any other energy consuming operations in water treatment processes besides transferring the water.

Consumers require a certain amount of energy, $E_{required}$. That energy is the actual useful part of the total hydraulic energy input E_{input} into the system. Losses occur in the network both due to the friction $E_{friction}$ and leakage E_{leak}. The energy input comes into the system in the form of potential energy or it is produced by the pumps E_{pump}. Pumps convert electricity into hydraulic energy only partially because of hydraulic losses occurring in the pump itself, $E_{hyd.losses}$, and losses in the motor and the variable-speed drive (VSD) $E_{motorlosses} + E_{VSDlosses}$, and thus electricity consumption is larger than the hydraulic energy input of the pump into the system.

Some energy use parameters gleaned from Saviranta [237] are presented in Table 1. The data represents five different Finnish water supply systems, each serving 20 000 to 250 000 inhabitants. On average, the electrical efficiency, $\eta = \frac{E_{required}}{E_{electrical}}$, is about 36 %; so there is clearly room for improvement.

In the systems examined, the raw water extraction and treatment used 14 % to 22 % of the total hydraulic energy use in the system, which is a clear indication that the water production energy use cannot be neglected, even if the focus is

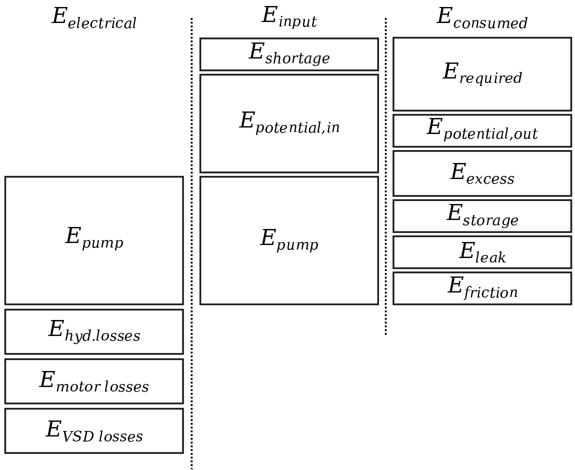


Figure 1. Energy balance components[255]

on the water distribution network. Likewise, hydraulic losses in pumps account for 29 % of the electrical energy input, and motor and VSD losses account for additional 12 %. In practice, the motor and VSD efficiencies must be accounted for when optimizing network operations to obtain correct results.

Table 1. Selected energy use parameters in Finnish water supply systems[237]

Parameter	Unit	Median	60 % Confidence Interval
Specific hydraulic energy use	kWh/m³	0.37	0.31–0.36
Specific electrical energy use	kWh/m³	0.45	0.45–0.50
Hydraulic efficiency	%	49	47–52
Electrical efficiency	%	36	32–38
Pumping total efficiency	%	59	52–61
Hydraulic losses in pumps	%	29	22–40
Pump motor and VSD losses	%	12	10–13
Excess energy delivered to users	%	18	18–22
Friction energy losses	%	19	17–22
Energy loss due to leakage	%	10	9–13
Hydraulic energy used for raw water extraction and treatment	%	19	14–22

The classification of the energy efficiency of an electrical motor is covered in IEC 60034-30 [123] standard. The standard introduces three efficiency classes of the international standard: IE3 premium efficiency, IE2 high efficiency and IE1 standard efficiency. IEC 60034-31 [124] introduces preliminary limits for the IE4 super-premium efficiency class. The minimum nominal efficiencies required by the classes are shown in Figure 2.

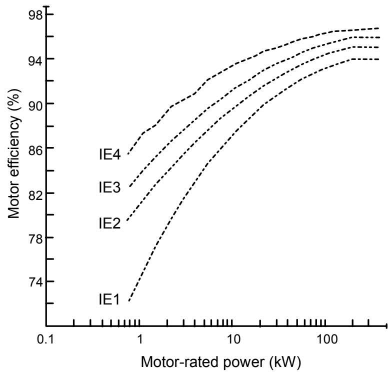


Figure 2. Lower nominal efficiency limits for different four-pole motor sizes as per IEC energy efficiency classes IE1–IE4. [73]

European Union Commission regulation EC/640/2009 [85], implementing Directive 2005/32/EC [86], requires that all motors (0.75 kW to 375 kW) have at least IE3 efficiency class or IE2 with variable speed drive, starting from 1st January 2017. From 1st January 2015, pumps with nominal power 7.5 kW to 375 kW had to meet the same limits. Comparative requirements have been applied in USA in 2010 [66]. These regulatory actions promise significant energy savings in the longer term. However, a large body of installed equipment remains, and optimization can reduce the energy usage for the older pumps and motors.

Besides considerable and immediate economic benefits, reducing energy consumption lowers the water utility's environmental impact, as energy production causes negative environmental effects and only a small percentage of the primary energy input is converted into kinetic energy of water, as shown in Figure 3.

Good design can save up to 30 % of the energy demand, but when the system is already functional, many aspects are fixed for long periods of time and cannot be easily or economically changed. The optimal design should also account for the specifics of the system, such as variable flow and head. The greatest energy savings can be achieved when energy usage optimization is incorporated into urban planning and the water supply system design from early on. [139]

Considerable amount of research has been done in the field of optimal network design. Also, redesign and replacement of pumps, building new storage capacity or new mains, optimizing the water treatment process or parallel pumping – all of these measures have significant savings potential.

Operational optimization changes the control settings and parameters, or the control algorithms the water supply control system uses for operating the vari-

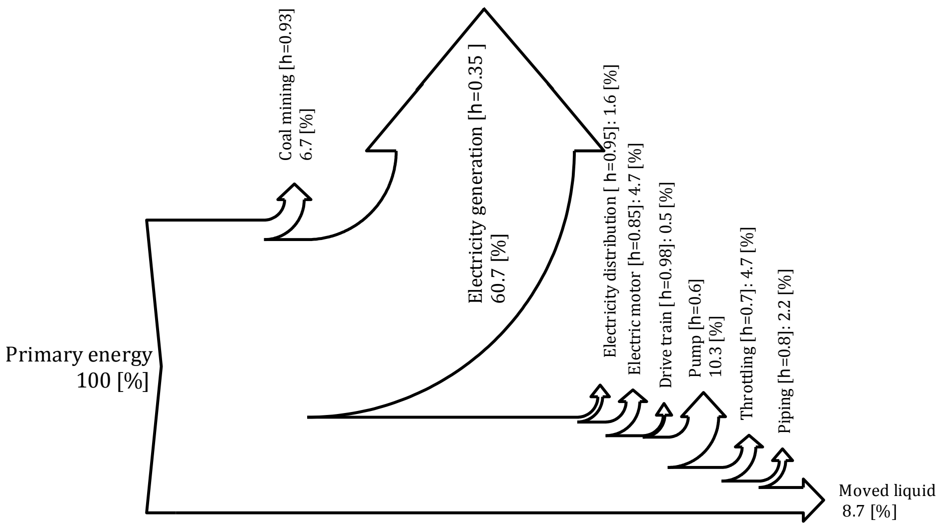


Figure 3. Typical energy flow from primary energy to kinetic energy of water [279]

ous pumps, valves and tanks installed in the system. Optimizing the operational aspects of the system is attractive, because significant benefits can be achieved in multiple fronts with simple changes to the control system or its parameters, and no investments in the network or equipment are needed.

The operational optimization, often named pump scheduling, finds parameters that result in optimal behavior in terms of total operational cost, energy consumption, water quality, system reliability or environmental impact, namely greenhouse gas emissions. These goals can be contradictory in part, and the conflicting goals are either formulated into constraints or penalties, or as a multi-objective optimization problem.

Often the water utilities have multiple sources of water with varying production costs. This adds an additional aspect to the optimization problem. It can be cheaper to produce water in a far away source, even though the energy costs may be higher than producing the water closer to the demand.

Usually the price of electricity is not fixed and can vary based on the time of the day, weekday, season, location and peak consumption. The price of electricity does not necessarily reflect the environmental effect of the energy production. Both wind and solar energy have better availability during daytime, but typically electricity is the cheapest off-peak, especially during night. Thus, lowering the energy consumption is always beneficial in terms of both economics and environmental impact, but minimizing energy costs and environmental impact may be conflicting goals.

One example of the complexity and size of the water network optimization problems is presented in [50] where a small case-study included four pumping stations having a total of 10 pumps and one valve, and five chlorine dosing locations. The search-space for the optimal settings for each pump, valve and chlorine dose was $8.9 \cdot 10^{28}$ in the study. Enumerating such a large number of solutions is impractical, and the search-space grows exponentially, as more stations are included in the optimization. Numerous numerical optimization methods have been developed by different authors to cope with such large search-spaces and non-linearity of the water supply systems.

1.2 Objective of the thesis

Focus in this thesis research is on the following questions:

1. Which components affecting energy use are typically missing from the operational optimization problem solutions, and how can they be included?
2. Can near real-time optimization be performed using a full-scale, all-pipe network model, including raw water extraction, conveyance and treatment, and an accurate pump energy model?

The main objective of this thesis research is to develop a method for near real-time whole-cost optimization of the operation of the water supply system (WSS) containing elevated storage and variable-speed driven pumps. Optimization has to take into account every pumping that happens in the system and all factors affecting the pumping efficiency and system energy consumption, including raw water extraction and conveyance, which are not included in the earlier research. The cost optimization must not violate the quality of service (QoS) constraints. The cost does not only include the cost of energy, but also water treatment costs at individual sources.

The optimization will be done on multiple interconnected layers all at once:

1. water distribution system (WDS) level: from where, where to and how much water is pumped or conveyed through valves in order to meet the water demand and hydraulic and operational constraints
2. water treatment level: what the production costs at different sources are
3. raw water extraction and conveyance level: what the energy costs of extraction and conveyance are
4. pump battery level: how the individual pumps working in parallel will be driven to achieve best efficiency

The design variables, or output from the optimizer, are the time-varying flow and pressure settings for all stations in the water supply system, and optimal frequencies for every pump in the system over the optimization horizon of 24 hours.

The hydraulic state including energy consumption and constraints is evaluated using a modified and extended version of the EPANET [226] simulator, originally developed by the Environmental Protections Agency of U.S.A. The constraint violations are handled using the penalty function.

The optimization framework has to be generic and fast enough for real-time use. It includes an easy way to integrate it with various supervisory control and data access (SCADA) systems in use at different water utilities.

The specific research objectives of the thesis are:

1. Chapter 3.2: Development of an accurate model for pump, motor and variable speed drive combination accounting for all loss components as a function of rotational speed

2. Chapter 3.3: Development of a method for finding globally optimal frequencies for pumps running in parallel for the whole operational range of the pump battery that can be used for pre-optimizing batteries of pumps working in parallel at different stations in the system

3. Chapter 3.4.1: Development of an EPANET simulator component that allows accurate and efficient modelling of flow and pressure control of variable-speed driven pumps working in parallel

4. Chapter 3.4.2: Development of a methodology for modeling complex water supply control strategies in EPANET, for example controlling raw water extraction, conveyance and treatment, and network pumping

5. Chapter 3.7: Development of an efficienct formulation for optimizing the whole water supply system, including the production side

6. Chapters 3.8 and 3.11: Finding out or developing a custom meta-heuristic algorithm that can be used for optimizing the whole system, including the listed developments in a near real-time setting

7. Chapters 3.4.3, 3.4.4 and 3.11.1: Ensuring satisfactory computational time of the optimization by improving hydraulic simulation and objective function evaluation performance.

8. Chapter 3.12: Implementation of a generally useable framework combining the listed developments

1.3 Layout of the thesis

This thesis is divided into five chapters.

Chapter 1 provides a general overview of the domain and sets the objective of the thesis.

Chapter 2 reviews the relevant literature. The objective is to focus on the studies of optimization methods applied in the water supply system design and operation, efficiency measures of a water distribution system, and finally, pumping efficiency and optimization. Hydraulic modeling and water demand forecasting are briefly described, because they are important for the operative optimization process.

Chapter 3 presents the real-time optimization framework and related developments in this thesis research. For example, extensions developed for EPANET, parallel pump optimization methodology, and SCADA access method are described along with the optimization problem formulation and the optimization algorithm.

The optimization framework is applied in a full-scale case study in Chapter 4. The chapter presents the case and relevant results.

Finally, conclusions and discussion, along with some future research paths are presented in Chapter 5.

2 WATER SUPPLY SYSTEMS

2.1 Introduction

Optimizing the operational aspects of a water supply system can yield considerable energy and cost savings. The optimization frameworks, however, tend to be complex systems of multiple components. Schematic presentation of a generic framework for the operation optimization of a water supply system is shown in Figure 4. First, a hydraulic simulator is needed for simulating the behavior of the system under various conditions. A demand forecast method is used for approximating the future water demand in various parts of the system to be optimized for the optimization network. Finally, an optimization algorithm drives the optimization process in order to find a nearly optimal solution.

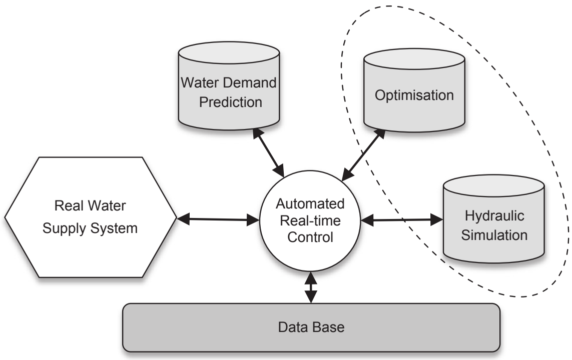


Figure 4. Framework for optimal operation of a water supply system using a supervisory control and data acquisition (SCADA) system. [65]

This chapter first provides a general overview of water supply systems, and then proceeds with a literature review related to the optimization and its constituents parts: hydraulic simulation, demand forecasting, design optimization, pump and pumping optimization, and finally operational optimization.

2.2 Water supply system structure

Water supply system produces and delivers potable water to the consumers using a complex network of pipes, pumps, valves, tanks, and treatment plants. According to Walski et al. [287], water supply system (WSS) can be divided roughly into three parts: water sources, water distribution network (WDN) and water consumers. All the parts are briefly discussed below and shown in Figure 5.

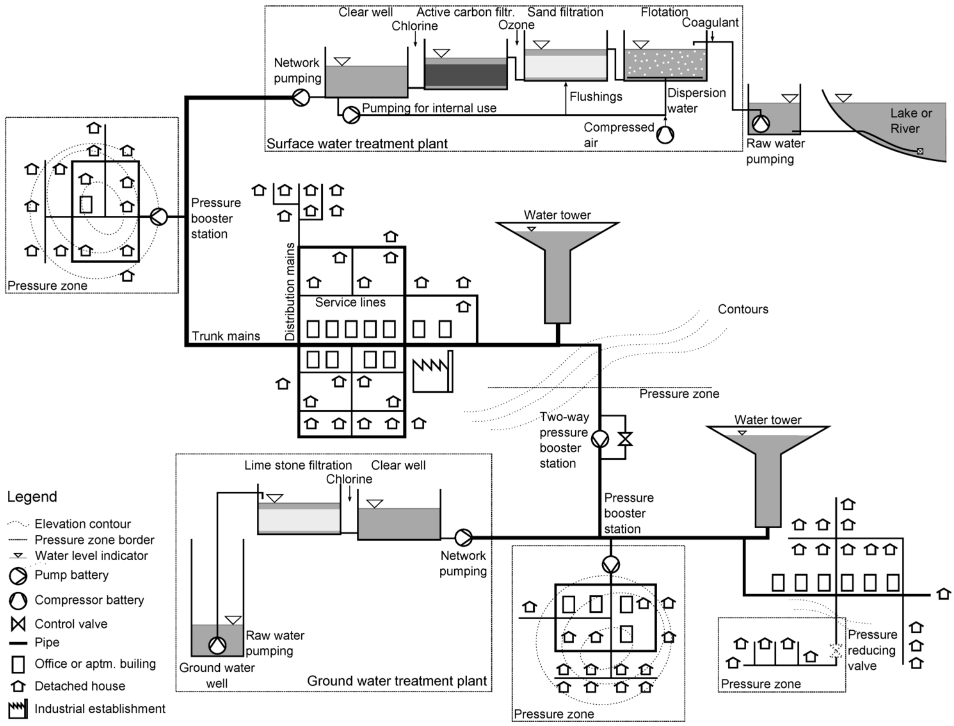


Figure 5. Schematic drawing of a water supply system with multiple sources, water towers and pressure zones

Potable water is produced from raw water at water treatment plants. Raw water is pumped to the water treatment plant, where its chemical and microbiological properties are changed so that the treated water meets all the requirements and recommendations set for potable water [6]. The exact treatment required depends on the source used and the properties of the raw water. Water typically flows gravitationally through the treatment process, where unit operations cause some potential energy losses.

Raw water can be surface water extracted from a lake or river, groundwater or artificial groundwater. Even sea water can be used as a raw water source if desalination is used.

Groundwater may be usable without any treatment, but groundwater treatment plants should have at least the disinfection readiness. Surface water typically requires more extensive treatment.

Treated water is stored in a clear well at the treatment plant. A clear well functions as a water source for the water distribution network, to which the water is delivered using pumps. A clear well also levels the changes in the water demand, chemical dosing, and improves reliability. Surface water treatment plants require some clean water, typically about 10 % of the production, for the treatment process, mainly for filter flushing. This water is normally taken from the clear well using pumps. [220]

The water distribution network (WDN) is formed by pipes of various sizes and materials dug in the ground connecting the water consumers with the water sources. WDNs are typically looped: there are several routes for water between any two points in the network.

WDNs typically include tanks that are used for leveling the differences in the water demand and ensuring that water supply functions in the case of electricity loss and pipe bursts. Because of ground elevation differences, WDN is divided into pressure zones that are connected using pumping stations (to raise the pressure) and control valve stations (to lower the pressure).

Moving fluid contains both potential and kinetic energy. The energy content is normally expressed as pressure in meters of water column relative to sea level or some other base elevation using Bernoulli's equation: [284]

$$H = z + \frac{p}{\rho g} + \frac{v^2}{2g} \quad , \tag{2.1}$$

where H is the total or energy head, z the elevation, p the pressure, ρ the density of the fluid, v the fluid velocity and finally, g acceleration due to gravity. The kinetic energy term $\frac{v^2}{2g}$ is usually very small compared to the potential energy $z + \frac{p}{\rho g}$ and it is thus often left out from the calculations.

Flow through a hydraulic element, for example, a pipe, depends on the energy difference between the ends of the element. Fluid flows always from the higher energy towards the lower. Besides the energy difference, certain physical properties of the element affect the flow. For example, pipe diameter, or more generally the cross-sectional area, has a major impact on the flow. [239]

Reynolds number

$$\mathrm{Re} = \frac{\rho v d}{\mu} = \frac{vd}{\nu} \quad , \tag{2.2}$$

where d is the pipe diameter [m] for round pipes, μ is the dynamic viscosity of the fluid and $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity, can be used for determining the flow regime. When $\mathrm{Re} < 2000$ flow is laminar, $2000 \leq \mathrm{Re} \leq 4000$ flow is transitional, and $\mathrm{Re} > 4000$ flow is turbulent. [287]

When a flow is laminar, the friction factor depends completely on the Reynolds number, and when a flow is fully turbulent, the friction factor depends mostly on the relative roughness. Thus, roughness does not affect the friction factor much, when flow is laminar, and the significance of viscosity becomes smaller when the flow is turbulent. [287]

Pressure loss h_L describes the loss of the head due to the friction in the pipe. According to the Darcy–Weisbach equation

$$h_L = f \cdot \frac{L}{d} \frac{v^2}{2g} = f \cdot \frac{8 \cdot LQ^2}{gd^5 \pi^2} \quad , \tag{2.3}$$

where L is the length of the pipe [m]. Friction factor f can be calculated using different methods, most common of which for turbulent flows are the Colebrook–White equation:

$$\frac{1}{\sqrt{f}} = -0.86 \cdot \ln \left(\frac{\epsilon}{3.7 \cdot d} + \frac{2.51}{\mathrm{Re} \cdot \sqrt{f}} \right) \tag{2.4}$$

and the Swamee–Jain equation:

$$f = \frac{1.325}{\left[\ln \left(\frac{\epsilon}{3.7 \cdot d} + \frac{5.74}{\mathrm{Re}^{0.9}} \right) \right]^2} \quad . \tag{2.5}$$

In both equations, ϵ is the roughness of the pipe [m. There are many other explicit approximations of Colebrook–White equation besides the Swamee–Jain equation. A through statistical review of different methods is presented in [98].

For a laminar flow, the Hagen–Poiseuille equation can be used for estimating the friction factor [226]:

$$f = \frac{64}{\mathrm{Re}} \quad . \tag{2.6}$$

Consumers are of utmost importance for the WSS – the whole system is built for serving the water demand of the consumers. Consumers include, for example, the inhabitants, industry and other companies, and public buildings, such as hospitals and schools, in the area served by the WSS. The consumers require that they always dispose the needed amount of safe and high quality water with high enough pressure. [287]

The amount of water consumed, demand, and both the spatial and temporal water demand distribution are central to the design, functioning and operating of a WSS. Water demand varies constantly for various reasons, such as consumer type, hour of day, weekday and season. The water supply system has to be able to meet the demand under all conditions. [287]

In practice, using hydraulic simulation is the only way to analyze how changes in different parameters affect the system as a whole. It is virtually impossible to measure all hydraulic parameters everywhere in the network, or make distributive tests on a live system without endangering the water supply, and often conducting such tests could be prohibitively expensive.

2.3 Pump energy use

Figure 6 shows a typical pumping system schematically. A pump is connected to an inlet reservoir via piping. The pump is driven by an electric motor, which may, in turn, be driven by a variable-speed drive (VSD). When the pump has a VSD, its rotational speed, and thus the flow and head generated, can be controlled programmatically. This kind of a pump is called a variable-speed pump (VSP). If no VSD is present or its setting is not to be changed, the pump is single or fixed speed pump (SSP or FSP).

The incoming electrical energy is transformed into mechanical rotation energy in the motor, and mechanical energy into hydraulic energy (pressure and velocity) in the pump. Piping leaves the pump and connects it to an outlet reservoir. [281]

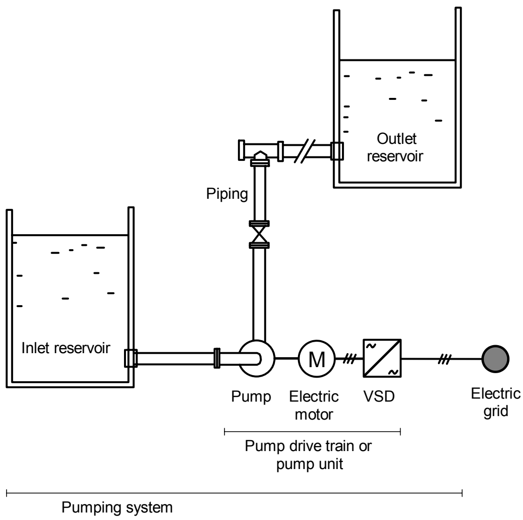


Figure 6. Pump drive train in a pumping system. Variable-speed drive may be included in the unit for rotational speed control. [279]

Energy losses occur in the pump itself, in the motor and in the VSD. The efficiencies of the components are typically in the range of 60 % for pumps, 85 % for motors and 95 % for VSDs, but the efficiencies vary based on the flow and speed. [33, 279]

Motors used in pumps consume 22 % of the electrical energy in the industry and 16 % in the service sector in EU. It is widely shown that using variable-speed pumps saves a considerable amount of energy: installing VSD would be beneficial for 33 % of the pumps in the industry and for 40 % in the services. [14, 164, 227, 228]

A pump's performance at its nominal rotational speed is described by two curves, one expressing the produced head H as a function of the flow Q (performance curve), and the other expressing the pump's hydraulic efficiency η_H or power P as a function of the flow Q (efficiency or power curve). The pump characteristic curves are provided by the pump manufacturer and they can also be independently measured. [281] An example of a typical set of curves is shown in Figure 7.

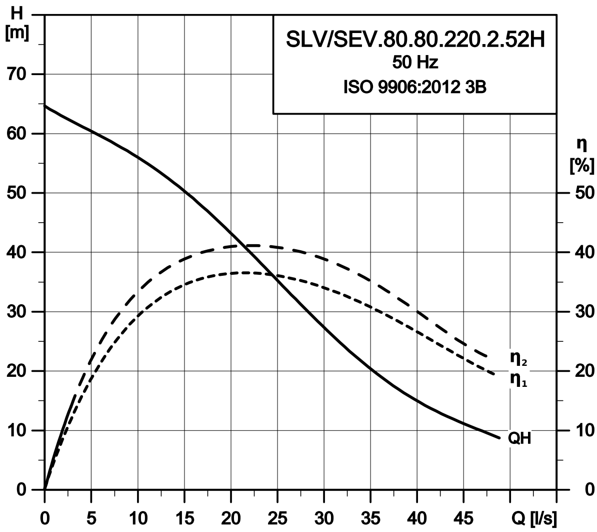


Figure 7. Example of pump characteristic curves for SLV.80.80.220 pump model as given by the manufacturer, Grundfos. The curves shown here include the performance curve QH, pump's hydraulic efficiency η_2 and pump's and motor's combined efficiency η_1. [7]

When either the head or the flow is known, the other can be looked up from the pump characteristic Q–H curve, and then the pump's hydraulic power P_H can be calculated:

$$P_H = \rho g Q H \quad . \tag{2.7}$$

By looking up the pump's hydraulic efficiency η_H from the pump curves, the pump shaft power is

$$P_S = \frac{P_H}{\eta_H} \quad . \tag{2.8}$$

Shaft power is the amount of power that the motor must produce.

Variable-speed drive (VSD) can change the motor's and thus the pump's rotational speed N. VSDs are introduced in order to control the produced flow and pressure in an energy efficient manner. The introduction of variable speed drives allows for significant energy savings and more flexibility in the control of pumping. [164, 227] The need for controlling pumping arises from significant variations in the water demand over time, and other changes in the system, like varying water tower levels.

Flow Q_2, head H_2 and power P_2 at some rotational speed N_2 are calculated using affinity laws, based on the known values Q_1, H_1 and P_1 at the nominal speed N_1 [281]

$$\frac{Q_2}{Q_1} = \frac{N_2}{N_1} \tag{2.9a}$$

$$\frac{H_2}{H_1} = \left(\frac{N_2}{N_1}\right)^2 \tag{2.9b}$$

$$\frac{P_2}{P_1} = \left(\frac{N_2}{N_1}\right)^3 \tag{2.9c}$$

While Equations (2.9a) and (2.9b) have been shown to be valid in a multitude of conditions, the last affinity law, Equation (2.9c), as it is, is shown not to describe the experimental data accurately. [e.g. 244] Thus, a more accurate model is required to describe the effect of the rotational speed on the efficiency.

Decrease in pump's hydraulic and overall efficiency at lowered pump rotational speeds has been reported by several authors. [244, 283, 99] Part of the observed efficiency loss compared to the affinity law is due to Equation 2.9c assuming a zero-head system, part due to the actual change in the pump's hydraulic efficiency curve when the rotational speed is reduced, part due to the lowering efficiency of the motor and the VSD on partial loads.

Various models have been developed to account for decrease in pump's hydraulic efficiency at lower rotational speeds. Gülich [105] states that it is complicated to solve the problem of efficiency scaling effectively because of considerable uncertainties in the process of predicting small difference between comparatively large figures accurately. There are, however, various methods to model the effect with reasonable accuracy for practical applications.

Gülich [105] proposes an accurate yet elaborate method that physically models the various efficiency affecting processes. A simpler, still a general method is presented in [250]. The proposed method is based on the friction factor f (see Equations (2.4) and (2.5)), and on the assumption that only part of the losses are dependent on it:

$$\frac{1-\eta_2}{1-\eta_1} = \frac{V + (1-V)\frac{f_2}{f_{\infty,2}}}{V + (1-V)\frac{f_1}{f_{\infty,1}}} \quad , \tag{2.10}$$

where V is the fraction of losses that depend on the friction factor, and f_∞ is the friction factor when $\mathrm{Re} = \infty$. Another common method to model efficiency scaling is based on the Reynolds number:

$$\frac{1-\eta_2}{1-\eta_1} = K + (1-K)\left(\frac{\mathrm{Re}_1}{\mathrm{Re}_2}\right)^m \quad , \tag{2.11}$$

where K is the fraction of losses that depend on the Reynolds number. Typical values for K range from 0.00 to 0.57 and m from 0.10 to 0.50 depending on the test data. Measurements are necessary to accurately model any specific pump. [293]

Traditionally, the motor efficiency is assumed to stay constant, especially in the 50–100 % load range [281, 269, 164], typically to simplify calculations. However, the motor's efficiency depends heavily on the load, and it is reported that there can be significant reductions in the efficiency even when the load is above 50 %, especially for small or low-efficiency motors. [55, 33] A more comprehensive review of the energy usage of an electrical motor is provided by Saidur [227].

The constant efficiency assumption can be valid when the load is close to the motor's nominal power, which is typically the case when the pump has no variable speed drive. The pump's hydraulic power, however, is inversely proportional to the relative speed cubed (see Equation 2.9c); thus, even small changes in the rotational speed can lower the power and the motor load considerably. This can cause considerable errors in the energy use calculations.

The exact motor efficiency η_M at different relative loads is motor specific, and the motor manufacturers provide load–efficiency curves. IEC 60034-31[124] standard also provides a general equation to calculate an approximation of motor efficiency at any partial load.

When the motor efficiency is known for the particular pump working point, the motor power

$$P_M = \frac{P_S}{\eta_M} = \frac{P_H}{\eta_H \cdot \eta_M} \tag{2.12}$$

can be calculated.

The VSD efficiency η_{VSD} depends on the relative load. IEC 60034-31[124] standard provides a lookup table for approximate efficiencies based on the nominal VSD power and rotational speed.

The final pump train electrical power consumption is

$$P_E = \frac{P_M}{\eta_{VSD}} \quad , \tag{2.13}$$

and the total pump train efficiency is

$$\eta_{TOT} = \frac{P_H}{P_E} = \eta_H \cdot \eta_M \cdot \eta_{VSD} \quad . \text{ [33]} \tag{2.14}$$

Frequency scaling and accurate modeling of a motor and variable speed drive efficiencies under different rotational speeds and loads are incorporated into the pump energy use model used in this thesis, as described in Chapter 3.2. According to the author's literature review, this results in the most accurate energy use model used in the operational optimization of water supply systems reported in the literature.

2.4 Optimizing parallel pumping

Studies focusing on optimizing variable-speed pumps working in parallel in water supply are scarce according to the literature review performed. Usually only single speed parallel pumping is considered as part of the pump scheduling problems, and only few sources mention variable speed pumping (see Chapter 2.8). Neither are there many papers tackling the parallel pump optimization problem separately from pump scheduling.

The methods used in the scheduling problems typically neglect many aspects affecting the pump energy use as outlined in the previous section, and they rely on the EPANET simulator for energy consumption calculations, even though EPANET is shown to give wrong efficiency and energy results when VSPs and reduced rotational speeds are used [165, 99].

Much of the VSP optimization research is related to heating, ventilation and air conditioning systems or control system engineering. These methods, three of which are presented next, avoid the use of mathematical optimization methods, and instead rely on heuristics, simplified system models and measurements done in real time, to facilitate easier implementation in programmable logic controllers (PLC).

Wang and Burnett [291] developed an adaptive and derivative control strategy for controlling heat exchanger pump pressure setting based on recursive least squares approximation of pump energy usage. The reported energy savings were 5 %.

Ma and Wang [157] developed several optimal strategies for controlling heat exchanger pumping in a building based on polynomial approximation of wire to water efficiency of the pumps. The method includes pump maintenance costs, but the optimization algorithms themselves are mainly heuristic. The optimal strategy using the optimal pressure differential set-points at the critical loops and optimal pump sequence control resulted in a savings potential of 12 % to 32 %.

Viholainen et al. [280] and [279] developed a reliable control method for parallel pumping based on the preferable operational area method. Based on each pump's measured flow and power measurements at each VSD, the working point for each pump is calculated, and a new reference speed is calculated, so that each pump would work inside the preferable area. The reported energy savings were 20 % to 25 %.

There are also some more generic, mathematical optimization based methods in the literature. Wu et al. [294] and Olszewski [183] used Genetic Algorithm (GA), Costa Bortoni et al. [68] used the dynamic programming method, and Yang and Borsting [304] and Koor et al. [143] both used the non-linear programming (NLP) method with Lagrange multipliers. All the other cited methods, except Wu et al. [294], can be quite easily implemented in PLC controlling the pump battery. In their respective models Wu et al. [294], Yang and Borsting [304] and Koor et al. [143], however, ignore the degrading effect of lower rotational speed on the pump hydraulic efficiency, and the motor and variable-speed drive efficiencies [105]. Costa Bortoni et al. [68] use penalty function to constrain the pumps to work close to their best efficiency points and the paper thus assumes that the motor and the VSD work in a high-efficiency range, and the motor and VSD efficiencies can be ignored. All except Koor et al. [143] allow the working pumps to work on different frequencies.

Koor et al. [142] build upon the earlier work presented in [143], and extend the methodology to work with non-identical pumps and to include frequency scaling. The method uses the Levenberg–Marquardt optimization algorithm (LMA) for calculating the optimal discharges for single pumps working in different working points.

Chapter 3.3 describes the parallel pump optimization method developed as part of this research and included in the optimization framework. Opposed to the earlier research, the method models accurately all energy loss components affecting the pumping, and uses exhaustive search to guarantee finding the globally optimal solution for minimizing the parallel pumping energy use for every possible working point of the pump battery. The parallel pump optimization process itself, however, is time-consuming and cannot be implemented in PLC, but the results can be used, for example, as a basis for regression model based control.

2.5 Hydraulic simulation

The aim of a hydraulic water supply simulation model is to calculate pressures, flows, the propagation of quality parameters, feeding of storage tanks, and the operation of pumps and control devices of the system under predefined loading conditions. [79]

Hydraulic simulation is an integral part of any water supply system optimization. Simulation is also typically the most time-consuming component of any optimization framework. Accuracy, available features, convergence and stability features of the simulator can pose limitations to the optimization methods available.

For optimal control of water supply pumping, hydraulic network modeling is done using one of the four main approaches. In mass balance models the head loss dependency on flow rates is neglected, and it is assumed that pumps work against constant head. Mass balance models, most often used in linear programming (LP) problems, are fast, but they do not guarantee hydraulic feasibility of the solutions. Regression models are based on a set of non-linear regression equations prepared for a specific water supply system and need to be reformulated if the system is modified. Simplified network hydraulic models are highly skeletonized versions of the full models [267, 191, 192]. Finally, full hydraulic simulation models include a set of quasi-steady-state hydraulic equations solved in terms of adjustment factors. Full hydraulic models are most accurate, but require considerably more computational resources than the other models. [186]

The rest of this section focuses on full hydraulic simulation models, as they are most general, accurate and widely used of the methods listed above.

Ormsbee [184] provides a good overview of the evolution of the hydraulic modeling. The article lists the most important methods to solve the flows and pressures in a water distribution system, starting with the Hardy Cross method [69], simultaneous node method, simultaneous loop method, linear method (simultaneous pipe method) and gradient method (simultaneous network method). Figure 8 shows various solutions developed for solving the hydraulic equations.

Hardy Cross method published in 1936 is an iterative method that can be manually calculated. The problem with the method is that initial guess for either the heads or flows has to be quite close to the final solution for the method to converge. The method was first computerized in 1957[156].

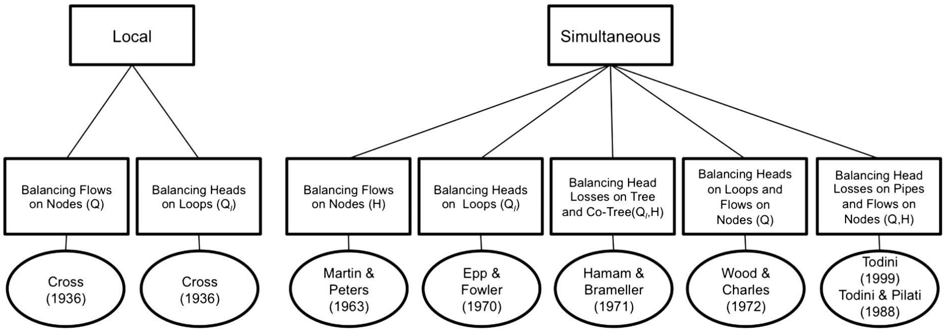


Figure 8. Different versions to the solution of the pipe network problem. Linearization variable is shown in parentheses. [262]

Currently, the most widely used hydraulic simulator [80] is EPANET [226] that uses the Global Gradient Algorithm (GGA) presented in [260]. The algorithm solves flows and heads simultaneously.

There has been renewed interest in rebooting EPANET development as a real open source project instead of being developed only in the U.S. Envinronmental Protection Agency (EPA). Some ideas for future development are presented in van Zyl and Chang [270], Rossman and van Zyl [223], and Rossman [222].

The latest official version of EPANET was released in 2008. Since then multiple corrections and enhancements have been published, but not incorporated into a common code base. Finally the converences of Water Distribution System Analysis (WDSA), 2014 and Computing and Control for Water Industry (CCWI), 2015 resulted in the creation of EPANET Open Source Initiative [38] and related code repository at https://github.com/OpenWaterAnalytics/EPANET.

Several ports of EPANET in different programming languages exist, like Python [249], C# [21], C++ [271, 108, 116], Java [1] and even Java Script that can be run in a web browser. Even without port to a language, EPANET toolkit can still be called from various other languages like Visual Basic [63] or Matlab [207].

Todini [259] and [262] provide a thorough analysis of the various algorithm formulations and their convergence properties. When choosing the solver, other factors besides the convergence speed should be evaluated: the size of the invertable matrix, symmetricity of the solution matrix, the matrix density, whether a fundamental set of loops must be identified, and whether a balanced set of initial flows is required. Based on these criteria, GGA and a new linear theory global algorithm (LT-GA) presented in [262] emerged as the most suitable and robust algorithms.

This thesis research uses an enhanced version of the latest publicly available version of the EPANET simulator. The enhancements are described in more detail below. In this work, the water supply system is modeled completely, including all pipes, raw water extraction, conveyance and treatment, and control system model, and the complete model is directly used in the optimization process, which is uncommon in previous studies. Extra steps are taken to ensure the hydraulic stability of the model under insufficient or excessive water supply by adding extra reservoirs to the system along with penalty costs. The model structure is described in Chapters 3.5 and 3.4.2.

2.5.1 Reducing simulation time

Because of the non-linear nature of water supply systems, iterative methods are used in hydraulic models. This increases the computational time required by the simulation. Various methods have been developed in order to reduce the computational time.

One popular method is the use of surrogate models reviewed in [215]. Surrogate modeling or meta-modeling replaces the computationally intensive model with simpler approximation. The performance of the surrogate model must be carefully assessed, especially for the critical points in the network, because approximation lowers the accuracy of the model.

Typically used surrogate modeling methods include the use of mass-balance models: e.g., [88], [160] and [29], the use of artificial neural networks (ANN): e.g., [168], [212] and [50], and model simplification or skeletonization: e.g., [267], [240], [191] and [19]. Behandish [30] uses the Graphics Processing Unit (GPU) based ANN surrogate model. The reported speed-ups compared to full-scale EPANET based hydraulic simulation can be up to 25 [234]. More creative use of a meta-model is reported in Chang and van Zyl [60] where compression heuristic method is introduced: only critical periods are simulated using a full-scale model and the meta-model is used otherwise, resulting in 8.8 times speed-up.

Van Zyl et al.[273] use two-point linearization instead of the tangent method. The proposed method results in better approximations for flows than the Newton–Raphson method used in EPANET, and thus reduces the number of iterations required. The method, however, is a trade-off between speed-up and accuracy.

More recent developments in speeding up the hydraulic simulation use various partitioning and decomposition algorithms, which deliver promising performance. However, no generally available mature implementations exist.

Alonso et al. [15] introduced the parallel EPANET solver based on graph decomposing. The methodology also runs quality simulation in parallel and synchronously to hydraulic simulation (in the normal EPANET, quality simulation can only be run after hydraulic simulation).

Deuerlein [79] has developed a general decomposition model for WDN. One of the presented applications was the high-performance hydraulic simulation. The Schur complement domain decomposition was used in [80]. The articles show 4 to 8 time speedups compared to EPANET.

Deuerlein et al. [76] developed the Graph Matrix Partitioning Algorithm (GMPA). The method reduced the problem size for different networks by 80 % on average. Abraham and Stoianov [11] have used partitioning and sparse null space algorithm to only update the changed matrix elements. The article reports computational time savings of up to 68 % over the Schur decomposition method.

Giustolisi et al. [104] introduced the Enhanced Global Gradient Algorithm (EGGA), which reduces the problem size by transforming the network topology while preserving the energy and mass balances. The computational time was reduced up to 90 % compared with GGA, as implemented in EPANET.

Luvizotto et al. [155] have introduced an interesting new method that avoids matrices altogether, but currently the method is two orders of magnitude slower than EPANET, even though it lends itself easily to parallel processing.

Paluszczyszyn et al. [193] developed a proof-of-concept hydraulic simulator based on quantized state system methods. The benchmarks show that event-based simulation is much faster on small networks than EPANET. Work is still needed to develop an actual hydraulic simulation tool based on the methodology.

Other attempts to improve the simulation speed include offloading the matrix calculations to the GPU. GPUs are highly parallel and very efficient in solving matrix equations. [208] The downside with the approach is that it is time-consuming to move the matrices between the computer's main memory and GPU, so that any benefits are lost if the network is not very large.

Guidolin et al. [109], [110] and [301] have explored the possible performance gains using single instruction multiple data (SIMD) instructions or GPU for hydraulic simulators in CWSnet and EPANET, respectively. None of these were able to achieve much improvement. Only little of the total simulation time is spent in the linear solver, as shown in Table 2; thus, even significant performance improvements in the code yield only small performance gains.

Various decomposition methods and other more efficient algorithms such as GMPA and EGGA would yield considerable performance improvements. Unfortunately, no publicly available robust and free implementations of those exist. As parallelizing the matrix solvers in EPANET either using GPU or multiple CPU cores does not seem to yield considerable performance, this work uses the normal EPANET with custom enhancements.

Consistent with the goals of the thesis research, this work uses no surrogate models, but instead makes use of a full-scale model, extended with raw water extraction, conveyance and treatment, aiming at maximal accuracy. While sur-

Table 2. CPU time allocation solving a 150 000 pipe hydraulic model with EPANET[301]

Computing Task	Time [s]	Time [%]
Total	67.45	100.0
Open Model	7.77	11.5
Solve Model	58.59	86.9
Solve Hydraulics	54.56	80.9
Demands & Controls	0.04	0.1
Network Solver	20.05	29.7
Linear Solver	8.94	13.3
Hydraulic Status	0.33	0.5
Save Results	4.04	6.0
Report	0.00	0.0
Close Model	1.08	1.6

rogate models offer great performance benefits, their preparation and validation are laborious and system specific processes, and their use cause inaccuracies, especially in the energy distribution [191].

In this work, sufficient computational performance for near real-time optimization is ensured by utilizing the most aggressive and modern compiler optimization techniques, applying some manual optimizations to the EPANET code, reusing the same, preloaded EPANET simulator instance and model for all optimizer evaluations and disabling all file input and output operations in EPANET.

A multi-threadable version of EPANET was developed to utilize the multiple cores available in current computers by running multiple simulations in parallel. The multi-threading approach taken here utilizes thread-local storage (TLS) variables available in modern C compilers instead of redesigning the EPANET API to be re-entrant, as was done in [149]. This work also uses simulation preemption [214], which saves computational time considerably.

Chapters 3.4.4, 3.4.3 and 3.11 describe the developed methods in more detail.

2.5.2 Modeling variable-speed pumping

Incorporating variable-speed pumping stations controlled for fixed head or flow has proven to be quite complicated. EPANET does not provide an easy way to use variable speed pumps, which causes problems in online modeling (e.g. [115]). EPANET simulator is also known to calculate wrong efficiencies for pumps at reduced speeds[244, 165, 99].

Wu et al. [296] used high elevation reservoir and flow control valve (FCV) to model the VSD pump and the author's [256] mention the use of a pump, FCV and pressure reducing valve (PRV) triplet. These methods failed to account for pumps' efficiencies directly at different rotational speeds. To solve the pump efficiency, the author's [252] used a pump and FCV so that the required pump speed reduction was calculated from the head loss over the FCV; however, the method had some numerical stability issues. The numerical stability issues caused by EPANET control devices are widely reported in literature (e.g. [243] and [78]).

Another methodology proposed in [125] and the author's [256] uses software Proportional Integral Derivative (PID) controller [32] integrated into EPANET for controlling the pumps. This, however, requires careful tuning of the controller parameters [180] and the use of very short simulation time steps, which increases the computational time considerably.

Todini et al. [261] modified the EPANET solver so that it can calculate pumps' relative speed when the pumps are controlled for a fixed head and [297] for a fixed flow. The methods presented in the papers, however, are patented under [263] and only available in WaterCAD and WaterGEMS simulation software by Bentley.

The other problem related to modeling the pumps is that EPANET does not enable calculation of the right efficiency values for variable speed driven pumps. For example, both in [165] and [99] modeled pump efficiency in EPANET is compared with experimental data. The papers show that EPANET is incapable of modeling the pump efficiency at lowered speeds. While [165] proposes the usage of affinity laws and assuming that best efficiency point (BEP) stays constant, [99] and [244] propose using the frequency scaling function proposed in [235] to provide more accurate estimates for the pump efficiency at lower rotational speeds.

An alternative way for accurate and efficient modeling of the flow or pressure controlled variable speed pumps and VSPs working in parallel was developed in this thesis research, as described in detail in Chapter 3.4.1. The developed method allows for both pressure and flow controlled pumping, also within one model, and allows changing the control mode dynamically. Pump battery is basically modeled either as FCV or PRV, but negative head losses are allowed. The pump performance and energy usage characteristics are modeled separately, as described in Chapter 3.3, solving the problems in modeling pump efficiencies correctly by utilizing the full pump drive energy calculation and optimization method developed in this thesis.

2.5.3 Online and operative modeling

One of the problems in EPANET regarding operational optimization is the inability to change the utilization pattern used for pump or control rules depending on multiple variables, such as time and tank level, programmatically [167, 150]. For example, Marchi et al. [167] developed an extension to EPANET toolkit to allow the control rules to be changed dynamically and López-Ibáñez [150] added a new variable, EN_UPATTERN, for `ENsetlinkvalue` function that allows the pattern to be changed.

Besides problems related to modeling variable-speed pumps, EPANET has convergence problems when modeling other control devices, such as flow control and pressure control valves. [243, 78]

Typically, hydraulic modeling is used offline as a tool for design problems, or as a part of a network design optimization process. The model can be, however, linked to the SCADA system for a real-time or online analysis of the system, and form one data source for an expert system.

Several publications have reported online modeling, for example, [125], [158], [77], [116], [61] and [255]. It has been used in various kinds of decision-making: most commonly in risk studies [187] and/or for fault detection [221], but also as a soft sensor and quality modeling, as in [77] and [255]. Risk studies have played an important role in water quality analysis, where any kind of intrusion or human error causes changes in water quality parameters that may cause a serious risk for human life [188]. Offline calculations are preferable due to the large amount of data analysis and calculations needed for any updates in the model. Online models need different problem descriptions to minimize the calculation time, or allocation of more processing resources either locally [296] or using cloud services [21, 199].

Online modeling is not important only because of water quality aspects, but using optimal control settings in the system at all times can save a lot of operational costs. [95, 195] Obviously, not all calculations can be done in real time. Therefore, in reality, offline calculations are combined with online calculations. Optimal pump scheduling in real time with or without near-optimal tank water levels has been studied in [177] and [31]. Offline calculations are common for some particular network components that do not change in time, for example, optimal pump working combinations that can be selected during an online calculation step so that the energy use will be optimal [254]. Any kind of real-time optimization needs also real-time measurements. Those measurements are usually received through SCADA and used in terms of the real-time control model [90, 277].

Real-time data usage poses some data quality and quantity problems, which have to be tackled.[35] For example, there can be missing or incorrect data, and time synchronizing problems, that the online model has to cope with.[287] Hatchett et al. [116] define real-time modeling as the integration of network hydraulic and quality model with operations data collected and stored via SCADA. They use open source hydraulic modeling packages EPANET[226] in conjunction with Real-Time Extension (RTX) module [115].

To further facilitate model online use Paluszczyszyn et al. [191] developed a methodology to skeletonize the hydraulic model real-time and Okeya et al. [181] used data assimilation to keep the model always up to date.

Online model provides a way to have better overall view of the current water supply system state, and to analyze the historical performance when the simulation results are stored in an appropriate format to facilitate analysis at a later time. An automatic anomaly detection can be performed by comparing the simulated and measured parameters. In particular, the online quality modeling can be a useful tool for improving the system performance and preventing quality problems. For example, water source tracing and water age that are both hard to measure can be readily simulated online and shown in SCADA to facilitate the decision-making process and the system analysis. [255]

While online modeling is understood well and the online simulation process in itself is simple, the problem with incorrect or missing data still remains. In the context of operational optimization, problems with measured data are most apparent in water forecasting calculations; they are discussed in the next section.

A new general library for accessing measurements in SCADA, laboratory information systems (LIMS) and other systems was developed in this research (see Chapter 3.6). The library is used for data access (fetching water tower levels, control system settings, flow measurements), calculating water balances and partly fixing the missing or incorrect data.

2.6 Demand forecasting

The most important aspect in operating a water supply system is to satisfy the consumer water demand. Accurate demand forecasts are required for strategic, operational and tactical decisions for water utilities. Short-term demand forecasting is a prerequisite for any optimal control system. [20, 140, 117]

There is no single established terminology for the forecast horizon used. [82] Here the focus is on the short-term, or tactical, water demand forecast defined as an hourly resolution for at least 24-hour forecast horizon.

Herrera et al. [117] list several benefits for accurate short-term demand forecasts available:

1. From an operative point of view, it enables water managers to determine optimal regulation and pumping schemes to supply the predicted demand. The aim is to improve the energetic efficiency through lower pumping energy consumption.

2. From the quality point of view, the more suitable combination of water sources to obtain a given standard in the supplied water may be selected.

3. From the vulnerability point of view, the comparison between the predicted and the real flow measurements can help pinpoint possible network failures (water leaks and pipe bursts). This provides the first step of a procedure for establishing an early warning management.

There has been considerable and continuing interest in developing methods to forecast the demand. Donkor et al. [82], Coelho and Andrade-Campos [65] and House-Peters and Chang [120] review the forecasting methodology and concepts in depth.

Commonly used methods include linear regression models and auto-regressive integrated moving average (ARIMA) models. More recently, various machine learning algorithms and Fourier analysis methods have been used.

Bakker et al. [28] argue, that the results are inconclusive in practical applications of ANN based methods compared to more traditional time series forecasting methods in short-term forecasting. According to Herrera et al. [117] the ANN and pattern models have not performed well, but Alvisi et al. [18] argue that pattern based methods work well, whereas ANN based do not, especially when the weekday changes. Earlier in Jain and Ormsbee [127], it was concluded that artificial intelligence (AI) methods perform better than the statistical methods.

The model accuracy and requirements, and thus the best forecast method, seem to depend on the externalities affecting the demand and explanatory variables chosen for the model. This might partly explain the partly contradictory and inconclusive results.

Water demand has strong daily and weekly patterns, and often exhibits clear yearly seasonality. The most commonly used explanatory variables in the literature are previous demand, especially at the same hour and same weekday, day of the year, and outdoor temperature and rainfall. The hotter the climate, the stronger the effect of temperature and rainfall on the demand. [317, 18]

Homwongs et al. [118] developed a method based on recursive least squares and Winters exponential smoothing algorithm. In [20], enhanced rough-set approach was used for automatic heuristic rule discovery based on observed data. The authors note that resulting if–then rules are easily understood by the users. Zhou et al. [317] developed a time-series based method including climatic correlation and auto-correlation for forecasting daily demands. In [107] pattern recognition is used.

Herrera et al. [117] present a comprehensive study on the accuracy of multiple different machine learning algorithms for short-term demand forecasting. The analyzed algorithms were artificial neural network (ANN), projection pursuit regression (PPR), multivariate adaptive regression splines (MARS), support vector regression (SVR), random forests, and finally, weighted pattern-based model as baseline. Weighted pattern based method and ANN performed the worst of all tested algorithms: root-mean-square error (RMSE) about 8 and 6, respectively, and SVR models performed the best, closely followed by MARS: RMSE about 4.5 in both cases.

Bai et al. [27] propose the variable-structure support vector regression (VS-SVR) method for demand forecasting, and Brentan et al. [47] use the hybrid SVR and adaptive Fourier series model for real-time demand forecasting.

Artificial neural network (ANN) based forecasting models have been widely used in demand forecasting. [128, 310] More recent developments in the field include the usage of dynamic neural network (DAN2) based approaches. The DAN2 algorithm was first introduced by Ghiassi et al. [102], and it has been successfully applied to water demand forecasting by several authors.

Traditional multi-layer perceptron with back-propagation ANN, DAN2 and two different hybrid models based on the ANNs and Fourier series method were compared in Odan et al. [178]. Hybrid model based on DAN2-H was found to be the most accurate.

Velásquez-Henao et al. [276] have improved the original DAN2 by using the ordinary least squares method (OLS), thus reducing the number of parameters and automatically estimating all the linear parameters.

An example of the neural–heuristic hybrid algorithm can be found in Yurdusev and Firat [308]. The adaptive neuro-fuzzy inference system was used for monthly demand forecasts. Alvisi et al. [18] used a two-level pattern based method.

Felfelani and Kerachian [93] examined modeling of water demand at significant changes in the population size over year. Their approach uses ANNs. Altunkaynak et al. [17] used time series and fuzzy logic for forecasting monthly demands for Istanbul. The method was compared with an auto-regressive model, and the proposed method performed favorably.

Recent research in [190] compares multiple short-term demand forecast models for the same two-year long datasets from seven different networks and districts. The compared models were the ANN model, the pattern based model, two moving time window methods, the probabilistic Markov Chain based model, and a naïve model using long-term hourly averages. The results show that different non-naïve models perform well and offer similar forecasting accuracy. However, the moving time window models perform best outside the calibration data set.

Much of the research focuses on longer time horizon forecasts and the effects of weather on the demand. AI models have gained more popularity lately, and they show promising performance. The problem with AI models is, however, that they require careful teaching, long input data time-series, and over-learning has to be avoided. Neither do they provide much insight into the reasons for varying demand.

The method used in this thesis research is described in Chapter 3.9. This work opts for a classic moving average model for the water forecasts. For each pressure zone or district metering area (DMA), three-month data is grouped by weekday and hour (noting national holidays), and a median, 10 and 90 % percentiles are calculated for each hour of each weekday. Previous 24-hour demand is calculated, divided by the typical demand at the same period. The typical usage, median, for the forecast horizon is then multiplied by the resulting factor. Missing and incorrect data is handled by limiting the hourly demand in each zone to 10 % to 90 % percentiles.

2.7 Water network optimization

The water network optimization problems can be roughly categorized into three different classes: calibration, design optimization and operational optimization. In terms of published literature, design optimization dominates the research.

All problems can be solved either offline or in near real-time. The different classes share much of the challenges, and the same optimization methods can mostly be applied in the different classes of optimization problems.

Calibration problems try to modify model parameters such that the error between some simulated and measured hydraulic parameters, such as pressures and flows, is minimized. Calibration is most commonly used to ensure that the model's hydraulic capacity matches the real system ([209], [275], [129]). Other uses include finding leakages ([302], [207]), locating closed valves ([285]) and calibrating quality parameters ([132], [233]).

Design problems relate to finding optimal pipe diameters, network structure, valve, pumping station and tank locations and sizes. The optimality is often defined as a minimum cost required to meet the constraints, but more recently, multi-objective optimization has become more and more common. Typical multiple objectives include, for example, cost and resilience [274] or cost and greenhouse gas emissions [138]. Design problems often include operational aspects, like optimizing pump scheduling problems [137, 138, 144].

As most of the network performance characteristics are decided during the design process and cannot be easily or economically changed later, the design optimization is very important. For example, significant cost reductions in water supply systems can be obtained by optimizing the storage tank volumes and levels in conjunction with the optimal control of pumping stations (see Table 3). [176]

Table 3. The estimated energy savings potential in water supply systems (adapted from [176])

Energy efficiency actions	Savings potential
The use of tanks for flow control and storage	10–20 %
Correct pump sizing	15–25 %
Real-time energy monitoring	5–20 %
The use of high efficiency motors	5–10 %
The use of variable-speed motor-pump sets	10–50 %
The operational optimization of pumping systems	15-30 %
Pump flow variation through VSDs instead of valve throttling	vary, >50 %

Operational optimization problems try to reduce costs of operating a WSS by changing some operational parameters, like pump and valve settings. In multi-objective cases, other parameters, such as greenhouse gas emissions or water quality, can be included, but multi-objective optimization is not commonly used – only 15 % of the operational optimization papers presented in the next section, Chapter 2.8, use multi-objective optimization.

One advantage of operational optimization over other measures is that it may be deployed without the large investments or changes to the network. In addition, the cost reductions from operational optimization are realized in a short term. [176]

Operational optimization literature is reviewed in more detail in Chapter 2.8. Some space is dedicated to network optimization too, as many of the problems and methods are shared between the different classes of network optimization problems. The network design problems often solve a pump scheduling problem besides the design, as the network design and operation are tightly interlinked. Multiple different optimization algorithms have been applied in design problems that would be readily applicable in other water network optimization problem classes too.

Yates et al. [305] prove that water system optimization problems are computationally NP-hard even for the simplest branched networks and even more so for complex looped systems. NP-hardness implies that only approximate methods exist for obtaining the optimum, and thus classical optimization methods do not work well.

Lansey [145] provides an analysis on the development of WSS optimization problems. The article recognizes three distinct phases in the development of WSS system optimization: linear and dynamic programming era from about 1968–1984, non-linear programming era from about 1986–1994 and stochastic era from 1994.

Alperovits and Shamir [16] developed a linear programming gradient method for design optimization; Lansey and Mays [146] have optimized network design for Anytown like WDN using dynamic programming; Gupta et al. [111] also use non-linear programming for design optimization, and Simpson et al. [245] introduced genetic algorithms (GA) to the WDN design optimization.

Some recent examples of linear and non-linear methods include [232] using optimal power use surface (OPUS) methodology and [230] using Mock Tree II algorithm in Hanoi, Balerma, R28 and Taichung networks, in which the algorithms quickly provided results similar to previously reported best designs. Price and Ostfeld [203] have used iterative linear programming for solving pump scheduling problems, and in [45] the classic network design problems are formulated as mixed integer non-linear integer programming (MINLP) problems and solved using a solver implemented in the basic open-source non-linear mixed integer programming (BONMIN) version 1.0 package.

Stochastic or meta-heuristic methods are efficient, both in terms of precision and computational effort, in solving many real-life optimization problems. Their definite benefit is that it is not required to formulate the problem in analytical form and the formulation can be non-differentiable.

Meta-heuristics fall in two categories: trajectory-based meta-heuristics and population-based meta-heuristics. The main difference is the number of proposed solutions used in each step of the (iterative) algorithm. [13]

A trajectory-based technique starts with a single initial solution and at each step of the search, the current solution is replaced by another solution found in its neighborhood. Trajectory-based meta-heuristic methods allow a locally optimal solution to be found quickly, therefore they are called exploitation-oriented methods. [13]

Population-based algorithms make use of a population of solutions. In this case, the initial population is randomly generated (or created with a greedy algorithm), and then enhanced through an iterative process. At each generation of the process, the population is replaced by newly generated individuals. These techniques are called exploration-oriented methods because their main ability depends on the diversification in the search space. [13]

Constraints are typically formulated as penalty costs, when meta-heuristic methods are used because they often do not support direct constraints. Disadvantage of penalty methods is that choosing penalty parameters is time-consuming and requires great care. In addition, penalty parameters are case-sensitive and do not necessarily steer the search toward the best solutions in every situation. [242] Afshar and Mariño [12] introduce a GA variant with self-adaptive penalty costs similar to [299]. Other self-adaptive fitness formulations can be found in [91] and [92]. Siew and Tanyimboh [242] present a penalty free approach for optimizing

WDNs by utilizing multi-objective optimization and pressure dependent simulation.

Examples of using stochastic methods in the WDN design include the following: genetic algorithm (GA) [245], harmony search (HS) [97], simulated annealing (SA) [70], ant colony optimization (ACO) [161], shuffled leaping frog algorithm (SLFA) [87], tabu search (TS) [71], particle swarm optimization (PSO) [173], memetic algorithm (MA) [23], hybrid discrete dynamically dimensioned search (HD-DDS) [265], genetic heritage evolution by stochastic transmission (GHEST) [40], honey bee mating optimization algorithm (HBMOA) [172], genetic expression programming (GEP) [288], differential evolution (DE) [312], and state transition algorithm (STA) [318].

The most commonly used meta-heuristic method in the water sector is definitely GA with its multiple variants. Some examples are presented in [112], [272], [42], [240], [189], [296], and [31]. The genetic algorithm simulates natural evolution: the algorithm begins with a randomly generated population of solutions, and after each iteration, the best solutions are most likely to survive into the next iteration (generation). The surviving solutions exchange design variable values (genes) with each other, and there is a chance for mutations (random changes in design variable values).

Each article typically compares the resulting costs and required computational time or number of iterations with some previous algorithms on the benchmark networks. [288] In [72], GA, PSO and DE in New York tunnels and Hanoi problems are compared. Artina et al. [22] compare BONMIN algorithm with NSGA-II and GHEST for optimal design in the Modena water distribution system.

The literature, especially in the design optimization, deals merely with single speed, on-off controlled pumps. One reason is that the typical benchmark networks are gravity-fed, and the difficulties in modeling VSD pumps in EPANET certainly play a part (see Chapter 2.5). [294] One of the few papers using variable-speed pumping, [296], approximates the pump energy usage employing a combination of a high elevation reservoir and a flow control valve in the EPANET model.

There seems to be a trend to introduce more methods that combine aspects of both stochastic methods and classical optimization. Some examples include [219] combining LP and GA for longer term operational optimization of a multi-reservoir system, [206] combines LP with hybrid discrete dynamically dimensioned search (HD-DDS), Geem [96] combines particle-swarm concept with harmony search, and Giacomello et al. [103] apply LP together with a greedy algorithm for pump scheduling problems in Anytown and Richmond networks.

Linear programming was combined with differential evolution in Zheng et al. [314]. Network is first partitioned into trees. Binary linear programming is used for optimizing the trees and finally, DE is used for optimizing the core of the network containing loops.

Multiple stochastic methods or the stochastic method and machine learning can also be combined together in order to utilize each algorithm's strengths. For example, PSO and GA were combined in [25] for design optimization, Raad et al. [210] have utilized a multi-algorithm, genetically adaptive multi-objective (AMALGAM) algorithm that uses multiple meta-heuristic algorithms simultaneously, and Dipierro et al. [81] analyse the performance of the extended version of hybrid evolutionary algorithms of efficient global optimization (ParEGO) [141] and the multi-objective evolution model (LEMMO) [133], combining a level of machine learning with evolutionary algorithms.

The use of surrogate models is a very popular way to speed up the optimizations (see 2.5). The most typical surrogate models are ANNs [49]. One example of advanced surrogate model usage is online retraining of ANN during the design optimization process in [34].

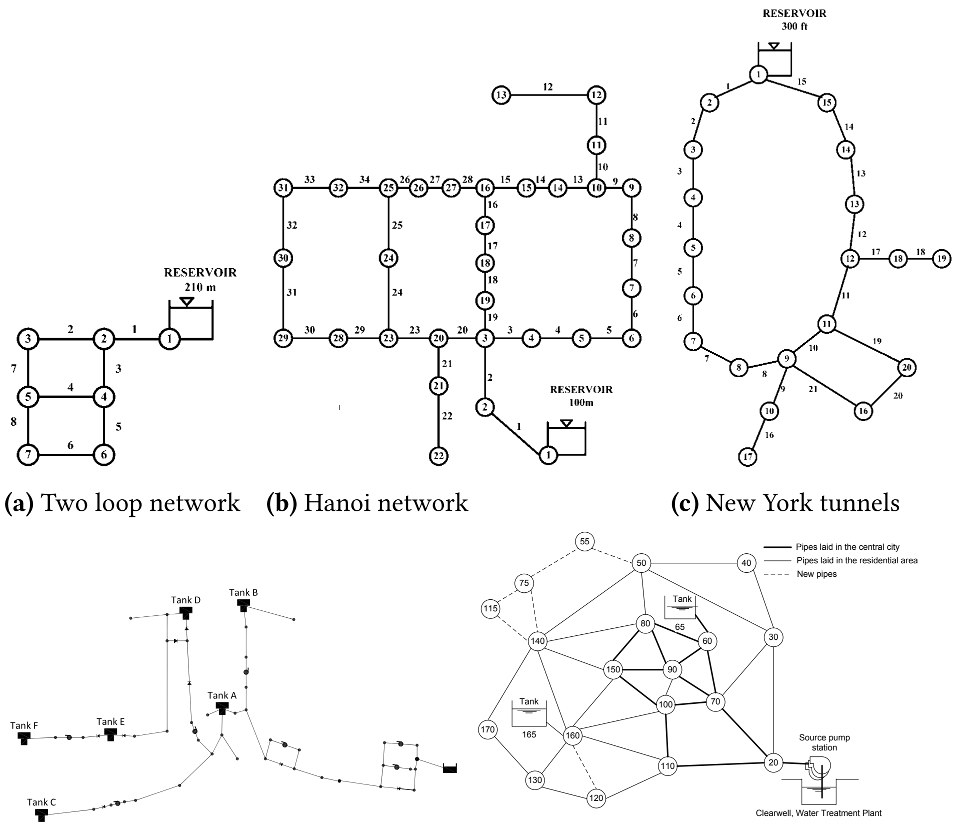
Graph decomposing can be applied in optimization algorithms in order to reduce the problem search space size and to divide the simulations into multiple much smaller units. One example of the graph decomposition approach to the network design problem using DE can be found in [315].

Many of the stochastic algorithms can be parallelized in order to reduce the computational time by utilizing multiple CPU cores or cloud computing services now commonly available. Trajectory-based meta-heuristics can be parallelized in three ways: the parallel exploration and evaluation of the neighborhood (parallel moves model), the parallel multi-start model, and the parallel evaluation of a single solution (move acceleration model). Two parallelizing strategies are common for population-based algorithms: parallelization of computations, i.e. each individual is evaluated in parallel, and parallelization of population, i.e. is population is split into different parts that can be exchanged or evolved separately, and then joined later. [13]

There is an ongoing research to develop completely new meta-heuristic algorithms that can better utilize multiple CPU cores and GPUs. [13] Even though the optimization algorithm itself is not parallelized, the objective function evaluations can benefit from parallel processing (e.g. [300] and [30]).

Other new hydraulic simulation developments, such as graph decomposition and quantised state-models, hold a lot of promise to speed up the simulations and optimization processes (see Chapter 2.5).

The most common benchmark networks used for evaluating the optimization method performance are Two loop, New York tunnels, Hanoi, Richmond and Anytown networks shown in Figure 9. Balerma irrigation network[217] has also been used as benchmark in several studies.



(a) Two loop network **(b)** Hanoi network **(c)** New York tunnels

(d) Richmond **(e)** Anytown

Figure 9. The traditional test networks for optimization algorithms[148], [272] apud [103], [286] apud [198]

Use of the current benchmark models (see e.g. [89], [137], [72]) has been criticized, especially for operational optimization because for the most part, they are gravity-fed and the networks do not represent actual large-scale network models too well. Many of the methods in literature, however, have been tested on these test networks in order to make it easier to compare the results.

Different algorithms and problem formulations have been shown to be efficient in reducing network investment and operational costs. New developments reduce computational time, use more accurate methods and often give better solutions than earlier methods. Currently, various meta-heuristic methods are the state-of-the-art solution for network optimization problems. New meta-heuristic and hybrid algorithms are constantly developed and succesfully applied in water network optimization.

2.8 Operational optimization

Operational optimization is directed to the optimal operation of the water supply system usually in terms of energy cost by finding out optimal time-varying settings for the various controllable devices, such as pumps and valves, in the water supply system, while ensuring sufficient quality of service.

The optimal operation of distribution systems with multiple water storage reservoirs and multiple sources is a large-scale nonlinear optimization problem with continuous and discrete variables, which makes the problem difficult to solve mathematically. [176]

Cherchi et al. [62] review the operational aspects of the water supply system management, focusing on the energy and quality aspects. The article reports operational costs savings of 8 % to 15 % and energy savings of 6 % to 9 %. An earlier review by Coelho and Andrade-Campos [65] focuses more comprehensively on all aspects of water supply systems and their energy optimization, starting from the system design. The review reports operational optimization cost savings of 6 % to 26 %.

Historical research is reviewed in Ormsbee and Lansey [185], and Lansey [145] reviews the evolution of WSS optimization in quite broad perspective. According to the paper, however, energy optimization was intensively studied 1988–1997, and the field was already complete by 2006, except that potential in reducing computation times existed. Research published afterwards suggests that many more questions had to be solved than it was concluded in [145].

Operational optimization can be performed offline or online. Offline operational optimization can generate optimal operational rules for different scenarios [272, 43, 278, 203, 167], from which the system operators can choose the solution to apply for the actual situation at hand.

Online optimization, on the other hand, predicts the future water demands and uses the current system state as initial conditions to find out the optimal way to operate a system in the short-term future, typically for the next 24 hours [169, 213, 238, 122, 179].

The online optimization requires a good automatic control system, in which tank levels, flow measurements, pump operations, and a decision-system tool are all linked together [167]. Some additional issues related to hydraulic simulation and data quality have to be tackled as well, as stated earlier in Chapter 2.5.

The operational optimization of WSSs can be performed through four steps, including (1) establishing the definition of the optimization problem, (2) carrying out the computational modeling of the system, (3) calibrating and validating the hydraulic model, and (4) performing the simulation and optimization procedures.

[176] The following subsections review the various constituent parts of the operative optimization.

Table 5 lists most of the published research related to the operational optimization of water supply systems starting from the 1990s. The parameters listed in the table are the algorithm used, whether it is reported that the method supports variable speed pumps (VSP), objectives, constraints and design variables (please refer to Table 4 for explanations of the values), number of and type of the objects for which optimal settings are sought, size of the hydraulic model used in the optimization, the possible surrogate model used instead of a full hydraulic simulation, time of optimization run and the best result reported in the article.

Next sections discuss the various aspects of the operational optimization in more detail: objective function, decision variables, optimization algorithms, real-time considerations, and finally, real-time operation optimization frameworks.

Table 4. Short-hands used for objectives, constraints and design variables in Table 5

Letter	Objective	Constraint	Design variable
A	Energy cost	Min pressure	Pump status
B	Production cost	Max pressure	Tank trigger levels
C	Chlorine cost/conc.	Tank level same in the end	Pump/valve setting
D	Leaks	Min tank level/volume	Valve status
E	Pump switches	Max tank level/volume	Chlorine content
F	Maintenance cost	Source and/or pump limits	Time triggers
G	Peak power	Number of warnings or errors	Station flow
H	Tank level variation	Tank capacity	
I	Min pump stop time	Pump switches	
J	Hydraulic reliability	Pipe flow/velocity limit	
K	Quality		

Table 5. List of articles on operational optimization of water supply systems (see Table 4 for key)

Authors	Year	Algorithm	VSP	Objective	Constraints	Design	Controls	Model size	Surrogate	Time	Result
Fallside et al.[88]	1975	DP	yes	A	A+B+D+E	G	10 stations	6/-	MB	90 s	5-10 %
Zessler & Shamir[309]	1989	DP (PO)	no	A	C+D+E+F	G	7 stations		REGR	15 min	-
Brion & Mays[48]	1991	NLP (GRG2)	no	A	A+B+C+D+E	A	1 station (3 pumps)	-/126	-	-	5.2 %
Jowitt & Germanopoulos[134]	1992	LP	no	A	C+D+E+F	A	22 pumps	87/122	-	41 s	15.6 %
Yu et al.[307]	1994	NLP (GRG)	no	A	A+B+D+E+F	G	4 stations	50/75	-	4.7 min	8.3 %
Mäckle et al.[160]	1995	GA	no	A	C+D	A	4 pumps	1/4	MB	20 min	20500 $
Ormsbee & Reddy[186]	1995	Heuristic search	no	A	A+B+C+D+E+J	G	2 stations (6 pumps)	33/50	SKEL	40 min	6.9 %
Nitivattananon et al.[175]	1996	DP	no	A	C+D+E	A	10 stations (39 pumps)	-	MB	15 min	20 %
Savic et al.[236]	1997	MOGA + local search	no	A+E	C+D	A	4 pumps	1/4	MB	-	20800 $
Ulanicki et al.[268]	1999	DP + NLP	no	A+B	A+B+D+E+F	A	5 stations and 2 valves	82/84	MB	50 s	7 %
Sakarya & Mays[229]	2000	NLP (GRG2)	no	A+K	A+B+D+E	A	1 pump	19/27			
Skarya & Mays[229]	2000	GRG2 (NLP)	no	A+C	A+B+D+E	A	1 pump	19/27	-		28 %
Biscos et al.[36]	2003	MINLP	no	A+C	D+E	A+D	1 pump and 4 valves	-/10	MB	-	-
McCormick & Powell[171]	2004	2-level SA	no	A(+B)	D+E+F	A	20 stations (35 pumps)	-	LIN	15 min	-
van Zyl et al.[272]	2004	GA + hillclimber	no	A	C+D+E+I	B	3 pumps	48/51	SKEL	21 h	26 %
von Lücken et al.[282]	2004	different GAs	no	A+E+F+G+H	D+E	A	5 pumps	3/6	MB	-	-
Barán et al.[29]	2005	different GAs	no	A+F+G+H	D+E	A	5 pumps	3/6	MB	-	-
Farmani et al.[90]	2005	different EAs	no	A+B	A+C+D+E	B+D	13 pumps and 7 valves	-	-	-	none
Gogos et al.[106]	2005	GA	no	A+G	D+E	A	-	-	MB	-	28 %

Continued on next page

Table 5 – continued from previous page

Authors	Year	Algorithm	VSP	Objective	Constraints	Design	Controls	Model size	Surrogate	Time	Result
López-Ibáñez et al.[153]	2005	SPEA2	no	A+I	-	F	3 pumps	-	MB	-	-
Bounds et al.[44]	2006	NLP + MINLP	yes	A+B	A+B+D+E+F	A+C	35 stations and 63 valves	252/530	SKEL	65 min	14 %
Martínez et al.[169]	2007	GA	no	A+B	A+B+C+D+E+F	A	17 pumps + 10 valves	725/772	ANN	10 min	17.6 %
Rao & Salomons[213]	2007	GA	no	A+B	A+B+D+E+F	A	3 pumps	21/41	ANN	-	-
Salomons et al.[234]	2007	GA	no	A	A+C+D+E+F	A	13 pumps + 1 valve	112/126	ANN	-	25 %
López-Ibáñez et al.[151]	2008	ACO	no	A+E	A+C+D+E+G	A	7 pumps	48/51	SKEL	2.5 h	92 £
Shamir & Salomons[240]	2008	GA	no	A	A+C+D+E+F	A	17 pumps + 1 valve	77/92	SKEL	40 s	10 %
López-Ibáñes et al.[149]	2009	ACO	no	A+E	A+C+D+E+G	A	7 pumps	48/51	SKEL	30 min	92 £
Pasha & Lansey[194]	2009	LP	no	A	D+E+J	A	4 pumps	21/37	-	-	-
Wu & Zhu[298]	2009	fmGA	no	A	A+B+D+E	A	2 pumps	91/115	-	100 s	-
Broad et al. [50]	2010	GA	no	A+C	A	B	11 pumps and valves	1271/1376	SKEL + ANN	1.4 h	14 %
Georgescu et al.[100]	2010	HBMOA	no	A	C+D+E	B	6 pumps	13/20	-	-	-
Skworcow et al.[247]	2010	NLP (CONOPT)	yes	A+B	A+B+C+D+E+F	A+C	8 pumps + 1 PRV	43/45	SKEL	2 min	34 %
AbdelMeguid & Ulanicki[10]	2011	GA	yes	A	C+D+E	B+C	11 stations		-		1916 £
Pasha & Lansey[195]	2011	SFLA	no	A	D+E	A	4 pumps	21/41	SVM	-	17124 kWh
Price & Ostfeld[202]	2011	LP (CBC)	no	A	D+E	A	1 pump	6/5	-	-	8.9 %
Selek et al.[238]	2012	micro-GA	yes	A	C+D+E+F	A+C	11 pumps	20/19	MB	-	-
Bagirov et al.[26]	2013	Hooke-Jeeves	no	A	A+B+D+E	A	2 pumps ; 2 pumps	92/122 ; 13/15	-	-	-
Giacomello et al.[103]	2013	LP + local search	no	A	A+C+D+E	A	7 pumps	48/51	SKEL	23 s	118.58 £
Kurek & Ostfeld[144]	2013	SPEA2	yes	A+C	G+H+C	C+E	2 pumps	92/122	-	4 h	-

Continued on next page

Table 5 – continued from previous page

Authors	Year	Algorithm	VSP	Objective	Constraints	Design	Controls	Model size	Surrogate	Time	Result
Zhuan & Xia[319]	2013	DP (RDPA)	no	A(+E)	D+E	A	3 pumps	-	MB	0.17 s	-
Ibarra & Arnal[122]	2014	Branch & Bound	no	A	D+E	B	-	-	MB	610 s	20 %
Jung et al.[136]	2014	GA	no	A	A+C+D+E+I	A	3 pumps	7/8	SKEL	160 s	19-27 %
Pasha & Lansey[196]	2014	LP+SFLA	no	A	D+E	A	4 pumps	21/41	SVM	5 min	18118 kWh
Price & Ostfeld[203]	2014	LP (CLP)	no	A	D+E	A	9 pumps	-/110	-	41 min	-
Puleo et al.[206]	2014	LP / LP+HD-DDS	no	A	D+E+F	A	4 pumps	21/37	-	-	-
Puleo et al.[205]	2014	LP	no	A+B+K	D+E+F	A	2 stations (8 pumps)	11/12	SKEL	-	-
Skworcow et al.[246]	2014	NLP	yes	A+B	A+B+C+D+E	C	13 pumps and 42 valves	175/391	SKEL	5 min	-
Babei et al.[24]	2015	ACO	yes	A+C+J	A+B+D+E	C+E	3 pumps (1 station)	25/45	-	-	-
Boano et al.[37]	2015	GA	no	A	A+C+D+E	A	9 pumps	3000/3000	-	-	740kWh/d
Bohorquez et al.[39]	2015	GA	yes	A+D	A+B+C	B	3 pumps	-/159	-	-	33 % (VSP)
Ghaddar et al.[101]	2015	DP	no	A	A+E	A	7 pumps	47/51	-	1.4 h	34 %
Odan et al.[179]	2015	AMALGAM	no	A+J	A+C+G+I	A	3 pumps	20/20	SKEL	16 min	13.2 %
Zheng & Huang[316]	2015	DP (IDPA)	no	A	D+E+F	A	2 pumps	-	-	0.572 s	60 %
Costa et al.[67]	2016	Branch & Bound	no	A	A+B+C+D+E+I	A	3 pumps	25/45	-	81 h	-
Marchi et al.[167]	2016	GA	no	A		B	3 pumps	16/18	-	-	9.23 %
Price & Ostfeld[204]	2016	Shortest path	no	A+K	A+D+E	A	11 pumps	388/444	-	28 s	-
Shi & Fengqi[241]	2016	NLP (OAGO-N)	no	A	D+E+F	A	5 pumps	25/26	MB	322 s	-
Bonvin et al.[41]	2017	various MIQCP	no	A	D+E+F	A	2 stations (6 pumps)	65/59	-	60 s	16.9 %
Makaremi et al.[163]	2017	NSGA-II	no	A+E	A+C+D+E+F	A	6 pumps	34/40	-	-	none

2.8.1 Objective function

Most commmpnly, the objective function includes only the cost of network pumping electricity while raw water extraction and water treatment pumping costs are left out, even though a major part of the electricity is consumed in these parts of the supply system [237]. Only 15 % of the literature cited in Table 5 uses explicit multi-objective optimization, though 45 % of the articles include multiple variables in the single objective function, and could be regarded as scalarized multi-objective problems.

Motor and variable speed drive efficiencies are not typically accounted for, except when motor efficiency is included in the pump's efficiency curve. The method is valid when the pump runs at its nominal speed, but otherwise it gives wrong results. [254, 165]

Water treatment costs are only rarely included in the objective function. Few articles, such as Farmani et al. [90], [44], [169], [247], [50] and [205], include any production costs. Some others, like [134] mention that including water production costs is straightforward.

Multiple studies have addressed multi-objective optimization. Savic et al. [236] have minimized energy and maintenance costs, [153] has optimized pumping costs and average minimum pump stop time, [29] has used electricity cost, number of pump switches, reservoir level variation and maximum peak power, [289] and [296] have optimized cost and environmental effects, [203] has optimized leakage and costs, and [24] has optimized electricity and chlorine costs, and hydraulic and quality reliability.

Production costs are not often included in the objective, but if the system has multiple sources with varying production costs, excluding them can result in solutions that are not optimal in terms of the total costs, as the production cost can be higher than the energy cost.

Pump maintenance costs are difficult to quantify, and often the number of pump startups is used as a surrogate (e.g. [236], [282] and [151]), but the cost of pump maintenance or replacements varies by the pump, and even the need for maintenance does not necessarily correlate directly with the number of pump startups. This thesis research limits the number of pump setting switches implicitly by the design variable formulation, as shown in the next chapter.

In this thesis the objective function is defined to include variable water production costs: energy and chemicals needed, and energy costs: every pump in the system be it part of the water treatment process or pressure booster station in the network. The average production costs are aggregated into source specific unit costs €/m^3. Water supply reliability is taken into account by using constraints formulated as penalty functions. More detailed description of the development of the objective function can be found in Chapter 3.7. The way water treatment processes and their energy consumption are modeled applying control system modeling [252, 256] is described in Chapter 3.5.

2.8.2 Decision variables

Decision variables of the pump scheduling problem can be formulated either explicitly as pump settings or implicitly using surrogate variables like tank trigger levels or pump station discharges. [185] Combination of both can also be used [167].

By far the most common approach is to explicit formulation use binary string for each single speed pump. Each bit in the string represents the pump status, on or off, at that time interval. This approach has been demonstrated, for example, in [160], [236], [170], [272], [151], and [218].

Tank triggers have been used, for example, in [272], [100] and [50]. When the operational rules are optimized offline, implicit formulation in terms of tank trigger level is regarded as more robust and it works better under uncertain water demands than explicit formulation, but generally explicit pump schedules tend to result in greater savings and utilize off-peak price tariffs better [10, 167].

Other decision variable formulations exist too. Different formulations can restrict the search space or allow for more flexible description of the problem. López-Ibáñez et al. [152] propose variable encoding based on the pumping period length with a fixed number of pump switches; similar formulations have been used elsewhere too, for example, in [26] and [179].

VSPs require some more work, and they are considered only in a subset of publications on pump scheduling problems. However, variable speed driven pumps are already quite common in water utilities, and they offer major energy savings and better controllability than single speed pumps (SSP) [164]. Using variable speed control tends to lower the pump maintenance costs [121].

Some examples of methods where VSPs were considered, include [88], [10], [296], [166], [114], [144], [39], and [24].

The problem with most VSP approaches present in the literature is that motor and VSD efficiencies at lower speeds are typically not considered, even though they have a major effect on the total efficiency [254, 255, 237], and frequency scaling is not taken into consideration [244]. Neglecting the effects, the accuracy of the published results considerably if no extra measures are taken to ensure that the pumps work close to the nominal speed (e.g. [68]).

While few articles, like [166] and [39], explicitly state how the energy usage of the pumps is calculated, it can be reasonable to assume that energy consumption values calculated by EPANET are used. EPANET, however, has major flaws in calculating VSP efficiency at reduced rotational speeds [166, 165]. Some of the published VSP optimization results can thus be inaccurate.

Variable speed pumps can be modeled and formulated in different ways, and this has a major impact on the problem complexity and computational time. Solving explicitly speed settings for every pump increases the search-space considerably; thus, methods that solve for the whole pump station's flow setting and then calculate single pump speeds based on the result can be much more efficient in terms of computational time. [309, 186]

Wu et al. [296] include flow controlled VSD pumps into a genetic algorithm driven system design and operational optimization problem by replacing pumping stations with high-elevation reservoirs and EPANET's flow control valves (FCV), and calculating the pump energy usage by the real inlet reservoir head and down-stream head of FCV.

Hashemi et al. [114] introduced a proper VSD controlled pumping station optimization with ant-colony optimization (ACO). The pumping station is replaced with a reservoir, the head of which is found through the optimization process. Resulting head and flow is divided between the pumps using ordinary, naïve variable speed control, and energy usage is calculated separately, based on the head and flow solved by the hydraulic simulator. A similar two-level method was also used in [24]. Analogical methods to reduce the number of decision variables have been introduced earlier for single-speed pumps in the 1990s (e.g. [309], [186] and [175]).

Several papers, e.g. [10], [166] and [144], use the optimization algorithm to directly solve the VSP speed and model the VSPs using ordinary EPANET pumps with relative speed settings.

This thesis research uses the purposely developed pump battery component in the EPANET model [256] presented in more detail in Chapter 3.4.1 to model pump stations. The parallel pump pre-optimization ([254], Chapter 3.3) is then used for solving each pump's speed and the total energy usage. In this way, the VSD controlled pumps in a pumping station can be efficiently modeled, as only the setting for the whole station has to be found, and the optimal way of producing the working point inside the station is ensured.

The decision variables for each station are the 24-hout pattern index and four flow or pressure settings that are used during different time periods, as described in more detail in Chapter 3.7.1. The method develops on ideas from [186], [114] and [152], and manages to reduce the number of decision variables from 24 per pump to five per station, while retaining much of the properties of the explicit pump schedule formulation. Proposed formulation reduces search-space to a fraction of the typical.

2.8.3 Optimization algorithms in operational optimization

Pump scheduling problems can be solved using a variety of optimization algorithms. Many of the algorithms used for network design optimization or calibration can also be used for operational optimization, as the problems are quite similar (see Chapter 2.7). Currently, meta-heuristic methods are typically used, because they are well suited for the problem type. The eras of various techniques are the same as in the network design optimization, and since about 1994 stochastic methods have dominated the field. [145]

Early literature on pump scheduling includes Fallside et al. [88] who used dynamic programming, and Jowitt and Germanopoulos [134] who used linear programming. Sun et al. [251] have developed the EMNET algorithm that solves LP with network structures very efficiently. The method was applied for operational optimization of a Southern Californian water utility. Ormsbee and Reddy [186] used non-linear heuristic combined with a KYPIPE hydraulic simulator and per pumping station settings for optimizing pumping schedules for Washington D.C. Nitivattananon et al. [175] used dynamic programming for Pittsburgh's water supply system operational optimization, and Zessler and Shamir [309] used the progressive optimality method, which is an iterative dynamic programming method. The method was tested on an unnamed regional water supply system. Mäckle et al. [160] was the first to use GA for pump scheduling optimization. The system examined consisted of a reservoir and a tank connected by four pumps.

Dynamic programming solutions are usually case-specific and cannot be readily applied to other systems. [186] Dynamic programming solutions also suffer more from the curse of dimensionality and are limited to smaller problems than other methods. [48]

Linear programming solutions are often case specific too. Discretizing continuous results and inaccuracies due to linearization cause difficulties. [194]

An example of more recent LP solution for pump scheduling is that of Pasha and Lansey [194]. They applied LP for optimizing pumping costs in Anytown-like network. Their proposed method, however, only works for a single source and single tank system. A series of papers [202], [203] and [201] present further developments and use linearization to solve pump scheduling problems.

A more recent example of dynamic programming can be found in [101], and non-linear programming in [44], which uses the generalized reduced gradient algorithm (GRG) and simple branch and bound (SBB) to optimize pump station flows, and Sakarya and Mays [229], which uses GRG2 and three different objective functions: pumping time, total cost and chemical concentrations.

Skworcow et al. [246] optimizes pump and valve schedules using CONOPT [84] NLP algorithm found in the general algebraic modeling system (GAMS) and skeletonized EPANET model [192]. The methodology allows for variable-speed driven pumps. The optimization method is fast, it took about five minutes to optimize for 24 hours in one hour intervals, but no energy savings data were reported. A similar method has been implemented in [36], but instead of using EPANET model for objective function evaluation, a simple linear mass balance model of WDS is used.

Bagirov et al. [26] encode pump start and run times as continuous variables and pump status at the start of the first period as a binary variable. In addition, the number of pump switching periods is limited to five. These reduce the search space remarkably. The problem is then transformed into a MINLP problem, and it is solved with grid-search and Hooke–Jeeves [119] search.

Savic et al. [236] applied multi-objective GA (MOGA) for optimizing cost and number of pump switches. McCormick and Powell [171] investigated the use of two-level simulated annealing. The total costs included energy cost, pump switching and maximum demand charges. An initial solution was produced by a descent method, then two-stage simulated annealing optimized the final schedule.

Kurek and Ostfeld [144] use the strength pareto evolutionary algorithm (SPEA2) to perform multi-objective optimization on both water quality and energy use. The design variables were relative pumps speeds for VSPs and chlorine concentrations at water sources, and tank diameters. The methodology was tested on one of EPANET's example networks.

Barán et al. [29] optimized pump schedules for four different parameters electric energy cost, maintenance cost, maximum power peak, and level variation in a reservoir using size different multi-objective evolutionary algorithms. The same case and algorithms were examined earlier in von Lücken et al. [282], which also used both parallel and sequential versions of the algorithms.

López-Ibáñez et al. [153] used the SPEA2 algorithm in the pump scheduling problem for minimizing energy and maintenance costs. Constraints were handled using a methodology based on the dominance relation rather than using penalty functions.

Gogos et al. [106] applied GA for optimizing pump schedules in 30 min intervals. Their method included an algorithm to repair infeasible solution chromosomes by adding or removing pumps so that reservoirs will not overflow or empty too much. The reported savings were 28 %, but few details of the system are provided. The paper does not explain how the pump power use is calculated exactly, but it seems that no hydraulic simulations are performed, instead a tabulated pump energy consumption values and a mass balance model are used.

AbdelMeguid and Ulanicki [10] solve the optimal pump scheduling problem in implicit form for a large real network consisting of both SSPs and VSPs using GA. Optimal tank trigger levels and relative rotational speeds for VSPs were generated for each pumping station for each tariff level. The optimization was done offline and the resulting tank trigger levels were then incorporated into PLCs controlling the pumping stations. A sensitivity analysis showed the methodology to be robust.

Marchi et al. [166] solve the optimal pump scheduling problem for both SSP and VSP using GA. Only two very small systems were studied. However, the study shows that VSPs improve the energy usage considerably over SSPs and that EPANET has major flaws in the way energy use is calculated for VSPs.

Bohórquez et al. [39] use GA for optimizing tank trigger levels for single speed pumps and speeds at different tank levels for VSPs. Only EPANET rule-based controls were used and the rotational speed was controlled directly. The method used included not only pumping costs, but also leakage costs in the objective function.

Some other algorithms used include neutral evolutionary search [238], honey bee mating optimization algorithm (HBMOA) [100], which optimized a set of tank trigger levels for pumps in the system, and ant colony optimization (ACO) was applied for explicit pump scheduling in [151] in van Zyl and Richmond networks.

Hybrid optimization algorithms combining multiple different optimization algorithms have been applied to operational optimization problems too. Some examples include those in [103].

Dynamically dimensioned search (DDS) algorithm was first presented in [264]. The algorithm is greedy, constant-time general purpose optimization algorithm, that first performs global search and as the number of objective function evaluations gets closer to the allowed number, the algorithm changes dynamically into more and more local search. The algorithm thus exhibits hybrid properties. The algorithm was later extended to support discrete variables and named hybrid discrete DDS (HD-DDS) [265].

Tolson et al. [265] compare DDS performance for optimizing WDN using classing New York tunnels problem, its double pipe version and Hanoi network, with different other algorithms such as GA, CE and PSO. The algorithm required less computational time and gave as good or better results than the other algorithms tested.

DDS performed well in relation to GA and various surrogate modeling approaches in [216] when several test functions were used, but [313] and [312] argue that DDS's performance in terms of speed and result quality is not in the best class in the configuration of the water supply network system. Puleo et al. [206] argue, on the other hand, that the principal advantage of DDS class of algorithms, compared with genetic and ant colony algorithms, is their good ability to find near globally optimal solutions while being significantly more computationally efficient.

Giacomello et al. [103] apply LP together with a greedy algorithm for pump scheduling problems in Anytown and Richmond networks. Puleo et al. [206] have used multi-stage LP for pump scheduling in Anytown. The results from LP were further optimized by HD-DDS[265]. Both studies show promising results by doing a rough approximate optimization with LP and then refining the results further with stochastic methods resulting in smaller computational time requirements.

Van Zyl et al. [272] use the hybrid algorithms the other way: first, optimizing more globally using GA and then refining the results using Hooke–Jeeves or Fibonacci hill climbing algorithms. The methods were tested using Richmond network.

Skworcow et al. [247] optimize pumping and water treatment costs for Yorkshire Water Services. The system includes both fixed and variable speed pumps and pressure reducing valves. The problem was solved using skeletonized [267] EPANET model including leaks modelled as emitters. CONOPT non-linear programming solver found in GAMS package was used as a solver. The continuous schedules solved by CONOPT were transformed into discrete schedules using an algorithm developed in Matlab. The reported savings were almost 34 %. The article explicitly includes the price of water treatment as fixed per-station unit cost. The use of fixed price, however, fails to capture the effect of varying energy losses in the treatment process due to friction, and pump and pump drive train efficiencies.

Nitivattananon et al. [175] decomposed the problem temporarily into short and long term sub-problems and spatially into several subsystems. Dynamic programming was then applied to optimize the pump schedules real time. Heuristics were used to rearrange the pump schedules in order to minimize the number of pump switches. The optimal discharges were calculated for each station, and the short-term optimization derived the single pump schedules. The only constraints are tank levels, and only rough approximation of the flow dependent pressure losses between tanks and pumping stations were considered. The method was applied in Pittsburgh and it showed 20 % reduction in pump energy costs.

Broad et al. [50] have optimized pumping and chlorine costs using the GA and ANN surrogate model. Optimization time of 1.4 h yielded operational cost savings of 21 % for the network of Wallan, Victoria, Australia.

Marchi et al. [167] extend EPANET rules engine in order to allow more complex rules, taking into account simultaneously several conditions (e.g., the time of the day and the tank level), to be generated and changed online. The extended EPANET is used with GA to optimize the operational rules based on both tank trigger levels and time of the day. The resulting solutions were cheaper than previously found simple rule based solutions.

Ostojin et al. [189] used GA optimized fuzzy logic control for real-time pump schedule optimization in a sewer pumping station resulting in 5 % energy cost savings. Zhuan and Xia [319] used reduced dynamic programming for operational pump schedule optimization. Farmani et al. [90] used GA for offline whole-cost optimization based on optimal pump-scheduling.

Babaei et al. [24] used multi-objective ACO to optimize pumping and chlorine costs, having explicit pump schedules and chlorine doses as design variables. The other objective was one of different reliability measures: hydraulic reliability, quality reliability or integrated reliability. For VSPs, an alternative reservoir method presented in [114] was used. The method was applied in the Anytown network.

Zheng and Huang [316] developed and applied a novel improved dynamic programming algorithm (IDPA) in the operational optimization of two-stage deep well pumping (single-speed). The method was compared with the traditional branch and bound (B&B) method. IDPA was almost two orders of magnitudes faster than B&B method, and the resulted costs were smaller.

Price and Ostfeld [200] and Price and Ostfeld [204] developed a novel way to solve pump scheduling problems by presenting the problem as a graph and using the shortest path algorithm to find the optimal pump schedule.

Based on the review, it seems apparent that more traditional optimization methods, such as LP and DP, have superior computational performance compared to meta-heuristics. However, meta-heuristics are much simpler to apply, and the literature shows that performance of different meta-heuristic methods, both in terms of the solution quality and computational resources required, can be reasonable.

While GA is most commonly used, many other methods can perform better, especially in terms of the number of solution evaluations. It was decided to explore the possibilities offered by the dynamically dimensioned search (DDS) in operational optimization, where it has not been applied before. DDS is very easy to implement, seems to converge on acceptable solutions quickly and it exhibits both global and local search properties like many hybrid methods, which have proven to be effective in this class of problems.

Based on the initial performance assessment, the DDS algorithm was slightly modified in this work. The performance was much better if a certain degree of non-greediness was allowed. The optimization algorithm is presented in Chapter 3.8.

2.8.4 Real-time considerations

The use of meta-heuristic optimization approach requires a great number of hydraulic simulations. Even though a single simulation can be reasonably fast, the simulations still are the bottleneck of the optimization process. Much of the research focuses on finding the best optimization methods in terms of the number of hydralic simulations performed (e.g. [229]). Some research dealing exclusively with more efficient hydraulic simulation was already presented in Chapter 2.5.

As multi-core processors and powerful graphical processing units (GPU) with general purpose programming facilities, such as Nvidia's CUDA and open computing language (OpenCL), have become more commonplace, there has been a growing interest in parallelizing hydraulic simulators and re-implementing the matrix operations on GPU, and thus reducing the computational time needed.

There have been several attempts to utilize better the multiple CPU and GPU cores available in the modern workstations. Two kinds of approaches are involved: improving the performance of the EPANET simulator by parallelizing the simulator or improving the performance of the optimization algorithm by running several distinct simulator processes in parallel [162].

Guidolin et al. [109] implemented the EPANET solver, the conjugate gradient method on GPU using sparse matrices. According to the paper, there is potential to reduce the computational time for repetitive runs.

Wu and Lee [301] replaced the linear equation solver in EPANET with a parallel version and compared how the parallelized version performed solving hydraulic models with 1000 to 150 000 pipes. While the matrix solver performance increase was significant, overall efficiency was significantly reduced by introducing the new solver. The slowdown was more pronounced with smaller models. Even with a very large model, only 29.7 % of the processor time is spent solving the network equations. Actually 19.1 % of the time is spent opening and closing the model and saving the results, which serves as a hint of major simulation time savings potential there. [301]

Wu and Zhu [298] and [149] use distributed and parallel simulations, respectively, to reduce the computational time required by the optimization. Von Lücken et al. [282] used asynchronous parallel multi-objective optimization. The parallel optimization framework used in [298] was later generalized for any parallel GA based optimization in [303].

Ibarra and Arnal [122] formulate an implicit pump scheduling problem as a mixed integer programming problem and use computational infrastructure for operations research (COIN-OR) toolkit to solve it using parallel processing. The solution is obtained using the branch and bound method. The method was applied to a small part of WDN of Granada, Spain, and the reported savings were 20 %.

Using multiprocessing via open multi-processing (OpenMP) and message passing interface (MPI) reduced the computational time almost linearly as the number of cores increased.

Broad et al. [50] used ANN for optimal control of water supply systems. The ANN method was several orders of magnitude faster than using EPANET for the Wallan, Victoria, Australia network. Behandish [30] used multiple ANNs, trained using GPU, for extended period simulation.

Razavi et al. [215] compared the computational performance of various surrogate models using Griewank, Ackley, Rastrigin and Schwefel functions as test functions. The paper concludes that using surrogate models is not always a proper solution for coping with limited computational budget. Choosing a suitable surrogate model is not simple, and a bad choice can be counter-productive. For example, in [30] accumulated tank level error from ANN usage was about 0.5 m to 1.0 m in the presented case study over a 168 h period. Still, surrogate models are widely used in operational optimization, as they can be up to 700 times faster than full-scale hydraulic simulation [49].

Paluszczyszyn et al. [191] and [192] present a model simplification methodology that can be applied online, thus enabling the real-time modeling. Compared with [267], [191] adds support for multi-threaded simplification, which allows simplification of a 3500 node network in 1 min to 37 min depending on the number of threads. The method compares both the hydraulic equivalence and energy distribution characteristics of the simplified and original models.

Hakimi-Asiabar et al. [113] uses self-learning (using self-organizing maps, SOM) multi-objective GA for optimizing reservoir operations. The method shows clear improvement of results quality over NSGA-II in the studied case: NGSA-II had to be run for 1000 generations and the run took 23 min while the self-learning genetic algorithm variant (SLGA) took only 100 generations and 6 min to satisfy the stopping criteria.

Zheng et al. [313] used the graph decomposition method to solve the design optimization problem more efficiently. The network is partitioned and each subnetwork is optimized separately using differential evolution. The method performed very well in terms of the optimized costs and computational time, and could be applied in the operational optimization setting to speed up the optimization.

Computational budget can also be saved by introducing preemption. When the objective function value is monotonic, and the optimization algorithm does not require the final objective function value, the hydraulic simulation can be preempted as soon as it becomes apparent that the result would be worse than the current best value. According to [214], preemption can save up to 60 % of the computation time. The methodology, however, cannot be used with many meta-heuristic algorithms, because the algorithms, for example GA and ACO, require that the final objective function value is known.

Pasha and Lansey [196] state that developing good initial solutions that already approximate the optimum, the computation time required by the stochastic optimization algorithm can be reduced significantly. The speed-up can be especially noticeable, when the optimization is combined with a surrogate model.

Pasha and Lansey [195] used LP to generate warm solutions for SFLA based stochastic optimization to speed up the optimization process. The optimization time is further reduced by using support vector machine as a surrogate model to avoid full EPANET simulations. The method was tested on the Anytown network. Unfortunately, no computation time information was published. The paper also recommends the use of previous day's optimal solution as a warm initial solution.

Jung et al. [136] uses GA with a skeletonized model and explicit SSP pump formulation. They use previous hour's results as a warm initial solution to speed up the optimization process.

Developing a surrogate model for full hydraulic simulation requires much effort and typically sacrifices some accuracy. The performance gains, however, can be significant. Parallelized versions of the GGA hydraulic solver or using a GPU does not yield significant speed-ups. New developments in hydraulic simulation, as discussed in Chapter 2.5, such as graph decomposing, hold a lot of promise, but it will take some time before practical implementations are readily available.

Avoiding unnecessary calls to the hydraulic simulator, avoiding IO-operations and caching as much of the simulator state as possible between different calls to the simulator, preempting simulation when the solution is proven to be bad, and generating good initial solutions for optimization algorithms are more easily implemented and provide more generally applicable solutions.

This work implements some code level optimizations and heavy compiler optimization for the simulator, avoids much of the IO of the simulations, uses preemption [214], results from previous optimization run as an initial solution for the next (as in e.g. [213] and [136]), and avoids the use of surrogate models. One goal of this thesis research is to show that it is feasible to use full-scale hydraulic models in conjunction with real-time optimization. The applied methodology is described in Chapter 3.11.

2.8.5 Real-time operational optimization frameworks

The aim of the real-time operational optimization is to minimize costs, energy usage or chemical consumption while ensuring sufficient quality of service for the consumers by varying the control parameters that can be changed remotely

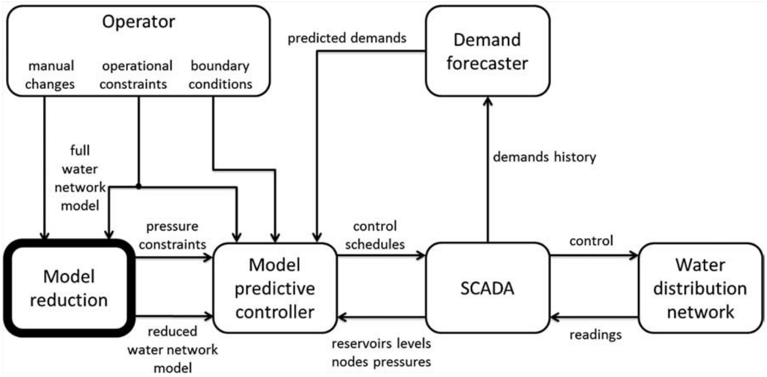


Figure 10. Control scheme for online energy and leakage management using skeletonized hydraulic model [192]

by the SCADA system in use. Typical control parameters include pump and valve settings. One example of a general model predictive control (MPC) system for energy and leakage management is shown in Figure 10.

The complete real-time optimization framework includes all the elements needed for the optimization: SCADA connection, demand forecasting, optimization algorithm, and solution evaluation module. The literature review performed, unfortunately, yielded only a few articles describing complete real-time operational optimization frameworks.

Bunn [51], Bunn [52] and Thorstensen [258] examine the benefits of using Decreto's online pump scheduling and operational management system in various US cities. The publications show that significant energy savings of 10 % to 15 % are possible in real systems, but the implementation details are not documented.

Zhao et al. [311] present a general framework for the online analysis and operational optimization of WSS. The framework has been in use for two years, and energy savings of 3.4 % have been reported, along with much fewer pipe burst, smaller leakage and better service pressure.

Odan et al. [179] describe a real-time optimization framework. Demand forecast is calculated using the DAN2-H algorithm and the operational optimization is done using the AMALGAM algorithm. Pump schedules are formulated as time-triggers. The optimization was multi-objective, including pumping costs and various reliability measures. Their case-study showed cost savings of 13 %.

Jamieson et al. [130] describe the POWADIMA project that developed a generic real-time operational optimization framework. The various aspects of the framework were presented in more detail in various papers. Rao et al. [211] and Rao and Salomons [213] focus on the optimization using the GA and the ANN surrogate model. The design variables are pump on-off statuses and valve settings for a 24-hour period. The optimization was applied to a hypothetical modified Anytown network [212, 213] and real networks Haifa-A [234] and Valencia [169]. For Haifa-A case, each solution had 408 bits, population size was 50 and the number of generations was 1000. The network model used for Haifa-A had 112 nodes. Using the GA-ANN, method the 24-hour optimization took about four minutes, which was 25 times faster than using the full hydraulic model with GA. The reported energy savings were about 20 % for both Haifa-A and Valencia cases.

The Haifa system was further optimized in [240] using the GA with a skeletonized version of more accurate, 867 node, hydraulic model. The skeletonized model had only 77 nodes, while retaining much of the accuracy of the original model. The framework laid in [130] remained otherwise the same. The reported energy cost savings were 10 % and the reduced model was reported to be 15 times faster than the full model – one optimization run took about 15 min.

This work develops a real-time operational optimization framework with the structure similar to those published earlier, as there is not much room for improvement in the framework structure in itself. The framework is described in more detail in Chapter 3.1.

2.9 Conclusions

This chapter provided a general overview of the water supply system and the relevant hydromechanics. The chapter reviewed literature related to various subjects needed for constructing a real-time operational optimization framework. The covered subjects included pump energy use and pump energy optimization, hydraulic simulation, demand forecasting, and WSS optimization.

Compared to the optimization of water distribution network design, the operational optimization is a subject relatively little studied, especially in a real-time setting. Much of the challenges and solutions apply equally to both classes of problems. The requirements for computational performance are typically, however, more pronounced in operational optimization problems when near real-time performance is needed.

In the literature, major short-comings in many reported operational optimization methods are shown to lie in focusing mostly on fixed-speed pumps and the lack of accuracy: surrogate models are used, VSP energy consumption is only roughly approximated or sometimes calculated incorrectly, and raw water pumping and treatment or chemical costs are rarely included in the objective function.

Multiple meta-heuristic optimization algorithms have been successfully applied to both design and operational optimization, and their performance has been good. Classical optimization methods have shown good overall performance, but implementing them is more problem-specific. Various hybrid algorithms combining classical and meta-heuristic or different meta-heuristic algorithms have also been successfully applied and they have shown good performance.

One major reason for the use of rough approximations is certainly the relatively long time it takes to simulate a full-scale network using the current version of EPANET. The new developments in hydraulic simulation, especially the promising decomposition methods, can change this in the near future. Another main reason for simplifications and the use of small benchmark models is that authors focus on the optimization algorithm itself, and try to produce results that are easily comparable to earlier research. Thus, authors tend to report results using small but widely available models such as Anytown.

The next two chapters describe the real-time operational optimization framework developed in this thesis that builds upon and addresses some short-comings in the earlier research.

The approach chosen in this thesis research is to use a model as accurate as possible. According to the literature review, novel developments of this thesis in the field of real-time operational optimization in the hydraulic modeling are: a full-scale hydraulic model that includes raw water extraction, conveyance and treatment (see Chapter 3.5), controlled by the real control system model (see Chapter 3.4.2) used in conjunction with an accurate model for pump energy usage (see Chapter 3.3).

The accurate pump energy use model, integration of flow and pressure controlled variable-speed driven pump batteries in EPANET (see Chapter 3.4.1), and the use of globally pre-optimized pump battery pump and frequency configurations (see Chapter 3.3) ensure, together with the accurate hydraulic model, that results obtained by the optimization accurately present the system performance and that the solutions are feasible.

The performance of hydraulic simulations is ensured by optimizing the simulator both manually and using the best optimizations modern C compilers (in this case the GNU C Compiler, GCC) have to offer (see Chapter 3.4.4). Parallel processing is utilized in the optimization process via a novel way of making EPANET thread-safe by utilizing thread local storage (TLS) features of modern compilers (see Chapter 3.4.3) without making any API changes.

More gains in computational efficiency are achieved by the preemption of the objective function evaluation, and loading, and initializing EPANET only once, and reusing the same simulator for all evaluations while avoiding as much of file input and output operations as possible (see Chapter 3.11.1), which, according to the literature review, can make the simulation about seven times faster compared to the straightforward use of a simulator.

While the optimization framework structure (see Chapter 3.1) itself is quite conventional if not for its generality (see Chapter 3.12.1), and likewise, the demand forecasting algorithm (see Chapter 3.9) is not very special, the problem formulation and optimization algorithms offer considerable novelty.

Thanks to the control system modeling and the pump battery EPANET component and pre-optimization, the optimization framework does not have to optimize each single pump's frequency, but it can find out the station specific optional settings, which are then transformed into pump-specific frequencies by the control system model and pump battery pre-optimization. This enables the system to include every single pump in the system, while managing to keep the size of the search space reasonable.

This makes it possible to accurately model and include water production energy costs in the objective function. Objective function (see Chapter 3.7) includes not only the energy costs, but also other water production costs, like chemicals, which are not typically included in the objective function in the literature.

Further reduction of search space is gained by the novel way to formulate the decision variables (see Chapter 3.7.1) as a hybrid of explicit and implicit formulation on the station level. The formulation developed in this thesis research fixes the number of setting changes to four and imposes minimum run length for each setting to ensure better usability and feasibility of the settings. The formulation, however, also allows high level of freedom for the algorithm to choose the times when the different settings are used.

The optimization algorithm used is based on both DDS and HD-DDS that exhibit both global and local search properties, and according to the literature seems to provide good performance. The algorithm has not been applied in operational optimization problems before. This work modifies the algorithm somewhat to allow for solving MINLP problems and allowing for temporal non-greediness of the algorithm, as described in Chapter 3.8.

Warm initial solutions are used, thus providing the algorithm with a known good starting point from the result of the last optimization run (see Chapter 3.10). The simulation preemption is also used to avoid simulation of further timesteps after it is apparent that the solution candidate is worse than the currently known best solution. This reduces the computational time remarkably.

The work always runs several independent optimization runs in parallel and selects the best solution to ensure high level of certainty of the optimality of the solution. A certain level of population-based properties is thus brought to the otherwise single solution DDS algorithm.

3 OPTIMIZATION FRAMEWORK

3.1 Introduction

This chapter describes the generic optimization framework for real-time whole-cost optimization of water production and distribution developed in this thesis.

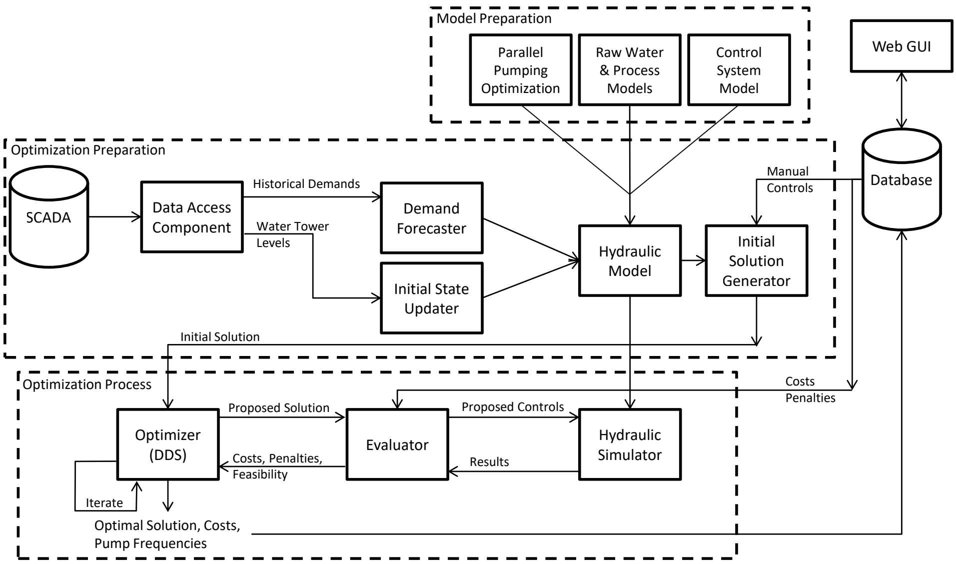


Figure 11. Structure of the developed optimization framework in its general form

Figure 11 shows the components of the optimization framework and their relations to each other. The optimization of a given historical or future time period can be initiated from the web user interface manually, programmatically using the provided Representational State Transfer (REST) Application Programming Interface (API), or using a scheduled task – as performed in the operational real-time setting.

The literature review in Chapter 2 shows, that while there is a considerable body of research addressing the operational optimization of a water supply system (WSS), only few papers focus on optimizing systems with variable-speed drive (VSD) controlled pumps.

The existing research typically ignores energy consumed by raw water extraction, conveyance and pumping at water treatment plants along with non-energy costs related to water production. Pump motor and variable-speed drive efficiencies are also typically neglected. Most of the published research uses simplified or surrogate models for the optimization instead of full-scale network models.

In order to accurately model and optimize the energy usage of the whole water supply system, the method presented here assumes the use of a full-scale hydraulic model, including all pipes, all pumps along with motor and VSD efficiencies (see Chapter 3.3), raw water extraction and water treatment processes (see Chapter 3.5) and the control system model controlling the pumps and valves in the WSS (see Chapters 3.4.2 and 3.4.1).

The method presented will achieve near real-time operational optimization using a full-scale hydraulic model by following:

1. reducing the number of design variables by using a novel problem formulation: only time patterns and four different flow settings are optimized on a station level (see Chapter 3.7.1 – treatment processes are driven by control system model (see Chapters 3.4.2 and 3.4.1), and internal pump battery optimization is done beforehand offline (see Chapter 3.3)

2. using highly optimized, parallel version of EPANET simulator (see Chapters 3.4.4 and 3.4.3)

3. using preloading, preemption and parallel processing to reduce computational time, when performing hydraulic simulations as part of the objective function and constraint evaluation (see Chapter 3.11.1)

4. using previous optimization results as warm initial solutions (see Chapter 3.10)

5. using a novel Modified Hybrid Discrete Dynamically Dimensioned Search (MHD-DDS) meta-heuristic optimization algorithm, which is efficient and supports a fixed number of evaluations (see Chapter 3.8).

The following sections describe the main components of the proposed framework in more detail, starting with the model preparation: pump train energy use model, parallel pumping optimization, EPANET enhancements and hydraulic model construction, optimization preparation: the tool for SCADA data access, and proceeds to the optimization problem itself: problem formulation, optimization algorithm, demand forecasting, generating initial solutions and solution evaluator. Finally, some implementation details and the concluding remarks are presented.

3.2 Pump energy use model

The pump energy model is also presented in the author's articles [253] and [254]. The related literature is reviewed in Chapter 2.3.

Variable-speed drive (VSD) can change the motor's and thus the pump's rotational speed N. VSDs are introduced in order to control the produced flow and pressure in an energy efficient manner. The introduction of variable speed drives allows for significant energy savings and more flexibility in the control of pumping. [164] The flows and pressures need to be controlled because the system parameters, for example water demand vary significantly over time.

Flow Q_2, head H_2 and power P_2 at some rotational speed N_2 are calculated using affinity laws.

Pump's hydraulic efficiency in relation to the rotational speed can be modeled using frequency scaling. More complex models are based on the Reynolds number [105], but if it is assumed that no losses are dependent on the Reynolds number, the frequency scaling function can be written as in [235]:

$$\eta_2 = 1 - (1 - \eta_1) \left(\frac{N_1}{N_2} \right)^{0.1} \quad . \tag{3.1}$$

While Equation (3.1) is approximate, according to [244], it provides reasonably good estimates if the rotational speed is not reduced more than 70 % from the nominal or the pump is small. The formulation is becoming accepted in the field [99].

Motor's efficiency depends on the load. The motor load

$$L = \frac{P_S}{\frac{P_{NOM}}{\eta_{M,100}}} \quad , \tag{3.2}$$

where P_{NOM} is the motor's nominal power and $\eta_{M,100}$ is the motor efficiency at the rated load. [5]

According to Equations (2.8) and (2.9c), shaft power P_S is approximately proportional to the cube of the relative rotational speed ω^3. Thus, lowering the pump's rotational speed to 50 %, lowers the shaft power – and the motor load – to about 12.5 %. The load L diminishes quickly as the rotational speed is reduced.

The exact motor efficiency η_M at different loads is motor specific, and typically the motor manufacturers provide load–efficiency curves. Generally, larger motors have higher efficiency, and higher efficiency motors can keep better efficiency at lower loads.

IEC 60034-31 [124] standard provides a general equation to calculate an approximation of motor efficiency at any partial load based on motor's rated ($\eta_{M,100}$) and 3/4 load efficiencies ($\eta_{M,75}$):

$$v_L = \frac{\left(\frac{1}{\eta_{M,100}} - 1\right) - 0.75 \cdot \left(\frac{1}{\eta_{M,75}} - 1\right)}{0.4375} \tag{3.3a}$$

$$v_0 = \left(\frac{1}{\eta_{M,100}} - 1\right) - v_L \tag{3.3b}$$

$$\eta_M = \frac{1}{1 + \frac{v_0}{L} + v_L \cdot L} \quad . \tag{3.3c}$$

Equation (3.3) can be used to approximate the motor efficiency when the exact efficiency curve is not available.

When the motor efficiency is known for the particular pump working point, the motor power

$$P_M = \frac{P_S}{\eta_M} = \frac{P_H}{\eta_H \cdot \eta_M} \tag{3.4}$$

can be calculated.

Based on experiments presented in [56] and [55], this work assumes that modern VSDs can mostly compensate the VSD generated losses in motors, and only VSD efficiency itself is considered as per IEC 60034-31 [124].

VSD load is calculated similar to the motor load in Equation (3.2). The VSD efficiency η_{VSD} is linearly interpolated from a lookup table constructed based on IEC 60034-31 [124]. The efficiency at various loads for VSDs of different nominal power is shown in Figure 12.

The pump train electrical power

$$P_E = \frac{P_M}{\eta_{VSD}} \quad , \tag{3.5}$$

and the total pump train efficiency

$$\eta_{TOT} = \frac{P_H}{P_E} = \eta_H \cdot \eta_M \cdot \eta_{VSD} \quad . [33] \tag{3.6}$$

The total electrical power for a pump expressed as the function of the working point (Q, H) becomes

$$P_E = \frac{P_H}{\eta_H \cdot \eta_M \cdot \eta_{VSD}} = \frac{g \cdot \rho \cdot Q \cdot H}{\eta_H \cdot \eta_M \cdot \eta_{VSD}} \quad . \tag{3.7}$$

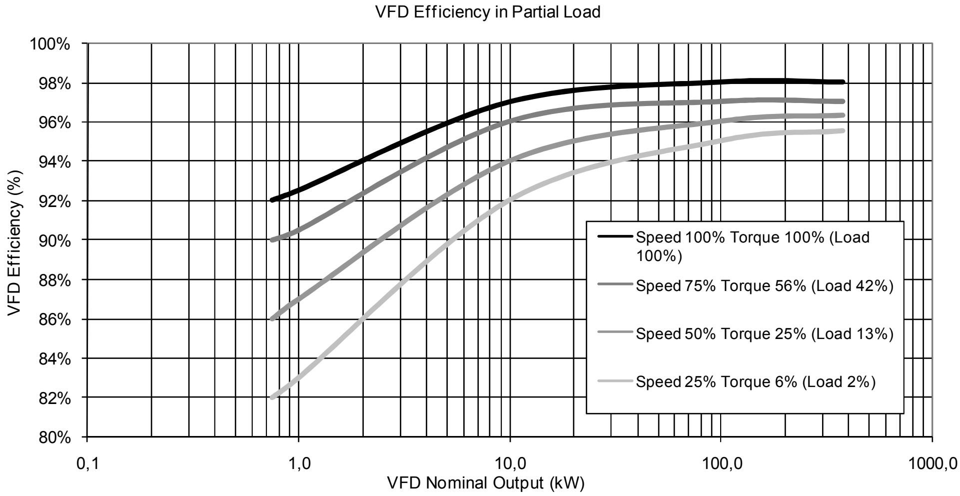


Figure 12. Typical VSD efficiency at different loads [124]

Table 6 shows how the load and different efficiency components change using the selected method, when pump's rotational speed is reduced in a zero static head system. The motor presented in the table is a 75 kW IE2 class motor, with a full load efficiency of 95.4 % and 75 % load efficiency of 94.6 %. The VSD is also 75 kW in power. Pump's best efficiency point (BEP) is 75 % at the nominal rotational speed at 50 Hz. It is assumed that pump's shaft power is 75 kW at BEP at the nominal rotational speed. While pump's BEP decreases from 75.0 % to 73.2 % when the rotational speed is reduced from 50 Hz to 25 Hz, motor's efficiency reduces from 95.4 % to 77.6 % and VSD's efficiency from 98.0 % to 95.3 %. This results in the total efficiency of 70.1 % at 50 Hz and only 54.5 % at 25 Hz.

Table 6. Different efficiency components at various loads and rotational speeds

Hz	Load	Efficiency			
		Motor	VSD	Pump	Total
50.0	100.0 %	95.4 %	98.0 %	75.0 %	70.1 %
45.4	75.0 %	94.6 %	97.9 %	74.8 %	69.2 %
39.7	50.0 %	92.8 %	97.3 %	74.4 %	67.2 %
31.5	25.0 %	87.2 %	96.5 %	73.8 %	62.2 %
25.0	12.5 %	77.6 %	95.8 %	73.2 %	54.5 %
18.4	5.0 %	58.2 %	95.3 %	72.4 %	40.1 %
14.6	2.5 %	41.1 %	94.9 %	71.7 %	28.0 %
10.8	1.0 %	21.8 %	94.6 %	70.9 %	14.6 %

3.3 Parallel pumping optimization

The pump optimization method is also presented in the author's articles [253] and [254]. The related literature is reviewed in Chapter 2.4.

The pump battery is described as a set of pumps. Each pump is given a characteristic curve, an efficiency curve, minimum and maximum allowed frequency, nominal motor power P_{NOM}, and either IE efficiency class and number of poles, for standard motor efficiency values based on IEC 60034-30[123], motor efficiency values at both 100 % and 75 % load, $\eta_{M,100}$ and $\eta_{M,75}$ respectively, or motor efficiency curve in tabular format. Minimum and maximum frequencies can be set equal, when no VDS is present or in use.

The power used by the whole battery of n pumps is

$$P_{TOT} = \sum_{i}^{n} P_{E,i} \quad , \tag{3.8}$$

where $P_{E,i}$ is pump i's electrical power use (see Equation (3.7)).

Mathematically, the problem of finding an optimal combination of pumps and their respective frequencies for a working point (Q, H) can be stated as

$$\min_{\bar{f} \in X} P_{TOT}(Q, H, \bar{f}) \quad , \tag{3.9}$$

where $\bar{f}$ is a vector of combinations of frequencies for different pumps and the search space X includes all allowed combinations of frequencies and pumps that produce flow Q and head H.

A parallel exhaustive direct search [119] is performed on the full pump battery working regime $(Q,\ H) \in \{Q_{min} \ldots Q_{max},\ H_{min} \ldots H_{max}\}$. For each working point $(Q,\ H)$, all allowed combinations of different pumps and their frequencies that can produce the flow Q and head H are considered, and for each working point, the optimal combination of pumps and their frequencies $\bar{f}$ in terms of total efficiency is chosen and stored in the results array.

First, each pump's working regime is determined. Minimum and maximum allowed head, and maximum allowed flow are calculated based on the pump characteristic curve and the allowed frequency range.

The calculation loops over the pump's allowed flow range for the frequency, and calculates matching the head and the total pump train efficiency η_{TOT}. If multiple frequencies result in overlapping working points in the $Q_{step} \times H_{step}$ resolution, the frequency that produces the highest total efficiency is chosen for that particular working point.

The results of the working regime calculation are stored in two pump specific lookup arrays shown in Equation (3.10). The first, **F**, contains the optimal frequency for all working points and the other, **H**, contains the total pump train efficiencies at those points. Arrays elements that present invalid working points are set to 0.

$$\mathbf{F} = \begin{bmatrix} f_{Q_1,H_1} & f_{Q_2,H_1} & \cdots & f_{Q_m,H_1} \\ f_{Q_1,H_2} & f_{Q_2,H_2} & \cdots & f_{Q_m,H_2} \\ \vdots & \vdots & \ddots & \vdots \\ f_{Q_1,H_n} & f_{Q_2,H_n} & \cdots & f_{Q_m,H_n} \end{bmatrix} \tag{3.10a}$$

$$\mathbf{H} = \begin{bmatrix} \eta_{Q_1,H_1} & \eta_{Q_2,H_1} & \cdots & \eta_{Q_m,H_1} \\ \eta_{Q_1,H_2} & \eta_{Q_2,H_2} & \cdots & \eta_{Q_m,H_2} \\ \vdots & \vdots & \ddots & \vdots \\ \eta_{Q_1,H_n} & \eta_{Q_2,H_n} & \cdots & \eta_{Q_m,H_n} \end{bmatrix} . \tag{3.10b}$$

Next, all the possible non-identical pump combinations are considered. For each combination the algorithm iterates over the allowed head range $[H_{min}, H_{max}]$ using the user-defined head step size H_{step}. Each head step H_i is added to a FIFO queue, where one of the processor threads picks it up for calculation.

A processor thread calculates all possible combinations of flows for the running pumps in the given pump combination for the head H_i. The flow step used in this step is $\frac{Q_{step}}{n}$, where n is the number of pumps running in the combination. Each pump's total efficiency is looked up from that pump's working regime array **H**. Every time there are multiple possible combinations that produce the same total flow, the one with best over all efficiency is chosen and stored in the results arrays.

The end result is two arrays that cover the full working regime of the whole pump battery. Each element represents an area defined by Q_{step} and H_{step}. Results array **C** contains the numerical presentation of the optimal combination binary string and **R** contains the optimal total efficiency of the pump battery:

$$\mathbf{C} = \begin{bmatrix} c_{Q_1,H_1} & c_{Q_2,H_1} & \cdots & c_{Q_m,H_1} \\ c_{Q_1,H_2} & c_{Q_2,H_2} & \cdots & c_{Q_m,H_2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{Q_1,H_n} & c_{Q_2,H_n} & \cdots & c_{Q_m,H_n} \end{bmatrix} \tag{3.11a}$$

$$\mathbf{R} = \begin{bmatrix} \eta_{Q_1,H_1} & \eta_{Q_2,H_1} & \cdots & \eta_{Q_m,H_1} \\ \eta_{Q_1,H_2} & \eta_{Q_2,H_2} & \cdots & \eta_{Q_m,H_2} \\ \vdots & \vdots & \ddots & \vdots \\ \eta_{Q_1,H_n} & \eta_{Q_2,H_n} & \cdots & \eta_{Q_m,H_n} \end{bmatrix} . \tag{3.11b}$$

Two naïve algorithms were implemented too, to facilitate easier comparison of various control strategies. Naïve 1 algorithm drives all running pumps with equal frequency, and naïve 2 algorithm adjusts only the last pump's frequency while the other pumps run at their respective maximum frequencies. Naïve 1 is the most common way to control parallel pumping in the field.

EPANET was modified to use the total efficiency calculated by the above method in all energy calculations instead of the default incorrect and inaccurate method.

The method is used for pre-computing the globally best combinations of running pumps and their frequencies for all sets of parallel pumps in the system to be modeled. The actual online-optimization then only needs to find the best settings on the station level, as the stations know what the most efficient way is to drive the pump battery at the station in order to produce the required flow or pressure.

3.4 EPANET enhancements

This section describes the enhancements developed for EPANET as part of this research. As shown in the literature review (see Chapters 2.3 and 2.4), EPANET lacks a proper component for modeling variable-speed controlled pumps and at reduced rotational speeds the pump energy consumption is calculated incorrectly. Together, the pump battery component and control system modeling framework allow accurate modeling of raw water extraction, conveyance and treatment, while reducing the number of decision variables.

Hydraulic simulation is also typically the bottleneck in the meta-heuristic optimization (see Chapters 2.5 and 2.8 in the literature view); thus EPANET was made thread-safe and various other computational speed enhancing optimizations were applied to enable the use of full-scale model in the optimization.

3.4.1 Pump battery component

EPANET has a pump component that can be used for modeling pumps - both single pumps and pumps working in parallel or series. EPANET provides means to control individual pump's status and the relative rotational speed ω but there are no means to directly regulate the flow or pressure. The related literature review is presented in Chapter 2.5.

In order to control the pumps based on flow or pressure, it is required to implement PID controllers externally using the EPANET toolkit. PID-control, however, requires that the hydraulic time step is a fraction of second rather than the typical time step of several minutes or an hour. Tuning the controller parameters can also be a time-consuming task.

Use of a short time step increases the computational time and makes the simulation more mathematically unstable because more numerical inaccuracies accumulate over the simulation as the number of steps increases.

Author's paper [256] introduces a new pump battery component into the EPANET hydraulic solver. The component enables one to model a pump battery consisting of one or more possible non-identical pumps working in parallel. The battery can be either flow, pressure or head difference controlled, and the control mode, setting and limit can be dynamically controlled using both application programming interface (API) and EPANET control rules.

To allow efficient and advanced pump battery analysis and optimization, the pump battery component in EPANET is mathematically very simple. The component only calculates the head and flow required to meet the given setting and limit in the active controlling mode.

The component also accepts a limit to the non-controlled parameter, for example if the pump battery is flow controlled, maximum allowed downstream pressure can be limited to a user-supplied value, typically 80 or 100 meters of pressure head. Alternatively, in constant pressure mode of operation the maximum allowed flow can be limited. In practice, especially when operating in constant flow controlled manner, the maximum allowed pressure is limited in order to avoid pipe breakage when the flow falls below the setting.

The component uses an externally defined callback function to check that the pump is working in an allowed regime, and the program running EPANET simulation or utilizing the hydraulic results calculates the internal pump configuration, each pump's frequency and energy consumption, based on the simulated head and flow. Thus, the more complex and time-consuming tasks are delegated to external code. The optimizing implementation of the back-end is presented in Chapter 3.3 and in the author's papers [253] and [254].

The callback is registered within EPANET using new API ENsetbattery-callback(int (*battery)(int, int, double, double)). The callback function is called every iteration from linkstatus() function in hydraul.c module for each pump battery in the model. The arguments are pump battery link index, current status, current flow, and current head, respectively. The callback must return the new status for the battery in question. Typical return values include OPEN, XFLOW and XPRESSURE for a normally functioning pump, invalid flow and invalid head, respectively.

The pump battery component is described in EPANET *.inp file by identifier, and start and end nodes. Optionally, the initial control mode (constant pressure/flow), initial setting (flow or pressure), and pressure/flow limit can be specified. An example is shown in Listing 1.

Listing 1. Example of defining pump batteries in EPANET inp file. First battery has the initial control mode set to constant flow at 10 $\frac{\mathrm{l}}{\mathrm{s}}$ with the pressure limit of 80 m and the second has no initial values and is initially closed.

```
[BATTERIES]
Battery1 Reservoir1 Junction1 TYPE FLOW SETTING 10 LIMIT 80
Battery2 Reservoir1 Junction2
```

The changes required in the EPANET source are minimal and localized. Besides introducing a new component type, the new link values and the code to read battery specifications from the *.inp file, a few new functions are added into the hydraul.c module: batterycoeffs(), which is called by newcoeffs(), and batterystatus(int index, char status, double h1, double h2), which is called by linkstatus(). The matrix coefficients in the global gradient algorithm[260] are calculated by batterycoeffs() and batterystatus(...) only changes the battery status based on the hydraulic results, and calls the possible external callback function to check that the battery is working within allowed regime.

When the pump battery is in the flow control mode or the flow limit is exceeded in constant pressure or pressure difference mode, the pump battery works similar to the flow control valve in EPANET, but the head loss over the link is allowed to be negative. The EPANET system matrix A coefficients [260, 226, 261] are

$$p_{ij} = \frac{1}{10^8} \tag{3.12a}$$

$$A_{ij} = A_{ij} - p_{ij} \tag{3.12b}$$

$$A_{jj} = A_{jj} + p_{ij} \tag{3.12c}$$

$$A_{ii} = A_{ii} + p_{ij} \quad , \tag{3.12d}$$

where i is the index of the start node, j is the index of the end node, p is the inverse of derivative of the head loss over link, and Q_{set} is the flow setting.

Correction terms in the EPANET solution system are set to

$$F_i = F_i - Q_{\text{set}} \tag{3.13a}$$

$$F_j = F_j + Q_{\text{set}} \tag{3.13b}$$

$$y_{ij} = Q_{ij} - Q_{\text{set}} \quad . \tag{3.13c}$$

When the pump battery is in the pressure or pressure difference control mode or the pressure limit is exceeded in the constant flow mode, the pump battery works simlar to the pressure reducing valve in EPANET, but the head loss over the link is allowed to be negative. The EPANET system matrix coefficients are

$$p_{ij} = 0 \tag{3.14a}$$

$$F_j = F_j + 10^8 \cdot H_{\text{set}} \tag{3.14b}$$

$$A_{jj} = A_{jj} + 10^8 \quad , \tag{3.14c}$$

where H_{set} is the head setting.

The EPANET API was extended to allow changing the mode and limit value, and to allow setting the callback function, which can check that the pump is working in the allowed regime and can limit the generated head and/or flow if necessary.

The added link values are named `EN_MODE`, accepting settings `CONST_FLOW`, `CONST_PRESSURE` and `CONST_DIFF`, and `EN_LIMIT`, accepting flow limit in model units when the battery is operated at constant pressure or constant pressure difference mode, and the pressure limit in model units when operated in constant flow mode. The values can be queried and set using the standard `ENgetlinkvalue` and `ENsetlinkvalue` functions.

The new `EN_MODE` setting was also implemented for valves, so that the control valve type can be dynamically changed between flow control valve (FCV) and pressure reducing valve (PRV).

The output from EPANET for a given pump battery is the time dependent working points (Q_t, H_t), setting and mode of operation. Thus, the higher level simulation package must implement some means to show and analyze each pump's properties, such as frequency, efficiency and power consumption at different working points.

In this research, the battery's internal state is checked from a lookup table generated by the parallel pump optimizer (see Chapter 3.3). The chosen method allows modeling the pump battery consisting of non-identical pumps with different allowed frequency ranges and different parallel pump control strategies: equal frequencies for all running pumps (naïve 1), only the last pump's frequency is controlled (naïve 2) or a globally optimal control strategy. In addition, the methodology handles the frequency scaling problem [244] and can model the pump's motor and variable-speed drive efficiencies, and thus give very accurate approximation of the real energy usage.

An example of a simple model and its results is shown in Figure 13. The pump battery's constant pressure setting changes to a higher setting for 8 am to 9 pm time period using EPANET control rules. The left-hand side of the figure shows the model and the water demand at the far end node, and the right-hand side shows the simulated head at the pump battery discharge node and the pump's relative speed.

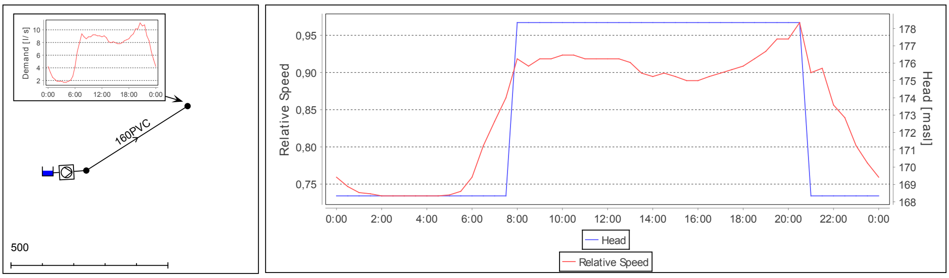


Figure 13. A small sample model of a pump battery working with different pressure settings and varying flow. The changes in the pump outlet head and relative pump speed are shown in the figure.

Every pump battery and flow or pressure controlled pump in the water supply system to be optimized is modeled using the pump battery component with the actual pump characteristic and efficiency curves, and with the correct motor size and efficiencies, and the battery is to be driven by the solutions proposed by the system optimization algorithm. Therefore, the globally optimal control for the battery's inner operations can be utilized.

The energy calculations used for EPANET's ordinary pumps were also changed to use the same back-end as the pump battery so that correct efficiency calculation, frequency scaling, and motor and VSD efficiencies can be taken into account.

3.4.2 Modeling of the water supply control system

EPANET provides only rudimentary tools for modeling the control system behavior. The tools offered are "controls" and "rules" that can change valve and pump setting, and open and close pipes based on time or some hydraulic variables. [226] Both control mechanisms are limited to changing the settings to a predefined constant values only. Thus, for example, using a PID controller requires implementing the control externally and interfacing EPANET from outside the simulation. Often, these control system models are built and executed from Matlab (e.g. [290] and [266]).

Author's paper [256] presents a control system modeling framework, originally developed in the Master's thesis [252]. The framework embeds a Python interpreter into EPANET. In order to allow for modeling complex control system algorithms, a Python 2.7.x programming language based framework was built.

Python is widely used [3, 2] modern multi-paradigm general purpose programming language. It supports both object oriented and functional programming, and it has very extensive built-in library and extensive set of third-party libraries. Python has arguably a low learning curve, and the programs and scripts written in it tend to be terse compared to languages like C or C++. [154]

Python is an interpreted language, which means that no tools other than text editor are needed for developing Python programs and libraries. The interpreter is easily embeddable in C programs [174], making it suitable for use as a scripting language for other programs. Implementations of Python exist for other languages too: Jython for Java [135] and IronPython for Microsoft's .NET-platform [94], which make it easy to embed and extend Python using those languages too and use either Java Virtual Machine (JVM) based or Common Language Runtime (CLR) based libraries from Python code.

The interpreted nature, feature set, easy embeddability, ease-of-use, strong set of programming libraries and popularity make Python a good choice for the control system model programming language. The method presented here makes the control system model code an integral part of the EPANET simulation process.

In `ENopen` function, the Python framework is initialized, and a Python module is searched, identified by the same filename as the EPANET model but with `*.py` extension. If the module is found, it is loaded using the Python interpreter and function pointers to `epanet_init`, `epanet_callback` and `epanet_close` functions are retrieved. During the Python module load, the module can import and use other Python modules and libraries, such as xlrd [159] for reading the control system parameters from MS Excel spreadsheet files.

After the hydraulic simulation is initialized in `ENopenH` function, the loaded Python module's `epanet_init` function is called. The function can then instate the EPANET link and node objects that are required for its functioning. Typically, this phase finds the indices of the controlled pump batteries and valves, and components representing the measurements needed in the operation in the EPANET simulator. The init function also sets initial settings for all controlled components.

After each simulation time step, in `ENstep` function, the Python module's `epanet_callback` function is called. The function can query the system state and alter settings for different components. This callback is where the control system model done in Python language is given full control over the simulated system and all the control algorithm calculations take place.

The framework provides mapping of the standard C language programming interface available in EPANET in a higher level object oriented Python API. The EPANET errors are mapped into Python exceptions and the node and link properties are accessed through an object oriented wrapper, part of which is shown in Figure 14. Properties are either read-only or read-write, depending on whether the parameter can be changed or not. A lower level 1:1 Python mapping to the C API is also available, but its usage is not recommended.

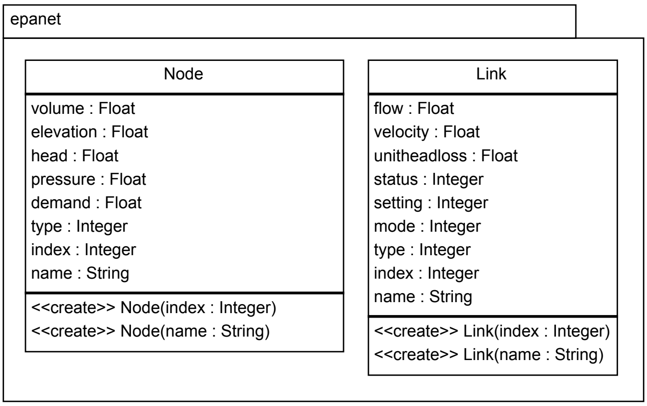


Figure 14. Part of the Python language object oriented wrappers around the low level EPANET API

Using the API, network state can be queried, controlled and altered during the simulation. It is possible, for example, to query the flow, head and pressure, and tank level and volume. The API allows open and close pipes, change valve and pump settings, and control pump batteries. Demands and emitter coefficients can be changed too. But in order to remain strictly a control system model, only those components that can be controlled in real world should be controlled.

Finally, when the hydraulic simulation is completed, the `ENcloseH` function calls the `epanet_close` function, which can, for example, store internal control system state results to a file for later analysis. After the call, the Python interpreter is closed.

The simulation and calling the control system model is wholly controlled by the EPANET simulator, and thus the use of the control system model is transparent to any program using the simulator. While the control system model can query and set hydraulic model parameters during the simulation, it cannot control the simulation in any other manner.

The control system model code can be divided into multiple modules which can call each other and EPANET at will, and all tools, libraries and programming techniques available in Python can be used freely. Typically, it is reasonable, for example, to create classes to present various system components or to read control system parameters. Porting code from any programmable logic controller (PLC) or supervisory control and data acquisition (SCADA) system is straightforward, and designing common libraries for often used components is easy.

A very simple example of a control system model is shown in Listing 2 and Figure 15. Pumping into the network is flow controlled, and the flow is linearly interpolated between minimum and maximum flow values based on the water tower level, such that when the water tower is at the upper level, the flow is minimal and vice versa. While the example is simple, similar control is commonly used, and cannot be implemented with the EPANET control rules. The example demonstrates some potential of using a general purpose programming language as a control system modeling tool.

Listing 2. An example of a control system model that interpolates pump battery flow setting based on a water tower level

```
import epanet

MIN_LEVEL = 2.0  # meters
MAX_LEVEL = 4.5
MIN_FLOW = 3.0   # l/s
MAX_FLOW = 13.0

def epanet_init(filename):
    global watertower, battery
    watertower = epanet.Node('WATERTOWER')
    battery = epanet.Link('BATTERY1')
    battery.mode = epanet.Link.CONST_FLOW
    epanet_callback(0)

def epanet_callback(time):
    global watertower, battery
    level = watertower.pressure
    if level >= MAX_LEVEL:
        battery.setting = MIN_FLOW
    elif level <= MIN_LEVEL:
        battery.setting = MAX_FLOW
    else:
        dL = MAX_LEVEL - MIN_LEVEL
        dQ = MAX_FLOW - MIN_FLOW
        battery.setting = MAX_FLOW - dQ * (level - MIN_LEVEL) / dL
```

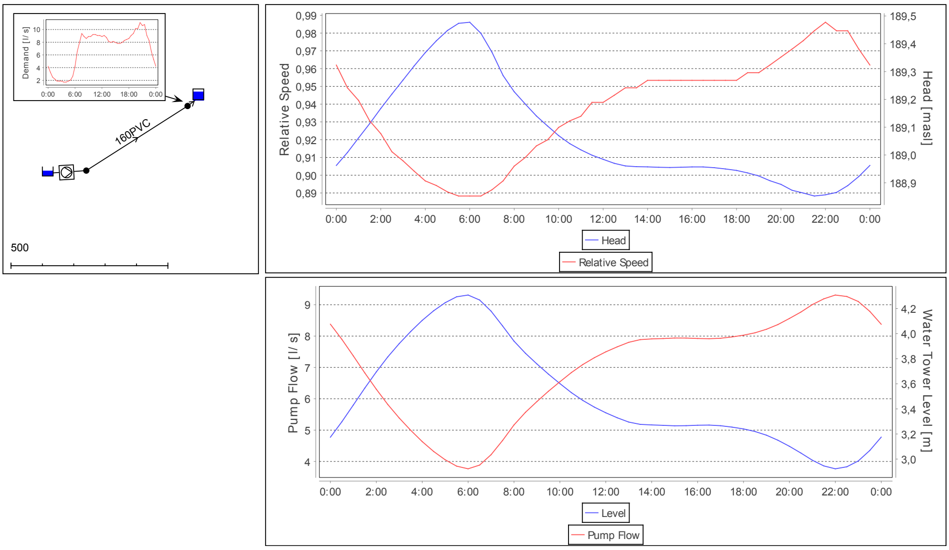


Figure 15. An example of control system model, that interpolates pump battery flow setting based on a water tower level.

In this work the control system model is mainly used for controlling the modeled raw water extraction and pumping inside the water treatment processes. Every water source typically has a clear well, from which the water is pumped using pump batteries to different parts of the network.

Typically the water treatment processes utilize constant level control: the clear well level is kept constantly close to the maximum level. The raw water extraction and any pumpings in the treatment process are flow controlled. The flow pumped into and through the treatment process is directly proportional to the flow pumped into the network from the clear well, and often greater than the network pumping. For typical surface water sources, the extracted and processed volume is about 10 % greater than the volume pumped into the network. This extra volume must be accounted for, in order to calculate the correct energy use.

For this purpose, a simple Python module was developed. It accepts a description of the system as a list of water sources. Names of the network pump batteries, internal pump batteries, valves, representing the hydraulic losses in the process, and raw water batteries are specified along with the raw water flow coefficient used for calculating the flow setting for each pump based on the network pumping. The code automatically sets the flow settings for all the components based on the amount pumped into the network and the water water flow coefficient.

The method enables calculating the energy use of the raw water extraction and treatment without introducing new design variables.

3.4.3 Parallel EPANET

EPANET uses global variables extensively, and as such only one thread can use the simulator at once. This is especially problematic in the Java EE environment, where multiple requests can be made in parallel. The limitation also means, that EPANET cannot be readily used to evaluate multiple solutions in parallel.

To alleviate this, several rewrites of EPANET have been proposed, such as those in López-Ibáñez et al. [149], Guidolin et al. [108] and Baseform's Java implementation of EPANET [1] to properly encapsulate the state in a variable that is passed along the calls to the simulator engine, thus making EPANET thread-safe. Besides CWSNet, the thread-safe variants use General Public License (GPL), making them unsuitable for proprietary development. It was determined that making only small modifications to the stock EPANET and avoiding any changes to the EPANET API would be an optimal solution.

The EPANET simulator was made thread-safe by marking all the 192 global variables with thread local storage (TLS) [59, 83] storage-class modifier `__thread`. An example of required modifications around the matrix variables is shown in Listing 3. The storage-class modifier instructs the C compiler to produce automatically code that makes the variable thread-local meaning that every thread has an own copy of the variable. While the official `thread_local` TLS storage-class modifier was standardized only in 2011 in C11 defined by the ISO/IEC9899:2011 standard [126], most of the C compilers have supported the modifier, `__thread`, as compiler specific extension for years.

The changes make the EPANET library completely thread-safe, though not re-entrant. Only the EPANET API functions were exported and link time optimizations were utilized, which together allow the compiler to the emit most efficient code for thread-local variable access, such as initial executable or local excutable access model [182, 4].

Listing 3. Some examples of the use of thread-local storage-class modifier __thread

```
[...]

EXTERN __thread double *Aii,   /* Diagonal coeffs. of A          */
                       *Aij,   /* Non-zero, non-diagonal coefs. */
                       *F;     /* Right hand side coeffs.        */
EXTERN __thread double *P,     /* Inverse headloss derivatives  */
                       *Y;     /* Flow correction factors        */
EXTERN __thread int    *Order, /* Node-to-row of A               */
                       *Row,   /* Row-to-node of A               */
                       *Ndx;   /* Index of link's coeff. in Aij */

[...]
```

3.4.4 Optimized EPANET

The EPANET simulator was compiled with highest level of optimizations available in the new GCC 6.2 C-compiler. The compilation was done especially for the target machine, latest generation of Intel Xeon processor, by specifying `–march=core-avx-i` machine architecture and enabling all SMID and other extended floating point operations: MMX, 3DNOW, SSE1–4.2, SSSE and AVX. Some examples of additional optimizations include the use of `–fast-math` and `–fno-math-errno` flags which reduce the time required for floating point operations. Linking time optimization `–flto` also provides measurable increase in the execution speed. According to [150], using GCC's `-O3` optimization level provides about 30 % speed-up.

Most of the internal EPANET functions were marked as `static inline` to enable the compiler to further localize and optimize their usage.

A memory leak that was small but significant over time, was fixed in the `saveoutput` function in `output.c` that allocates an array of size $max(Nnodes, Nlinks)$ but fails to free it upon return.

Some time-consuming and redundant operations were reduced. For example, the `linsolve` function in `smatrix.c` allocates three arrays of size `Njuncs` each time it is called (on each iteration). The size of the arrays is the same each time. Instead of allocating and freeing the buffers multiple times, the code was modified to allocate the arrays only once in `allocsparse` and free them on the simulator close in `freesparse`. The arrays are then only zeroed out on each call to `linsolve` using `memset` function calls.

Other code optimizations and changes for EPANET that are described in Appendix D in López-Ibáñez [150] were included to further reduce the simulation time and correct some aspects of the EPANET simulator. Problems in energy calculations were also corrected and frequency scaling applied as per Marchi and Simpson [165] and Simpson and Marchi [244].

As shown in Wu and Lee [301] and Table 2, 19.1 % of total running time of an EPANET simulation is spent opening and closing the model and saving the results. All these steps are completely unnecessary in the optimization process, if all the model parameters can be changed dynamically, which is the case in the methodology presented in this thesis research. Thus, calling the simulator appropriately, i.e. opening the model once and making multiple hydraulic simulations on the same model by calling only ENinitH, ENrunH and ENnextH functions repetitively, can save a considerable amount of simulation time.

3.5 Constructing hydraulic model for optimization

The objective function and constraints are evaluated using a hydraulic model. The accuracy of the model limits the accuracy of the results. One of the key elements of this thesis is to use most accurate methods available to model the system behavior and energy use. This section details some specific elements of the hydraulic modeling that have to be taken into account.

The hydraulic model used in the optimization framework ought to be a calibrated, full-scale model of the network that includes all pipes in the system, except the consumer connections. Every pressure zone of interest must have one demand pattern that is used by bulk of the demands in the zone. Separate patterns can be used for water users with well-known usage patterns that are different from the general pattern.

Every pump transferring bulk water in the network, including pumps used in raw water extraction and inside water treatment processes, is modeled along with the pump's efficiency curve, motor and VSD type and efficiencies in order to correctly model and optimize the total energy usage of the pumping (see Chapter 3.3).

Whenever a set of pumps can be variable-speed controlled based on flow or pressure, the pumps are modeled using the pump battery component presented in Chapter 3.4.1. In this way, the pumping can be easily controlled by the simulator, and the internal workings of the battery can be pre-optimized separately offline.

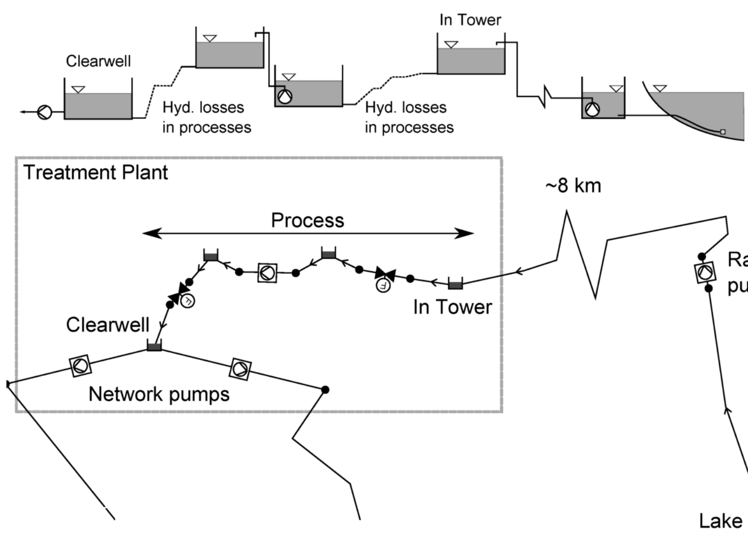


Figure 16. An example of how a water treatment plant is fully modeled: profile of the raw water extraction and treatment process, and the resulting hydraulic model

Figure 16 shows an example of how a water treatment plant, raw water extraction and the hydraulic head losses inside the treatment process are modeled. The head levels in the reservoirs match the actual or typical target head values. Flow control valves are used to model the power lost due to the head loss in the treatment operations between the constant head reservoirs. All pipe sizes and lengths and pump parameters match those of the actual plant. Suction sieve is modeled using a pipe with correct minor loss coefficient.

There must be a control system model (see Chapter 3.4.2) present, that controls pumping inside the water treatment processes and raw extraction similar to the actual system to be modeled. It is especially important that the raw water extraction and treatment processes pumping include any excess water required by the process but not pumped into the water distribution system. In particular, surface water treatment typically requires around 10 % of the raw water for the process.

All stations must be modeled to the same detail in order to obtain accurate and equivalent results for energy consumption.

EPANET is known to exhibit problems with control devices, such as valves and the pump batteries [78, 243]. The problems become especially apparent, when the devices are flow controlled, and too much or little water is supplied into the system.

In order to keep the model mathematically stable even when extremely infeasible solutions are simulated, every pressure zone in the model must include a low-head feeder reservoir and a high-head discharge reservoir connected to the network via check-valves. Producing too much water causes extra costs and violates water tower maximum level limit, and producing too little water violates water tower minimum level and capacity, and node minimum pressure limits, making such solutions highly unfavorable.

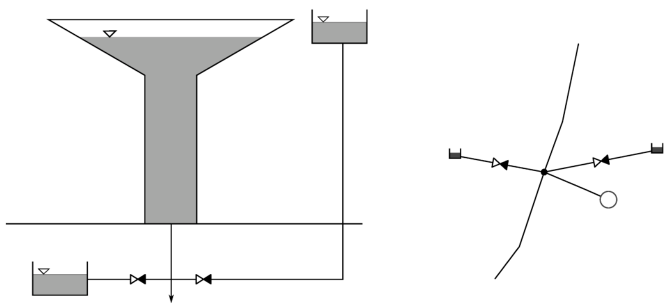


Figure 17. An example of how extra reservoirs can be connected to the modeled network to ensure mathematical stability of the simulation

Figure 17 shows an example of the reservoirs connected to a pressure zone close to the water tower. The low-head reservoir supplies the pressure zone with water if the solution does not ensure sufficient supply. Because the reservoir has very low head, the minimum pressure and minimum water tower capacity penalties ensure that these solutions will be very expensive and will thus be avoided. Likewise, the high-head reservoir accepts any extra water, when the demand is less than the volume pumped into the system, and there is no water tower in the zone or the tower is full. High head ensures that pumping energy costs and maximum pressure and water tower maximum level penalties will be high, and the solution will be avoided.

3.6 SCADA data access

Chapter 2.5 reviews some literature related to online modeling and the WSS SCADA data access. SCADA connection is needed in the optimization process for two reasons. First, the initial state for the system before optimization has to be fetched. The required information consists of the water levels in each water tower. Second, the connection is used for calculating the water usage in all demand measurement areas (DMA) and pressure zones. The historical demand information is used for producing a demand forecast for each area in the network. Both the initial levels and the forecast are set in the hydraulic model used for simulations and evaluating the objective function.

A tool [255] was developed for accessing and analyzing the SCADA data. It was developed in Java programming language version 8 [8], and it provides both a graphical user interface (GUI) for end users and an API for developers.

The software can connect to a variety of different data sources that can present any numerical data in a time series, via different APIs, including but not limited to SCADA system connections via direct SCADA API usage or Open Process Control (OPC), to relational databases and SCADA systems using Java Database Connectivity (JDBC) or Open Database Connectivity (ODBC), to tab and comma separated files and Excel-worksheets, and to various laboratory and customer information systems. The data sources can have different time zones, and different and time-varying time resolution.

An Extensible Markup Language (XML) configuration file describes the data sources and describes which values are available and how those values are calculated based on the data read from the sources. The calculations can include, for example, calculating a water balance for a pressure zone based on the flows in and out of the zone and changes in the possible water tower volume.

Each position can freely perform calculations on data from all declared data sources. The raw data can be either lagged or interpolated at this stage to cope with varying time intervals in different sources. The expression language supports all typical arithmetic operations and mathematical functions, such as `floor`, `ceil` and `sqrt`. In addition, the expression system supports both boolean algebra and time algebra.

An example of a configuration file fragment defining positions is shown in Listing 4. In the example, the demand for Pressure Zone 1 is calculated as the difference between incoming and outgoing flows to the area defined by the `in` and `out` attributes at station definitions. The water tower flow is calculated as the volume difference divided by the time between two measurements in hours. Station 100 pumps water out of Pressure Zone 1 into another zone identified by the code “AREA02”. The station’s flow is defined differently before and after 2014-01-01.

The tool enables return data for multiple parameters at once for a user-requested time-span using a user-defined time step. All the required raw data are fetched at once from the different data sources, and all requested parameters values, like water use for a certain area, are calculated in the user defined time steps. Raw data are averaged, interpolated and extrapolated as needed in a deterministic and user-defined manner. Typically, for example, hourly averages for data stored in a minute long interval are retrieved.

The tool used in this work to retrieve initial water tower levels for optimization, and historical water consumption data for pressure zones in order to facilitate demand forecasting.

3.7 Optimization problem formulation

The aim of the optimization process is to minimize the costs of the water production and supply by choosing appropriate time-dependent flow and pressure settings for all the stations, and ultimately the frequency settings for all pumps in the network, while ensuring a sufficient quality of service (QoS), so that pressures are satisfactory, water source yields are not exceeded and water tower levels and capacities stay within the constraints.

Mathematically, the optimization can be described as the minimization of the objective function $f(\bar{x})$ subject to constraints $g_i(\bar{x})$:

$$\begin{array}{ll} \min\limits_{\bar{x}\in X} & f(\bar{x}) \\ \text{subject to} & g_i(\bar{x}) \leq 0, \quad i = 1, \ldots, m \end{array}, \tag{3.15}$$

where $\bar{x}$ is vector containing design variable values chosen from the set of possible values X. Objective function includes the costs associated with the operations:

Listing 4. An example of defining a few stations and an area with water balance calculation

```
<data-sources>
  <data-source name="hdata" native-interval="3600000" ... />
[...]
</data-source>

[...]

<area name="Pressure Zone 1" number="AREA01">
  <parameter name="Demand" expression="IN - OUT">
    <value name="IN" position="Flow" all="in"/>
    <value name="OUT" position="Flow" all="out"/>
  </parameter>
</area>

<watertower name="Water Tower 1" number="TOWER01" out="AREA01">
  <parameter name="Level" expression="hdata:wt01_li"/>
  <parameter name="Volume" expression="hdata:wt01_V"/>
  <parameter name="Flow" expression="(hdata:wt01_V-
      hdata:PREV_wt01_V)/(step/3600)"/>
</watertower>

<source name="Source 102" number="STATION102" in="AREA01">
  <parameter name="Flow" expression="hdata:source102_fi"/>
</source>

<pumpingstation name="Station 100" number="STATION100" in="AREA02"
     out="AREA01">
  <parameter name="Flow" expression="if(now < date(2014,1,1),
          hdata:stat100_fi-hdata:stat100_fi2,
          hdata:stat100_fi2-hdata:stat100_fi)" />
</pumpingstation>
```

water production and pump energy costs. Constraints define, for example, the acceptable pressure range.

Water distribution system optimization problems are NP-hard [13] because various aspects of the water supply system exhibit a non-linear behavior. Pressure loss h_L in a pipe is a non-linear function of the flow Q

$$\begin{aligned} h_L(Q) &= f \cdot \frac{L \cdot v^2}{2 \cdot g \cdot d} \quad \Bigg| v = \frac{Q}{A},\ A = \frac{\pi d^2}{4} \\ &= f \cdot \frac{8 \cdot L \cdot Q^2}{g \cdot d^3 \cdot \pi^2} \quad , \end{aligned} \tag{3.16}$$

and the pumping power

$$P = \frac{\rho g Q H}{\eta} \quad , \tag{3.17}$$

where the pump head H depends on the pressure in the network, and thus on the head losses and workings of the other pumps in the network, and the total pumping efficiency η is a non-simple function of the pump working point (Q, H).

While energy costs and constraints cannot be readily expressed analytically, they can be evaluated iteratively using a hydraulic simulator, such as EPANET. The use of the hydraulic model for objective function and constraint evaluation makes it difficult to calculate or estimate partial differentials of the design variables. The lack of derivative functions, and the non-linear nature of the energy and constraint functions make the use of classical optimization methods, such as linear or dynamic programming, complicated, without heavy linearization and approximation.

In order to make the optimization problem simpler and to make it behave better when using meta-heuristic optimization methods, the constraints are included in the objective function as penalty costs. Thus, the objective function becomes

$$f(\bar{x}) = W(\bar{x}) + E(\bar{x}) + P(\bar{x}) \quad , \tag{3.18}$$

where $W(\bar{x})$ is the sum of water production costs, $E(\bar{x})$ is the sum of pumping energy costs and $P(\bar{x})$ is the sum of penalty costs, or constraint violation costs.

The proposed formulation extends the existing research, for example [211], by including raw water extraction, conveyance and treatment pumping and chemical costs in the objective function, and by accurately modeling the pump energy usage. Some chemical costs have been included in earlier research, for example, Broad et al. [50] included chlorination costs.

The following sub-sections describe the system, design variables and their interpretation, and objective function evaluation in more detail. Objective function evaluation is done by the evaluator module using a modified EPANET hydraulic simulator. The evaluator is presented in Chapter 3.11.

3.7.1 Design variables and encoding

Traditional pump scheduling problems use one binary design variable per hour for each pump, that is 24 binary variables per pump for the whole 24 h optimization period [153]. The approach works well if pumps are on-off controlled and minimum allowed pumping time is one hour. This work, however, uses a different approach, inspired by the in-station scheduling presented in Hashemi et al. [114], in order to optimize flow or pressure controlled pumping stations with logic control and variable-speed driven parallel pumping.

The number of design variables is reduced from 24 *per pump or valve* to five *per station*. Every optimizable station has the following design variables: an integer identifying the time pattern and four real valued settings for different times of the day: morning, day, evening and night settings. The optimization problem thus becomes a mixed integer non-linear programming (MINLP) problem in terms of the design variables.

Time pattern is identified by an integer $0 \ldots 529$. The time pattern is a string of 24 characters from the set M, D, E and N, representing the morning, day, evening and night settings, respectively. The active setting is chosen based on the pattern character at the position of the active hour of the day $(0 \ldots 23)$.

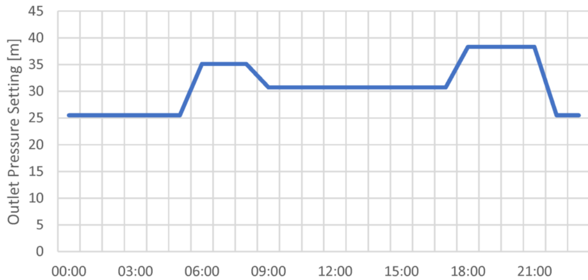
All feasible time patterns were enumerated and stored in a database beforehand. Morning values can be used from 05:00 to 12:00, day values from 07:00 to 21:00, evening values from 14:00 to 04:00 and night values from 20:00 to 10:00. Each setting must be present in every pattern, and the minimum length for the different settings is 2 hours for morning, 5 hours for day, 2 hours for evening, and 4 hours for night. The minimum lengths ensure the setting is not changed too frequently. "NNNNNMMMDDDDDDDDEEEEENNN", "NNNNNNNNMMDDDDDDDDDDDDEE" and "EEENNNNNMMDDDDDDDDDDEEEE" are a few examples of the generated patterns.

If a station is flow controlled, the settings are encoded as $S = Q_{\text{setting}} - Q_{\text{min}} + 1$, where Q_{setting} is the flow setting and Q_{min} is the station's minimum allowed flow. The upper bound for the encoded setting thus becomes $Q_{\text{max}} - Q_{\text{min}} + 1$. If the encoded setting, $0 \leq S < 1$, the station is closed. Pressure controlled stations work analogous to the flow controlled stations, excepts that instead of flow limits Q_{min} and Q_{max}, pressure limits P_{min} and P_{max} are used.

The station level settings are interpreted into the sub-process and sub-operation specific settings using fully modeled stations (see Chapter 3.5) and con-

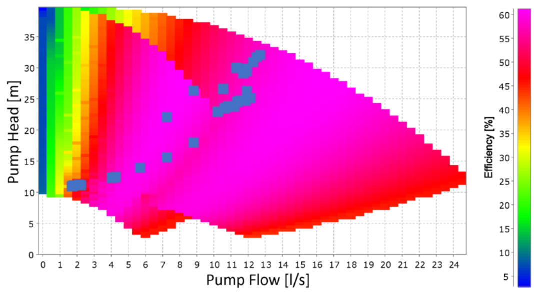
		Pattern	Night	Morning	Day	Evening
1.	Design variables:	110	25.5	35.1	30.7	38.3
2.	Pattern from table:	*NNNNNNMMDDDDDDDDDDEEEENN*				

3.



Pressure booster station outlet pressure setting

4.



Simulated head and flow generated by the pumps (blue squares) and the pre-optimized total efficiency (fill color) and pump frequencies

5. Station electrical energy usage over the simulation period: 75.9 kWh

Figure 18. An example of how design variables are decoded and energy usage calculated based on the simulation results and pump station pre-optimization

trol system model (see Chapter 3.4.2), and finally, to optimal pump specific VSD frequency settings using the parallel pump optimization framework (see Chapter 3.3), which converts the simulated working points (Q, H) for each of the station's pump batteries into optimal pump specific frequencies. The parallel pump pre-optimizer also returns the energy use for the evaluator to calculate the energy consumption, as shown in Figure 18.

3.7.2 Penalties

The solutions are constrained by including a penalty function $P(\bar{x})$ in the objective function (Equation (3.18)). [213]

The method presented here uses the following constraints:

1. there must always be a minimum volume of water in the water towers

2. there should be enough water in each tower to supply the zone for a defined number of hours

3. the water towers should not be over-filled

4. pressure at all points must be at least at the specific minimum level of a pressure zone

5. pressure must not exceed the specific maximum level of a pressure zone at any point.

These constraints ensure a level of resilience and reliability in the distribution system on the one hand and sufficient quality of service on the other hand.

Minimum and maximum pressure limits are very commonly used in the literature (e.g. [39]). Tank end levels are often (e.g. [309, 236, 149, 246]) constrained to be equal to the initial levels. The constraint, however, does not guarantee any reliability, and it assumes that the initial levels in tanks are optimal, and thus restricts the possible solutions. Often, the initial level can be too high or low for what is needed for reliable operations and in terms of optimality, especially in a real-time setting, where the levels are affected by demand variability and pipe bursts, and thus this constraint is not included in the method presented here.

Water tower capacity at the instant t_1 is defined as the number of hours that the total volume in the zone's water towers V_{t_1} suffices for the zone's forecasted water demand F_t. The capacity is $C = t_2 - t_1$, where t_2 is solved from

$$V_F = \int_{t_1}^{t_2} F_t \, dt \quad , \tag{3.19}$$

when for the first time

$$V_F \geq V_{t_1} \quad . \tag{3.20}$$

Together, the minimum volume and capacity provide an intuitive way to define the desired minimum level of reliability.

Additionally, the system must ensure that the daily yield of a water source is not exceeded. While the optimization algorithm ensures that hourly settings are within predefined minimum and maximum, some water sources can have yields that are smaller than $24 \cdot Q_{\max}$ and thus, the yield can be exceeded, and a constraint must be defined.

The penalty parameters, violation inequations, limit and penalty coefficient units, penalty coefficient notations, and which element defines the limit and penalty coefficient are shown in Table 7. The model specific parameters for penalty calculations are defined along with the other model parameters in the system model.

Table 7. Penalty parameter definitions

Parameter	Violation	Limit Unit	Pen. Notation	Penalty Unit	Defined in
Min. tower capacity	$\sum C < C_{\min}$	h	σ_C	$\frac{€}{\mathrm{h\,h}}$	Pressure Zone
Min. tower volume	$\sum V < V_{\min}$	m^3	σ_V	$\frac{€}{\mathrm{m^3\,h}}$	Pressure Zone
Max. tower level	$h > h_{\max}$	m	σ_h	$\frac{€}{\mathrm{m\,h}}$	Water Tower
Min. pressure	$p < p_{\min}$	m	$\sigma_{p_{\min}}$	$\frac{€}{\mathrm{m\,h}}$	Pressure Zone
Max. pressure	$p > p_{\max}$	m	$\sigma_{p_{\max}}$	$\frac{€}{\mathrm{m\,h}}$	Pressure Zone
Max. yield	$\sum Q > Q_{\text{yield}}$	$\frac{\mathrm{m}^3}{\mathrm{d}}$	$\sigma_{Q_{\text{yield}}}$	$\frac{€}{\mathrm{m^3\,d}}$	Source

Penalty cost is calculated by multiplying the magnitude of the violation with the penalty coefficient σ. In the following equations $\max(0, \text{violation})$ notation is used to make penalty zero, when the constraint is not violated. Thus, the penalty function for the time step t becomes

$$P(\bar{x})_t = \sum_{i}^{n_{\text{zones}}} P(\bar{x})_{i,t} + \sum_{i}^{n_{\text{towers}}} T(\bar{x})_{i,t} + \sum_{i}^{n_{\text{source}}} S(\bar{x})_{i,t} \quad , \tag{3.21}$$

where pressure zone specific penalty for zone i

$$\begin{aligned} P(\bar{x})_{i,t} = {} & \sigma_{p_{\min},i} \cdot \sum_{j}^{n_{\text{junctions},i}} \max(0, p_{\text{min, i}} - p(t)_j) \\ & + \sigma_{p_{\max},i} \cdot \sum_{j}^{n_{\text{junctions},i}} \max(0, p(t)_j - p_{\text{max, i}}) \\ & + \sigma_{C,i} \cdot \max\left(0, C_{\text{min, i}} - \left(\sum_{j}^{n_{\text{towers},i}} C(t)_j\right)\right) \\ & \qquad + \sigma_{V,i} \cdot \max\left(0, \left(V_{\text{min, i}} - \sum_{j}^{n_{\text{towers},i}} V(t)_j\right)\right) \quad , \end{aligned} \tag{3.22}$$

tower level penalty for water tower i

$$T(\bar{x})_{i,t} = \sigma_h \cdot \max(0, h_{\min,i} - h(t)_i) \quad , \tag{3.23}$$

and finally, the source yield penalty for source i

$$S(\bar{x})_{i,t} = \sigma_{Q_{\text{yield}}} \cdot \max\left(0, \left(\sum_{u=t-24}^{24} Q(u)_i\right) - Q_{\text{yield},i}\right) \quad . \tag{3.24}$$

The penalty function and the objective functions as a whole are evaluated using the hydraulic simulator by the evaluator module.

3.8 Optimization algorithm

Dynamically dimensioned search (DSS) is a global optimization method first introduced in Tolson and Shoemaker [264]. DSS is a single-solution heuristic algorithm that works with a specified maximum objective function evaluation limit. Besides the maximum number of evaluations, the original algorithm has no other stopping criteria.

As mentioned in the literature review, the performance and computational efficiency of the algorithm are attractive. Being a constant time algorithm, DDS is a good candidate for near real-time optimization. DDS can also be used together with preemption, which further drives down the computational time. The only tunable parameter of the algorithm is r, the relative perturbation size. The default and recommended value $r = 0.2$. [264]

First, the algorithm starts with global search and by iteration, the search becomes more local, by dynamically and probabilistically reducing the number of dimensions searched in the neighborhood. [264]

Candidate solutions are created by perturbing the current solution values in randomly selected dimensions. Perturbation magnitudes are random, and they follow normal distribution with a mean of zero. DDS is a greedy algorithm: the current solution is always the best found so far, and it is never updated with an inferior solution. [264]

Pseudo-code for the algorithm is shown in Algorithm 3.1. The DDS inputs are: r, maximum number of function evaluations m, vectors of lower $\bar{x}_{min}$ and upper bounds $\bar{x}_{max}$ for all n decision variables $\bar{x}$, and initial solution $\bar{x}_0$. First, the objective function is evaluated at the initial solution and the result is stored as current best. Then, the perturbed dimensions are chosen randomly, and they are perturbed according to normal distribution. Finally, the objective function value is evaluated for the new solution. If the new solution is better than the previous best, the new solution replaces the previous.

While the original DDS uses continuous values for the variables, Tolson et al. [265] have introduced a hybrid discrete version of the algorithm (HD-DDS). The algorithm works almost identical to the continuous version, except the decision variable, x_j boundaries are defined to be $(x_j^{min} - 0.5, x_j^{max} + 0.5)$ and rounding to the nearest integer occurs in the perturbing phase. The modified part of the algorithm is shown in Algorithm 3.2. As can be seen from Algorithm 3.2, the algorithm

Algorithm 3.1 Dynamically dimensioned search algorithm [264]

$f_{best} \leftarrow f(\bar{x}_0)$
$\bar{x}_{best} = \bar{x}_0$
for $i \leftarrow 1, m$ **do**
    Randomly select the decision variables that will be perturbed.
    $p \leftarrow 1 - \frac{\ln i}{\ln m}$
    $N \leftarrow \emptyset$
    for $d \leftarrow 1, n$ **do**
        $X \sim U([0,1])$
        if $X \leq p$ **then** $N \leftarrow N \cup \{d\}$
    end for
    if $N = \emptyset$ **then** ▷ Ensure variable change
        $X \sim U([1,n])$
        $N = \{X\}$
    end if
    Construct new solution by perturbing the current best
    $\bar{x} \leftarrow \bar{x}_{best}$
    for $\forall j \in N$ **do**
        $x_j \leftarrow x_j^{best} + r \cdot (x_j^{max} - x_j^{min}) \cdot N([0,1])$
        if $x_j < x_j^{min}$ **then**
            $x_j \leftarrow x_j^{min} + (x_j^{min} - x_j)$
            if $x_j > x_j^{max}$ **then** $x_j \leftarrow x_j^{min}$
        else if $x_j > x_j^{max}$ **then**
            $x_j \leftarrow x_j^{max} - (x_j - x_j^{max})$
            if $x_j < x_j^{min}$ **then** $x_j \leftarrow x_j^{max}$
        end if
    end for
    Evaluate the objective function value for the new solution
    $f \leftarrow f(\bar{x})$
    if $f \leq f_{best}$ **then**
        $f_{best} = f$
        $\bar{x}_{best} = \bar{x}$
    end if
end for

could be easily adapted to work with both discrete and continuous variables at the same time.

Algorithm 3.2 Hybrid discrete dynamically dimensioned search algorithm[265]

[...]
Construct new solution by perturbing the current best
$\bar{x} \leftarrow \bar{x}_{best}$
for $\forall j \in N$ **do**
  $x_j \leftarrow x_j^{best} + r \cdot (x_j^{max} - x_j^{min}) \cdot N([0,1])$
  if $x_j < x_j^{min} - 0.5$ **then**
    $x_j \leftarrow 2x_j^{min} - x_j - 1$
    if $x_j > x_j^{max} + 0.5$ **then** $x_j \leftarrow x_j^{min}$
  else if $x_j > x_j^{max} + 0.5$ **then**
    $x_j \leftarrow 2x_j^{max} - x_j + 1$
    if $x_j < x_j^{min} - 0.5$ **then** $x_j \leftarrow x_j^{max}$
  end if
  $x_j \leftarrow \lfloor x_j + 0.5 \rfloor$ ▷ Round to nearest integer
  if $x_j = x_j^{best}$ **then** $x_j \sim U([x_j^{min}, x_j^{max}])$ ▷ Ensure variable change
end for
[...]

The algorithm implemented in this thesis combines the HD-DDS and DDS variants to allow solving mixed integer non-linear programming problems. After initial testing, the algorithm was changed to temporarily accept results that are worse than the current best in order to broaden the search neighborhood. The resulting algorithm is called Modified Hybrid Discrete Dynamically Dimensioned Search (MHD-DDS).

MHD-DDS first chooses the perturbation algorithm between DDS and HD-DDS based on the design variable type. Every fifth dimension starting with index 0 is the integer coding time pattern index, and all other dimensions are real valued.

MHD-DDS allows for the solution to worsen temporarily. The implementation keeps track of the current result $\bar{x}_{current}$ the last result that was better than the previous result $\bar{x}_{last\,improvement}$ and the best result so far $\bar{x}_{best}$. The algorithm also counts the number of iterations when no improvement to the current result is made n_{equal} and the number of iterations when the result was worsenig since first accepting a worse result n_{worse}. If the current result has not improved during 50 iterations, then solutions the cost of which, $f < 1.15 \cdot f_{\text{best}}$ and $f < 1.05 \cdot f_{\text{last improvement}}$, are accepted as current.

If the current result does not improve for 50 more iterations, the best known solution is restored as the current solution, and the counters are zeroed. Whenever the cost is lowered, the counters are zeroed, and thus the search can continue in the direction as long as the solution cost is less than $f < 1.15 \cdot f_{\text{best}}$ and there is still any progress.

The modified part of the algorithm is shown in Algorithm 3.3.

3.9 Demand forecast

The typically used water demand forecasting methods in the literature include time series analysis based methods, various variations of auto-regressive (AR) and moving-average (MA) models and their generalization, auto-regressive integrated moving average (ARIMA) models. More recently, artificial neural networks (ANN) and other machine learning algorithms, such as support vector machines (SVM), have gained popularity.

As some research has shown (e.g. [28] and [190]), moving-window and pattern-based can perform similarly and even surpass more sophisticated machine learning based algorithms, while being simpler to implement and more general.

The short-term 48 h demand forecasting algorithm used in this work is a simple moving time-window based method. For each zone in the network, hourly median and 10 % and 90 % percentiles for different weekdays for the past $13 \cdot 7 = 91$ days are calculated. The measured zonal demands are fetched using the data access library described in Chapter 3.6.

Because national holidays and their eves affect water demand considerably, an automatic, national holiday calendar was implemented. The current implementation only includes the Finnish holidays, but it is easy to add other nations' calendars.

Holiday calendar calculates the dates for Easter, Christmas, New Year, Independence Day and so on, and the dates of the holidays' eves. Apparent weekday and a demand multiplier can be given for each holiday and its eve. The calendar then returns the multiplier and apparent weekday for any given date. The apparent weekday is the actual weekday for any non-holiday date. In this way, the demand on holidays is not categorized incorrectly as working day usage, and appropriate scaling can be used for forecast evaluation.

The forecast $F_{0\ldots47}$ is then constructed by utilizing the week day specific hourly median values $\operatorname{med} D_t$ and the measured hourly usage data D_t:

$$F_t = \frac{\sum\limits_{i=t-24}^{24} D_i}{\sum\limits_{i=t-24}^{24} \operatorname{med} D_i} \cdot \operatorname{med} D_t \quad . \tag{3.25}$$

Algorithm 3.3 Modified part of (HD-)DDS algorithm as used in MHD-DDS, when temporarily worse results are allowed

[...]
Evaluate the objective function value for the new solution
$f \leftarrow f(\bar{x})$
if $f \leq f_{current}$ **then**
  $n_{worse} \leftarrow 0$
  $n_{equal} \leftarrow 0$
  $f_{current} \leftarrow f$
  $\bar{x}_{current} \leftarrow \bar{x}$
  $f_{last\,improvement} \leftarrow f$
  $\bar{x}_{last\,improvement} \leftarrow \bar{x}$
  if $f \leq f_{best}$ **then**
    $f_{best} = f$
    $\bar{x}_{best} = \bar{x}$
  end if
else if $n_{worse} > 50$ **then**
  Restore the best solution as current solution
  $n_{worse} \leftarrow 0$
  $n_{equal} \leftarrow 0$
  $f_{current} \leftarrow f_{best}$
  $\bar{x}_{current} \leftarrow \bar{x}_{best}$
  $f_{last\,improvement} \leftarrow f_{best}$
  $\bar{x}_{last\,improvement} \leftarrow \bar{x}_{best}$
else if $n_{equal} > 50$ **and** $f < 1.05 \cdot f_{last\,improvement}$ **and** $f < 1.15 \cdot f_{best}$ **then**
  Accept the solution as current solution, tough it is worse
  $n_{worse} \leftarrow n_{worse} + 1$
  $f_{current} \leftarrow f$
  $\bar{x}_{current} \leftarrow \bar{x}$
else if $n_{worse} > 0$ **then**
  $n_{worse} \leftarrow n_{worse} + 1$
else
  $n_{equal} \leftarrow n_{equal} + 1$
end if
[...]

The resulting hourly forecasted demand is restricted to be between the 10th and 90th percentiles, $D_{t,10\%}$ and $D_{t,90\%}$ so that $D_{t,10\%} \leq F_t \leq D_{t,90\%}$, in order to avoid producing overly large or small forecasts because of, for example, measurement errors, missing data or short-term pipe bursts.

3.10 Initial solution generation

Optimizations methods require some initial solution. Random solutions are typically used for population based optimization algorithms, such as genetic algorithm and particle swarm optimizations.

Pasha and Lansey [196] propose the use of warm solutions to speed up the pump scheduling optimization process. The strategies presented in the paper are using linear programming, support vector machine and historical solutions as a warm initial solution in order speed up the optimization and to provide more optimal solutions.

The method presented here uses a historical solution as an initial solution. The latest solution covering the start time of the optimization is chosen. If multiple such solutions exists, the one with the lowest total cost is chosen as initial. If no previous solution is available, then a deterministic initial solution is generated.

The setting pattern is determined on a pressure zone level. The setting pattern is formed by analyzing zone's demand forecast. Analysis starts from $t = 00{:}00$. While the demand D is above the 24 hour average demand D_{avg} and $t < 04{:}00$ the evening setting is used. When for the first time $D < D_{\mathrm{avg}}$ night setting usage begins. Night setting is used while $D < D_{\mathrm{avg}}$ or $t > 09{:}00$. Morning setting is used until $D < 1.1 \cdot D_{\mathrm{avg}}$ or for a maximum of three hours. Day setting is used after morning setting, while $D < 1.1 \cdot D_{\mathrm{avg}}$ or until $t \geq 22{:}00$, whichever occurs first. The rest of the 24 character pattern string is filled either evening setting or night setting, depending on whether $D > D_{\mathrm{avg}}$ or $D \leq D_{\mathrm{avg}}$. The pattern string created in this way is then fuzzily matched to the pregenerated patterns, and the closest match is chosen. The same pattern is used for all the stations pumping into the zone.

All settings, morning, day, evening and night, of every station are set to the midpoint between minimum and maximum flow or pressure allowed for the station. Two-way station settings are set to zero.

3.11 Evaluator

Evaluator is responsible for calculating the value of the objective function for a solution and ensuring that the constraints are met. Evaluator uses a simulator to perform the hydraulic simulations, results of which are used by the evaluator to calculate the objective function value.

The framework allows using any hydraulic simulator with a required set of features with a reasonable effort. In this research the evaluator was built on the modified and extended version of EPANET simulator [226], as described in the previous sections. EPANET is a public-domain hydraulic extended period simulator for pressurized systems. EPANET is based on Todini's formulation of hydraulic equations known as the gradient method [260]. EPANET is the most widely used and researched simulator for water distributions systems, extensively used in the optimization problems for objective function and constraint evaluation.

The EPANET simulator is loaded and prepared only once per optimization run, because loading the model into EPANET and closing the model is slow, and can take up to 13 % [301] of the total simulation time.

A control system model interprets the design variables as settings for valves and pumps batteries and drives the EPANET model dynamically, using `ENsetlinkvalue` calls to set pump battery (see Chapter 3.4.1) and valve settings. This enables to bypass EPANET's built-in controls and rules, and their limitations [150, 167, 256].

Writing simulation results and a report into files is disabled in order to reduce the simulation time by about 6 % [301]. Instead, the evaluator accesses the node results programmatically using `ENgetnodevalue` and link results using `ENgetlinkvalue` as the simulation progresses.

Certain extensions to the vanilla EPANET were made in the course of this research. The C programming language and the programming techniques used in EPANET are somewhat outdated. The code is, however, well-structured and documented. Some issues related to EPANET programming are described in [270] and [224]. Several projects have developed more modern versions of EPANET, such as van Zyl et al. [271], Steffelbauer and Fuchs-Hanusch [249]; however, this thesis used the latest published official EPANET version 2.00.12 from 2008 as a basis for the extensions.

3.11.1 Real-time concerns

In order to speed up the optimization process, the hydraulic simulator was parallelized (see Chapter 3.4.3), so that multiple simulations can be run in parallel using multiple computer threads. The parallelization was done similar to the recent parallelization of EPASWMM [53] – using OpenMP library and more specifically its thread local storage (TLS) functionality, which allows declaring certain global variables as being thread specific and thus enabling calling EPANET from multiple threads. While the DDS algorithm has to proceed sequentially, parallelization still provides benefits, as multiple optimization runs can be performed in parallel.

The simulator was also heavily optimized manually and using the latest options, such as link time optimization and single instruction multiple data (SIMD) extensions provided by the compiler (see Chapter 3.4.4).

Wu and Lee [301] report that in EANET 19.1 % of the total simulation time is spent opening and closing the model and saving the results (see Table 2 on page 45). The evaluator developed in this thesis only opens the model once per thread when the optimization process starts. Afterwards, the same prehypinitialized simulator is used as only the pump settings vary between different solutions.

Neither are the results nor the simulation report generated or saved into a file to save time both on EPANET's and the evaluator's side. Instead, the results are dynamically read using EPANET API. This also avoids the quality simulation run required by the traditional EPANET toolkit usage to generate the final results file.

Preemption is used for avoiding unnecessary objective function evaluations. Preempting means that full objective function evaluations are unwarranted if the candidate is predictably poor or infeasible. Razavi et al. [214] propose a formal method to preemptively stop the ongoing model evaluation, when it becomes apparent that the current individual presents a low-quality solution and it is not going to affect the optimization algorithm.

The method proposed in [214] can be used when the fitness function value is monotonic during the model evaluation. The objective function used in this thesis is monotonic, and thus suitable for preemption.

The other constraint is that preempting the simulation cannot affect the optimization algorithm behavior. Thus, GA and ACO, for example, cannot be used as the final value for objective function because every individual must be known for the algorithm to work, but PSO and DSS can be preempted as the final objective function values are not needed. The modified DDS version developed in this thesis, MHD-DDS, is suitable for preemption.

The evaluator module preempts the simulation when the cumulative cost $f > 1.1 \cdot f_{current}$. The preemption is implemented by throwing a Java exception of type `PreemptionException`, which is caught in the simulator loop. The simulation is then interrupted, and the cumulative cost so far is returned to the optimization algorithm.

The testing done in the case study shows that preemption avoids simulating and evaluating more than 50 % of the time steps, on average, and therefore halving the time required for objective function evaluations.

The MHD-DDS optimization algorithm used in the framework only uses a fixed number of objective function evaluations. This makes the algorithm well suitable for real-time setting, because the approximate run-time for each optimization is known in advance and can be easily tuned according to the available processor speed, model size and time available for the optimization.

3.12 Implementation details

The framework was developed in Java programming language [8], version 8, and using Java EE 7 technologies [131, 9] for web and back-end development. The framework provides both REST API and a graphical web user interface. The optimization framework can also be readily interfaced from SCADA systems.

Java was chosen because it has a strong feature set, a wide array of programming libraries and standards for web development and scientific purposes, and it is widely accepted in the industry and is regarded as the most popular programming language [3, 2].

The following section presents the implementation details of the key parts of the framework.

3.12.1 System model

In order to remain general, the optimization framework allows configuring and storing one or more systems in a database. The system model describes the components forming the system to be optimized and their relations, along with costs and penalties. This section describes the high-level system model used by the framework. The model is shown in Figure 19.

The system model has name and database identifier used for managing the system model. Locations of the base hydraulic model file and configuration file for SCADA access are also stored in the model object.

The basic object for optimization is the station. There are several classes of stations: source, transfer and two-way station. The source produces water into the system, the transfer station conveys water from one pressure zone to another, and the two-way station convesy water between two pressure zones either way. The complete list of stations available for optimization is stored in the model.

All station classes share common properties with each other, such as name, EPANET model link name, minimum and maximum allowed flow and pressure settings, default control type (flow or pressure controlled), and electricity price function. The source stations has additional properties, for example, for production costs, $\frac{€}{\mathrm{m}^3}$, and daily yield, $\frac{\mathrm{m}^3}{\mathrm{d}}$. The two-way station has additional properties for minimum and maximum flow and pressure to the other direction, and for name of the EPANET link that controls the flow to the other direction.

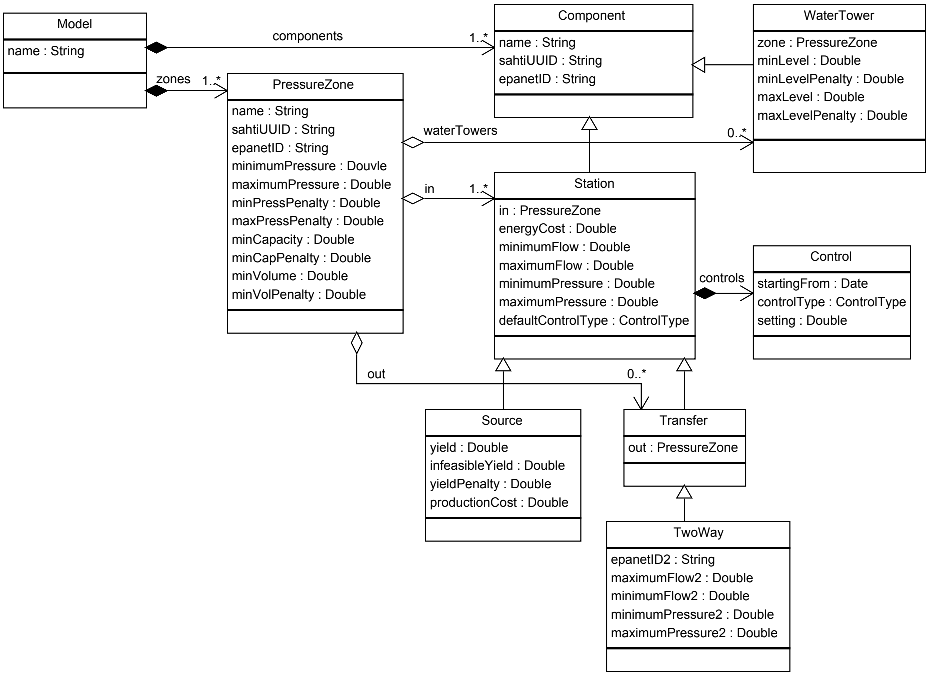


Figure 19. Unified modeling language (UML) class diagram of the optimization framework system model

The transfer and two-way stations link pressure zones together via in and out relationships. The source stations only specify the pressure zone receiving the produced water.

Pressure zones have name, EPANET pattern name, and universally unique identification (UUID), which identifies the position in the SCADA access configuration that contains the zone's water demand. The other properties of the zone include allowed minimum and maximum pressures along with the penalty costs [€/mh], minimum water tower capacity expressed in hours and the related penalty cost [€/hh], and minimum and maximum water tower volume [%], and the related penalty costs [€/mh].

Stations that are related to a zone are available via `inStations` and `outStations` relations. A zone also has zero or more water towers. Water tower components have UUID for SCADA access (initial water tower level) and EPANET component name.

Besides the system components, the model also includes a possibility to control any station manually or change station's control method (flow or pressure) at any point of time. Likewise, water towers can be marked to be disabled. This functionality enables the optimizer to optimize correctly, even when some stations or water towers are unavailable or working at partial or fixed capacity.

The model, its components and all parameters can be changed in the web interface or directly in the database.

3.12.2 Optimization process implementation

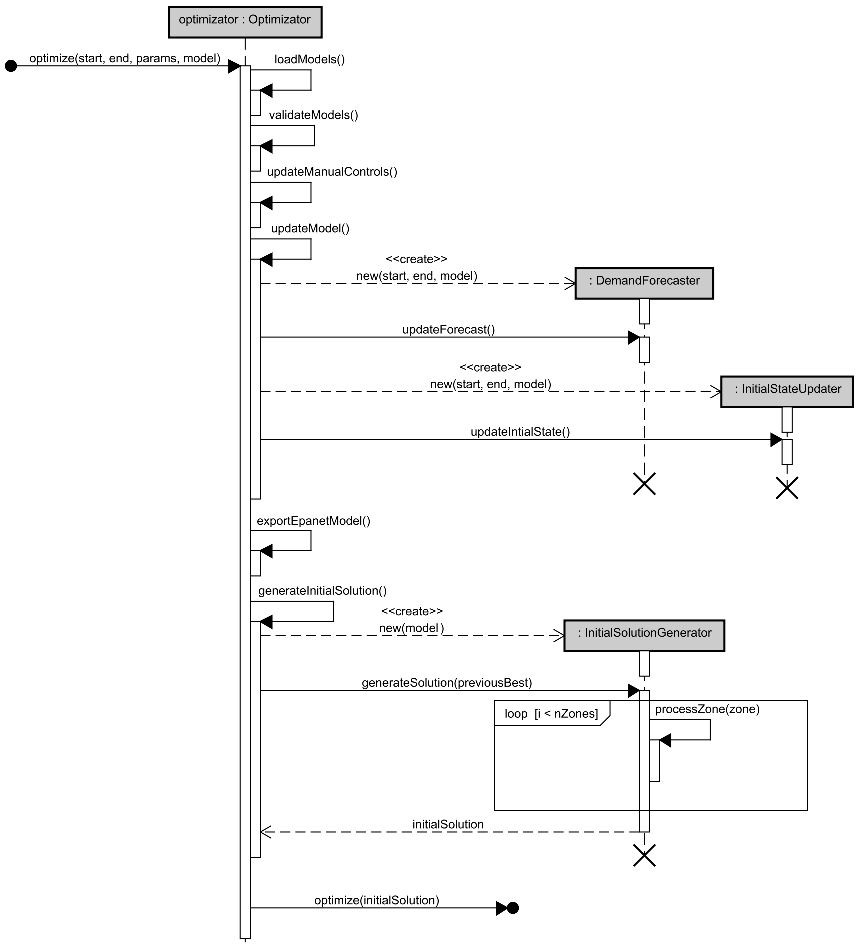


Figure 20. Sequence diagram of the optimization preparation process

When the optimization process is first initiated, either by a user, an external program or a scheduled task, an `Optimizator` class instance is created and its `optimize` method is called. First, the Optimizator performs the optimization preparation as depicted in the unified modeling language (UML) sequence diagram in Figure 20.

First, the EPANET model is loaded, connection is made to the SCADA for data access, and both EPANET model and SCADA data are validated against the system model. The hydraulic model is also validated for simulation.

After loading and validating the models, the manual control overrides are fetched from the database and updated into the model. `UpdateModel` method creates the demand forecasts for all the pressure zones and updates the demands in the hydraulic model. The method also sets the initial levels of the water towers to match the measured levels read from SCADA.

When the updated model is ready, it is exported into EPANET inp file, which is loaded into the modified EPANET simulator for objective function and constraints evaluation.

Finally, an initial solution is generated. The system model, the exported EPANET file, and initial solution are passed on and used in the optimization process itself.

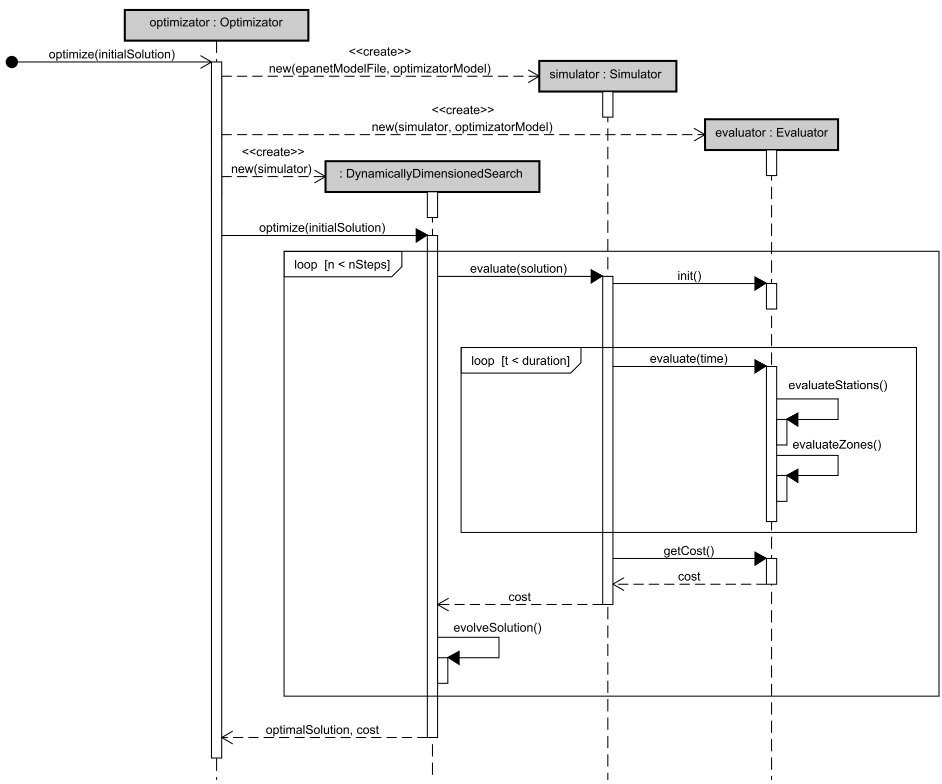


Figure 21. Sequence diagram of the optimization process

The optimization process is shown in the sequence diagram in Figure 21. Simulator instance is created and an evaluator is created and attached to the simulator, and finally, the MHD-DDS implementation `DynamicallyDimensionedSearch` instance is created, and its `optimize` method is called with the initial solution as an argument.

The search algorithm evolves the solution passing solution candidates to the simulator for evaluation. The simulator returns the objective function value, the total cost, to the optimizer, which continues the process, until the number of iterations is exhausted, or the process is interrupted by the user. The final best solution and its cost information is returned to the caller and stored in the database.

3.12.3 Java interface for EPANET

The optimization framework presented here is written in the Java programming language. It was thus necessary to be able to use EPANET from Java. Because hydraulic simulation is the performance bottleneck in the meta-heuristic optimization, it was decided against porting EPANET to Java or using existing Java versions of EPANET. Instead, the optimized version (see Chapter 3.4.4) of the C language version EPANET was used by adding a Java interface for it.

A Java Native Interface [147] (JNI) module was written, that provides almost 1:1 mapping of the EPANET Toolkit [225] application programming interface (API). Instead of returning error codes, the JNI function calls throw an exception of type `gov.epa.EpanetException`, and instead of using pass by reference variables to return values, the function calls directly return the result to the caller. Thus, the API is simpler to use and more modern: there is no need for reference variables and error code checking.

The JNI module forms a basis for the simulator module used by the objective function value evaluator presented in the next section.

3.12.4 Evaluation of the objective function value

The evaluation of the objective function is split into two classes in the implementation of the framework. Simulator class is responsible for driving the EPANET simulation process, calling the model specific control system model, controlling the optimizable stations according to the design variables in the solution to be evaluated, and finally calling the `Evaluator` class to calculate the objective function values.

The simulator is instantiated once per optimization process for each parallel thread. During the initialization, the modified EPANET simulator is prepared calling `ENopen` function with the previously exported EPANET inp file, containing the base model, possible manual control overrides, the latest demand forecast and the initial tank levels as read from SCADA (see Chapters 3.9 and 3.12.2), and the Python interpreter is initialized and control system model is loaded. Saving any report or hydraulic results are explicitly disabled.

When a new solution is required to be evaluated, the search algorithm calls simulator's `simulate` method, with the current solution, $\bar{x}$, as argument. The simulator first calls back the evaluator's `init` method, which zeroes the accumulated costs and penalties. The simulator then starts actual hydraulic simulation by calling the `ENopenH` and `ENinitH` functions in EPANET.

Then, hydraulic simulation is processed time step by time step. First, the current manual controls, if any, or the controls specified by the design variables in the current solution, $\bar{x}$, are applied using EPANET's `ENsetlinkvalue(`*`setting`*`, EN_SETTING)` function. Then the simulation is progressed by calling `ENrunH` and `ENnextH` functions in EPANET.

When the hydraulic results and the next time step, t_2, are known after the `ENnextH` call, the evaluator's `evaluate` method is called with the time step length, $t = t_2 - t_1$ as argument.

Evaluator then calculates the production costs at sources $W(\bar{x})$ by looping over the sources, and inspecting the flows going out of the source using `EN-getlinkvalue(`*`index`*`, EN_FLOW)` function call, converting the flow into cubic metres and multiplying the result by the source's production cost $f_{\text{prod.},i}$:

$$W(\bar{x}) = \sum_{i}^{n_{\text{sources}}} Q_i \cdot t \cdot f_{\text{prod.},i} \quad . \tag{3.26}$$

Energy costs $E(\bar{x})$ are calculated by looping over all the pumps in the system and calling parallel pump optimizer's back-end power function $P_i(Q_i, H_i)$ to calculate the power. The value of Q is determined by calling `ENgetlink-value(`*`index`*`, EN_FLOW)` and H by calling `ENgetlinkvalue(`*`index`*`, EN_UNITHEADLOSS)`. Station specific energy price $f_{\text{e},i}$ is used for calculating the price:

$$E(\bar{x}) = \sum_{i}^{n_{\text{pumps}}} P_i(Q_i, H_i) \cdot t \cdot f_{\text{e},i} \quad . \tag{3.27}$$

Finally, the penalties $P(\bar{x})$ are calculated. The water tower penalties and pressure penalties are calculated by inspecting the water tower levels and junction pressures using `ENgetnodevalue(`*`index`*`, EN_PRESSURE)` function calls and then applying the calculation logic described earlier in Chapter 3.7.2. The yield penalties are calculated in a similar manner.

The various costs are summed together and accumulated over time

$$f(\bar{x})_t = \sum_{u=0}^{t} W_u(\bar{x}) + E_u(\bar{x}) + P_u(\bar{x}) \quad . \tag{3.28}$$

If the resulting accumulated total cost $f_t > 1.1 \cdot f_{current}$, then the simulation process is preempted by throwing an exception of type `PreemptionException`. Otherwise, the simulation goes on until the end.

Finally, `ENcloseH` is called to clean things up for the current hydraulic simulation run, and the objective function value f is returned to the caller – the search algorithm. The EPANET simulator remains initialized and ready for the next simulation run.

3.12.5 User and application programming interfaces

Both the user and the application programming interfaces were constructed using the same Java EE version 7 technologies [75] as the rest of the optimization framework.

The user interface was developed using Java Server Faces [54] (JSF) library PrimeFaces version 5.3 [64]. The user interface is web based and it works using any modern Internet browser on PCs and tablets.

The user interface allows creating, browsing and modifying system models and their parameters. Optimization process can be initiated using the interface, and previous optimization runs along their results can be examined. The web interface allows users set up the online-optimization process parameters, such as the demand forecast and optimization horizon, and whether and how often the optimization is run automatically by a scheduled task.

The intended operational use-case is that a scheduled task runs the optimization process every few hours for the next 24 hours. The results are stored in the optimization frameworks own database, and the optimal controls — both flow and pressure settings on the station level and frequencies for every pump — are sent to the SCADA system so that an optimal solution is readily available for the operators' use in their preferred system.

A REST API was developed using Java API for RESTful Web Services (JAX-RS) version 2.0 [197]. The REST API provides the same functionality as the user interface to enable external programs to call the framework. The API was extensively used in this thesis to automatically analyze the optimizer performance.

3.13 Conclusions

A general whole cost optimization framework for water supply system operations was developed. The optimization framework finds optimal time dependent settings for all the stations and ultimately optimal frequencies for every single pump in the system. The framework is suitable for near real-time optimization.

The presented framework implements an efficent problem formulation and choice of design variables minimizing the search-space size. Pump efficiency frequency scaling and motor and VSD efficiencies are all modeled as the function of pump rotational speed and used in the pre-optimization of the pump batteries at all stations. A full-scale hydraulic model, including raw water extraction and treatment and driven by a control system model, is used for objective function evaluations in the optimization process. The optimization algorithm used by the framework is MHD-DDS with simulation preemption.

The optimization framework provides a web based interface along with a REST API for calling the framework programmatically. The framework was designed to enable the results to be readily integrated into a SCADA system.

The research and development work has resulted in multiple EPANET extensions that can be used outside the optimization framework regime. The thread-safe version of EPANET makes it possible to efficiently and easily utilize multiple cores for hydraulic analysis, and pump battery component and parallel pump optimization along with the control system model framework open for novel possibilities for water supply system modeling.

4 CASE STUDY – TAMPERE WATER SUPPLY SYSTEM

4.1 Background

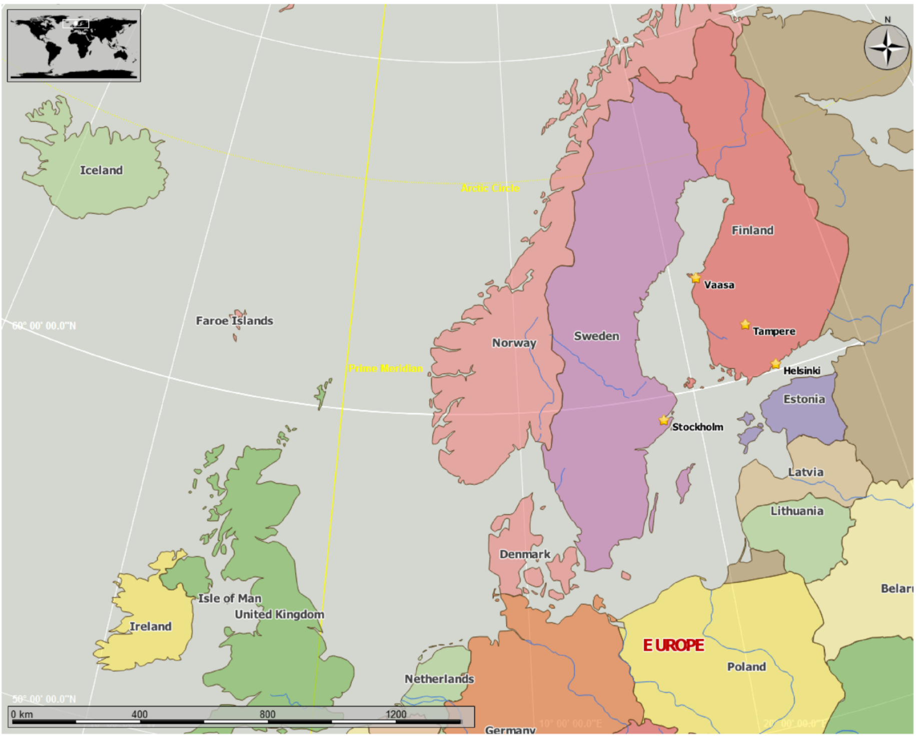


Figure 22. Location of the city of Tampere

TAMPERE Water utility serves the city of Tampere, some 150 km North from Helsinki, and the municipality of Pirkkala, with a total population of 244 182 (as of 2015-12-31). Some water is sold also to other neighboring municipalities: Lempäälä, Kangasala and Nokia.

The network is divided into two parts by a ridge located in the middle of the city. It is possible to pump water over the ridge both ways, but both sides of the network are self-subsisting and thus the pumping is usually avoided. About 78 % of the population lives in the eastern part of the water distribution system (WDS), where both of the surface water treatment plants are located. Average water demand in the eastern part is about 35 500 $\frac{m^3}{d}$ and in the western part about 12 000 $\frac{m^3}{d}$ – totaling 47 500 $\frac{m^3}{d}$.

The network is divided into eight pressure zones – six of which have own elevated water storage. Within larger zones, a few small pressure boosted zones exist, but they are of limited interest with regard to the cost optimization. They are controlled using the default settings, but they are included in the hydraulic simulation and their energy consumption is included in the total.

The network is supplied by eight water sources, three of which are located in the eastern part and the remaining five in the western part. The water sources in the eastern part have a capacity of 105 000 $\frac{m^3}{d}$ and the sources in the western part a capacity of 21 500 $\frac{m^3}{d}$.

In 2014, energy amounted to 9.8 % of the total expenses.

4.2 Water supply system and hydraulic model

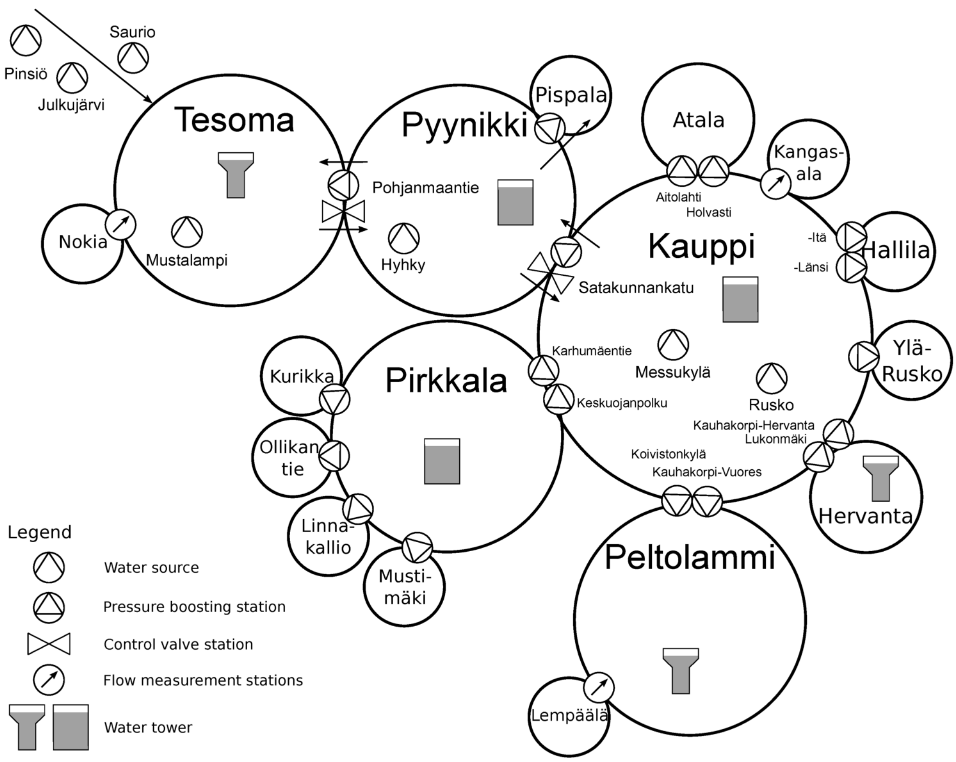


Figure 23. Schematic presentation of the Tampere water supply system

The water supply system is shown schematically in Figure 23. There are 14 pressure zones and three measurement stations delivering water to neighboring towns in the network. The pressure zones are listed in Table 8, along with their respective network lengths, average daily demands, non-revenue water (NRW), inhabitants and water tower volumes. Six zones have their own water tower.

Water usage data are available through the SCADA system for ten of the pressure zones. The four zones, Kurikka, Ollikantie, Linnakallio and Mustimäki, that have no data available for the optimization of their own water usage, are all located within the municipality of Pirkkala, and two of them are using negligible amount of water. The whole Pirkkala is regarded as one pressure zone when the water balance is calculated and updated in the model.

Table 8. Pressure zones in the Tampere water supply system

Zone	Network [km]	Demand [m³/d]	NRW [m³/d]	NRW-% %	Inhabitants	Water Tower [m³]
Kauppi	332.3	26 760	3 328	12.4	114 356	12 000
Tesoma	150.2	7 357	1 067	14.5	36 798	5 500
Pyynikki	60.2	4 837	517	10.7	28 318	3 200
Hervanta	32.9	4 803	256	5.3	23 100	1 000
Atala	56.8	2 629	421	16.0	17 673	
Pirkkala	79.2	3 404	551	16.2	15 779	1 200
Peltolammi	42.9	2 785	243	8.7	8 436	2 300
Kurikka	6.4	290	63	21.8	3 009	
Hallila	6.8	429	33	7.6	2 922	
Pispala	3.2	157	34	21.7	2 489	
Ylä-Rusko	7.8	204	61	29.8	2 478	
Ollikantie	6.0	298	45	15.0	2 471	
Mustimäki	0.9	29	11	36.7	468	
Linnakallio	5.8	39	39	100.0	260	
Lempäälä		1 420				
Kangasala		0				
Nokia		0				
Raw Water	17.8					
Sum	809	55 440	6 667	12.0	258 557	25 200

The network has eight water sources, with a current maximum daily capacity of 126 500 m^3. When the renovation of the Kaupinoja plant was completed in 2017, the daily capacity was raised to 178 500 m^3. Table 9 lists the water sources, their capacities allowed by the environmental permits, raw water sources, raw water multpliers (i.e. the ratio between the extracted raw water and the amount pumped into the network), minimum and maximum hourly flows, and production costs.

Two of the sources, Rusko and Kaupinoja, use lake water as raw water. Together, the surface water sources provide about 70 % of the water used in the system. One of the sources, Saurio, serves nearby city of Ylöjärvi, and it is used in Tampere only when extra capacity is required.

Six of the sources are ground water sources. Their combined capacity is 28 500 m^3. All the ground water sources, except Messukylä, are located on the western side of the network.

Table 9. Water sources in the Tampere water supply system

Station	Capacity [m³/d]	Raw Water	Raw Water Multiplier	Min. Flow [m³/h]	Max. Flow [m³/h]	Prod.cost [€/m³]
Rusko	78 000	Lake	1.1035	500	3000	0.0753
Kauppi	20 000	Lake	1.1035	100	800	0.0700
Messukylä	7000	Ground water	1.0628	20	300	0.0645
Hyhky	3000	Ground water	1.0603	10	130	0.0534
Mustalampi	5000	Ground water	1.0311	10	210	0.0513
Pinsiö	8000	Ground water	1.4165	10	330	0.0185
Julkujärvi	3500	Ground water	1.7500	10	145	0.0133
Saurio	2000	Ground water	1.0000	10	80	

Figure 24 shows the different components of the production costs. The costs are calculated by dividing yearly costs by the volume pumped into the network, and thus, the values are yearly averages. Chemical costs include expenses of major chemicals used in the treatment. Active carbon costs include approximate yearly regeneration and replacement costs. Electricity component includes total electricity used at the source, except for the energy used for pumping the bulk water. Where available, the pumping energy was calculated using power data in SCADA. Otherwise, the simulated energy consumption was used [255].

In total, there are 79 pumps transferring water through the system, excluding the pumps at currently (as of 2016) renovated Kaupinoja surface water plant. All pumps are variable-speed drive controlled. The pumps are located in 25 different remote controllable stations. Table 10 lists all the stations (including sources) and their pumps. The table indicates source and destination pressure zones for the stations, and the pump types.

Most of the stations can only supply water one way. Two of the stations, Pohjanmaantie and Satakunnankatu, can deliver water both ways. Pohjanmaantie can supply Pyynikki with water through a control valve and Tesoma through pumps. Satakunnankatu uses pumps for serving Pyynikki and a control valve is used when water is supplied to Kauppi. Together, the stations enable transfer of water over the ridge through Pyynikki both ways.

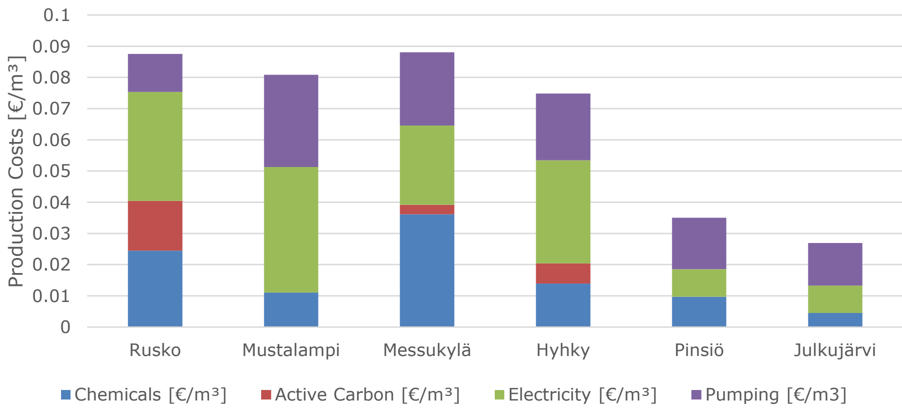


Figure 24. Production cost structure at the water sources in Tampere WSS

Table 10. List of all pumps transferring water in the Tampere water supply system and included in the hydraulic model

Station	From	To	Pump Type	Pump Model
Aitolahti	Kauppi	Atala	Pressure Booster	Z-H12N-1
			Pressure Booster	NK50-160/162
Hallila Itä	Kauppi	Hallila	Pressure Booster	APP-32-65
			Pressure Booster	APP-32-65
Hallila Länsi	Kauppi	Hallila	Pressure Booster	PF-27/315
			Pressure Booster	PF-24/200
Holvasti	Kauppi	Atala	Pressure Booster	MEN-80-65-160
			Pressure Booster	MEN-80-65-160
Hyhky		Pyynikki	Network	KSB UPA250-41/3C
			Network	KSB UPA250-41/3C
			Raw Water	KSB UPA150S-65/3
			Raw Water	KSB UPA150S-65/3
			Raw Water	TVS8_2-1_VV_L6W552D
Julkujärvi		Tesoma	Well to Network	PN83-3
			Well to Network	PN83-3
			Well to Network	PN83-3
Karhumäentie	Kauppi	Pyynikki	Pressure Booster	LP100-125/137
			Pressure Booster	LP100-125/137
Kauhakorpi-Hervanta	Kauppi	Hervanta	Pressure Booster	MEN-100-80-200L
			Pressure Booster	MEN-100-80-200L

Continued on next page

Table 10 – continued from previous page

Station	From	To	Pump Type	Pump Model
			Pressure Booster	MEN-125-100-200L
Kauhakorpi-Vuores	Kauppi	Peltolammi	Pressure Booster	MEN-100-80-160
			Pressure Booster	MEN-125-100-200L
Keskuojanpolku	Kauppi	Pirkkala	Pressure Booster	LP100-125/137
			Pressure Booster	LP100-125/137
Killo	Pirkkala	Linnavuori	Pressure Booster	CR32-4
			Pressure Booster	CR32-4
Koivistonkylä	Kauppi	Peltolammi	Pressure Booster	NK65-160/173
			Pressure Booster	NK65-160/173
Kurikka	Pirkkala	Kurikka	Pressure Booster	CR16-40
			Pressure Booster	CR20-4
Lukonmäki	Kauppi	Hervanta	Pressure Booster	NK80-200/222
			Pressure Booster	PPL12
Messukylä		Kauppi	Network	QN83-7
			Network	QN83-7a
			Raw Water	PN104-3
			Raw Water	SP160-2A
			Raw Water	SP160-2A
Metsäkylä	Tesoma	Tesoma	From Tank to Network	APP-32-125
			From Tank to Network	APP-32-125
			From Tank to Network	APP-44-150
Mustalampi		Tesoma	Network (Epilä)	ELL10-2
			Network (Epilä)	ELL10-2
			Network (Epilä)	ELL10-2
			Network (Tesoma)	ELL10-3
			Network (Tesoma)	ELL10-3
			Network (Tesoma)	ELL10-3
			Raw Water	PN82-2
			Raw Water	QN65-4
			Raw Water	PN84-2A
			Raw Water	PN84-2A
Mustimäki	Pirkkala	Mustimäki	Pressure Booster	CR4-80
			Pressure Booster	CR4-80
Ollikantie	Pirkkala	Ollikantie	Pressure Booster	CR30-30
			Pressure Booster	CR30-30

Continued on next page

Table 10 – continued from previous page

Station	From	To	Pump Type	Pump Model
Pinsiö		Tesoma	Well to Network	Q-82-3
			Well to Network	Q-82-4
			Well to Network	Q-82-3
Pohjanmaantie	Tesoma	Pyynikki	Pressure Booster	Z-K15R-380
			Pressure Booster	Z-K15R-380
Rusko		Kauppi	Network (Kauppi)	Z-V35T-450
			Network (Kauppi)	Z-V35T-450
			Network (Ylä-Rusko)	APP-33-100
			Network (Ylä-Rusko)	APP-33-100
			Lift inside the Process	AFP-2006
			Lift inside the Process	AFP-2006
			Lift inside the Process	AFP-2006
			Lift inside the Process	XP201G CB2
			Lift inside the Process	XP201G CB3
			Lift inside the Process	XP201G CB4
			Raw Water	2PLP-30
			Raw Water	2PLP-30
			Raw Water	2PLP-30
			Raw Water	2PLP-30
Satakunnankatu	Kauppi	Pyynikki	Pressure Booster	Z-K15R-380
			Pressure Booster	Z-K15R-350
Ylä-Pispala	Pyynikki	Pispala	Pressure Booster	KCF-5-140
			Pressure Booster	KFF-8-145

Currently, the operators operate the water supply system manually by changing the flow and pressure settings at different stations. There are only a few established rules on how the system should be operated, and thus the behavior and costs depend on the operator making the changes. The basic principle, however, is to try to avoid transferring water over the ridge, and keeping the flow settings as constant as possible, and thus utilizing the water tower volume to level the changes in the water demand.

The full-scale network model was built using the modified EPANET hydraulic modeling software, and the model contains 5443 nodes and 6457 links. The length of the modeled network is 809 km with an average inner pipe diameter of 185 mm. There are 21 368 water users in the model. The ground elevations vary from 78 m to 170 m above the sea level. The network model is shown in Figure 25.

Figure 25. Network model used in the optimization process

The model includes all the pipes in the network, raw water extraction and transfer, along the hydraulic behavior of the water treatment processes at all sources, and every pump in the system. All pumps are modeled using the pump battery component (see Chapter 3.4.1), and the internal operation of the pump batteries is pre-optimized using the parallel pumping optimization tool (see Chapter 3.3).

The water treatment processes and raw water extraction is controlled by a control system model (see Chapter 3.4.2). The flow pumped into the network Q is multiplied by the source specific raw water coefficient (see Table 9), resulting in the raw water flow $Q_{\mathrm{raw}} \geq Q$. The control system model controls how Q_{raw} is pumped through the process and raw water extraction. In this way, the full energy costs can be calculated.

The original model used a pressure-dependent leakage model utilizing EPANET's nodal emitters[257]. The global emitter coefficient was 1.0. Each zone's calculated non-revenue water is assigned to nodes proportionally to the connected pipe length, pipe diameter and average pressure by iteratively calibrating the emitter coefficients.

However, analyzing the leakage flow from the zonal demands and calibration of emitter coefficients in a real-time setting was deemed a problem that could be deferred for a later project. For the online operational optimization, it was decided to replace the pressure-dependent emitters by fixed nodal demands calculated based on the total zonal leakage and proportionally the connected pipe length, pipe diameter and node's average pressure. The leakage is scaled with the rest of the zonal water usage when the demand forecast is applied to the model.

4.3 Optimization problem formulation

The goal is to optimize the total production and energy costs associated with the Tampere Water utility water supply system online. Chapter 3.7 describes the general problem formulation in detail. The case specific details are presented in this chapter.

The small pressure booster stations, for which no SCADA data are available, were excluded from the set of stations to be optimized. The exclusion set includes Mustimäki, Ollikantie, Kurikka, Killo, and Pispala. The final list pressure booster stations included in the optimization, along with their minimum and maximum flows and pressure parameters used in the optimization are listed in Table 11.

It is worth noting that even though five pressure booster stations were excluded from the optimization set, their energy consumption is still included when calculating the objective function.

Table 11. Pressure booster station parameters

Station	Flow		Pressure		Flow – direction 2	
	Min. $[\frac{m^3}{h}]$	Max. $[\frac{m^3}{h}]$	Min.. [m]	Max. [m]	Min. $[\frac{m^3}{h}]$	Max. $[\frac{m^3}{h}]$
Kauhakorpi-Hervanta	30	500	70	85		
Lukonmäki	30	500	75	90		
Hallila-Itä	5	126	44	80		
Hallila-Länsi	1	72	42	80		
Aitolahti	10	210	50	80		
Holvasti	10	210	48	80		
Koivistonkylä	20	280	65	90		
Kauhakorpi-Vuores	50	350	45	75		
Keskuojanpolku	10	150	58	70		
Karhumäentie	10	200	52	65		
Satakunnankatu	50	540			10	420
Pohjanmaantie	10	420			50	540

The optimization also includes all sources listed earlier in Table 9, except Saurio, which is reserved strictly for emergencies. Mustalampi source includes two pumping stations that pump into the same pressure zone but via different valve separated routes. The pumping stations, Mustalampi–Tesoma Mustalampi–Epilä, have separate pumps and can be controlled separately. The stations' respective minimum and maximum flows are 10 $\frac{m^3}{h}$ to 280 $\frac{m^3}{h}$ and 10 $\frac{m^3}{h}$ to 210 $\frac{m^3}{h}$.

In total, there are eight sources and twelve pressure booster stations, two of which can pump both ways. The final list of design variables includes the setting pattern and morning, day, evening and night settings for the 20 stations. Hallila-Itä, Hallila-Länsi, Aitolahti and Holvasti are pressure controlled by default, and all other stations are flow controlled. The control type can be changed, and the settings can be manually overridden using the web interface.

Using the traditional formulation for pump scheduling, there would be $79 \cdot 24 = 1896$ design variables. The formulation used in this thesis, together with the control system model, reduces the number of design variables to $20 \cdot 5 = 100$, which is 5.3 % of the classical amount, while still providing optimal frequency for every single pump in the system. Even the pumping stations excluded from the system level optimization are still optimized locally using the parallel pumping optimization.

Pressure zone penalty parameters are listed in Table 12. The penalty costs are the same for all zones: 10 $\frac{€}{\text{h}\,\text{h}}$ for capacity penalty, 10 $\frac{€}{\text{m}^3\,\text{h}}$ for volume penalty, 0.1 $\frac{€}{\text{m}\,\text{h}}$ for minimum pressure penalty and 0.01 $\frac{€}{\text{m}\,\text{h}}$ for maximum pressure penalty.

Table 12. Pressure zone penalty parameters

Pressure Zone	$C_{\mathbf{min}}$ [h]	$V_{\mathbf{min}}$ [m^3]	$p_{\mathbf{min}}$ [m]	$p_{\mathbf{max}}$ [m]
Kauppi	4	2 400	25	70
Tesoma	4	1 100	25	80
Pyynikki	4	640	25	65
Hervanta	1	200	25	65
Atala			25	75
Pirkkala	7	240	25	70
Peltolammi	7	460	25	70
Hallila			25	65
Ylä-Rusko			25	65
Lempäälä			25	65
Kangasala			25	65
Nokia			25	90

The maximum allowed water level for all water towers was fixed to 95 % of the overflow level, and the penalty of exceeding that level was set to 200 $\frac{€}{\%\,\text{h}}$, to make it costly to overfill the towers.

Rusko and Messukylä have a reasonable yield, that is below the maximum allowed. The yield limits are 50 000 and 5000, respectively, which is about 70 % of the full capacity in both cases. These limits were stored in the database, with the penalty cost 0.2 $\frac{€}{m^3}$.

The price for electricity was assumed to be 0.085 $\frac{€}{kWh}$. There are no different tariffs in use.

4.4 Baseline costs

In order to analyze the optimization framework performance, the historical costs of the system were first analyzed. For this purpose, the two-week period between 2nd November and 15th November 2015 was chosen. During the period, water demand was close to the typical and there were no major incidents in the network.

The cost calculation was done day-by-day, by first preparing the model to match the measured situation, and then simulating and evaluating the costs using the same exact cost and penalty parameters, and the code as the optimizer uses. The optimizer programming API provides a function to calculate historical costs automatically.

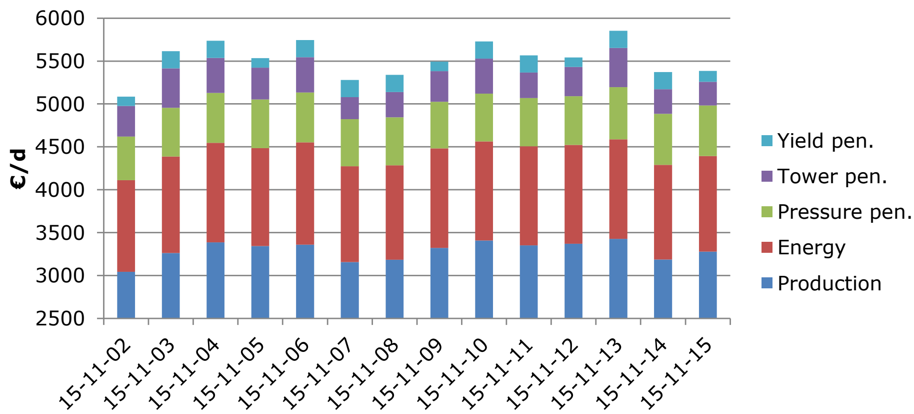


Figure 26. Historical costs and cost components for 2nd–15th November 2015

The resulting costs and cost components are shown graphically in Figure 26 and in tabular format in Table 13. The figure groups tower capacity and volume penalties into one, and volume penalty is not shown in the table, because the average value was below 1 $\frac{€}{d}$ and the maximum was 2 $\frac{€}{d}$. During the period, total costs are 5086–5519 € and real costs, i.e. the sum of production and energy costs is 4113–4589 €. Average values are respectively 5519 € and 4428 €. The penalties account for about one fifth of the total cost. Pressure penalty, including both minimum and maximum pressure penalties, is consistently close to the average 566 €. Tower capacity penalty varies more, average being 356 €. Yield capacity penalty

Table 13. Historical costs and cost components for 2nd–15th November 2015

	Cost [€]				Penalties [€]		
Date	Total	Real	Prod.	Energy	Pressure	Capacity	Yield
2015-11-02	5 086	4 113	3 044	1 069	507	356	110
2015-11-03	5 614	4 388	3 261	1 126	567	460	199
2015-11-04	5 736	4 546	3 388	1 158	581	409	200
2015-11-05	5 533	4 486	3 344	1 141	566	370	110
2015-11-06	5 745	4 553	3 360	1 192	580	413	200
2015-11-07	5 280	4 274	3 157	1 116	549	258	199
2015-11-08	5 340	4 285	3 184	1 101	559	296	199
2015-11-09	5 493	4 481	3 321	1 161	544	357	110
2015-11-10	5 727	4 564	3 410	1 154	556	407	200
2015-11-11	5 566	4 507	3 353	1 154	561	299	199
2015-11-12	5 542	4 523	3 371	1 152	566	343	110
2015-11-13	5 852	4 589	3 427	1 162	607	454	200
2015-11-14	5 372	4 290	3 187	1 103	594	287	199
2015-11-15	5 386	4 394	3 277	1 116	587	277	129
Min	5 086	4 113	3 044	1 069	507	258	110
Max	5 852	4 589	3 427	1 192	607	460	200
Average	5 519	4 428	3 292	1 136	566	356	169

averages at 169 €. Sum total cost for the two-week period is 77 272 € and sum real cost is 61 991

4.5 Optimization results

The savings potential offered by the optimizer and its computational performance was analyzed by comparing both cold start performance on 2nd November 2015 to the historical values, and long-term sustained performance by performing ten different optimization runs for the whole 2nd to 15th November 2015 period in twelve hour intervals.

The calculations were performed on Dell Precision T7610 workstation with two six-core Intel Xeon E5-2620 v2 @ 2.10 GHz processors with hyper threading, 32 GB memory, and 500 GB solid state drive hard-disk. The operating system was 64-bit Windows 7 Enterprise, the Java runtime was 64-bit and version number 1.8.0u66. The optimizer software was run inside GlassFish 4.1.1 application server. The final hardware will be similar, except the processors will be Intel Xeon E5-2620 v3 @ 2.40 GHz, and thus more performant.

Penalty parameters were initially estimated by analyzing the hydraulic model. Later they were tuned based on the historical performance and early optimization results. Currently, the penalties are on average one fifth of the total cost.

4.5.1 Cold start performance

Initial optimizator parameter tuning was performed on 2nd November 2015. The relative perturbation size r, penalty parameters, the number of iterations, aggressiveness of preemption, initial solution algorithm and upward trending were all considered. Monday, 2nd November 2015 was optimized 100 times using each different setting combination, and the results were analyzed.

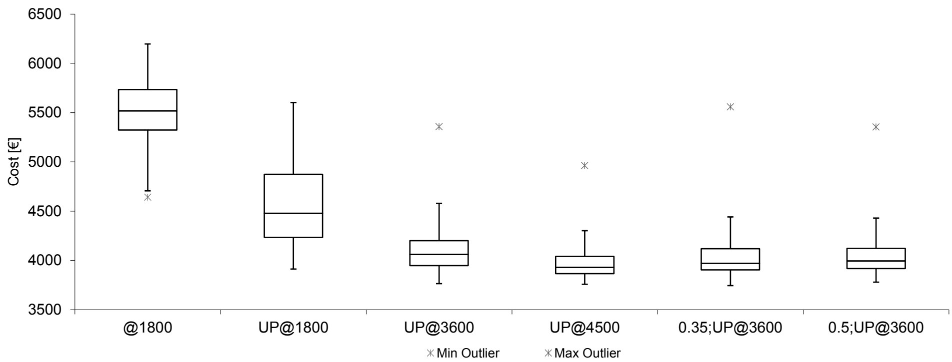


Figure 27. Box plot of the total cost results for 100 optimization runs with different parameters for 2nd November 2015

The total cost results of key parameter combinations are shown graphically as box plots in Figure 27 and real cost results in Figure 28. The same results are shown in tabular format in Table 14. Combination names in the figures are as follows: the number after @ sign signifies the number of iterations, 1800, 3600 or 4500, word UP before the @ sign signifies that the runs were allowed to accet temporarily worse results, and finally the number before @ sign, 0.35 or 0.50, signifies the relative perturbation value, r. If the value is missing, the default $r = 0.20$ is used. The analyzed combinations respectively are: 1800 iterations, 1800, 3600 and 4500 iterations with worse results allowed, and 3600 iterations with worse results allowed with $r = 0.35$ and $r = 0.50$.

The savings on the 1st quartile compared to the baseline performance on 2nd November 2015 are shown in Table 15. The baseline total cost is 5086 € and the baseline real cost is 4113 €. Comparison is done on the 1st quartile, because the

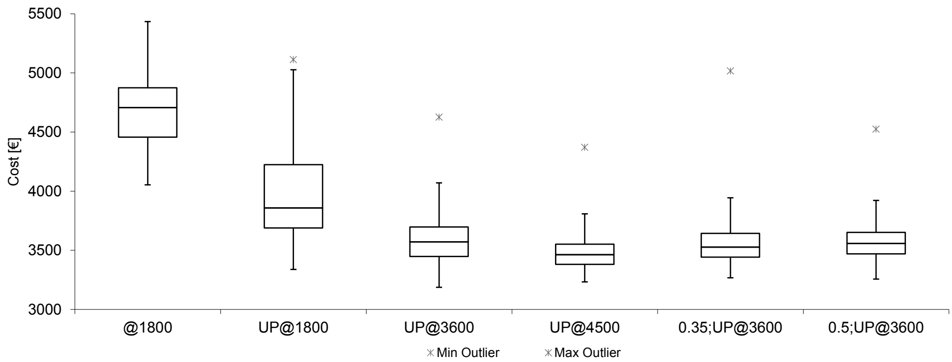


Figure 28. Box plot of the real cost results for 100 optimization runs with different parameters for 2nd November 2015

Table 14. Total and real cost results of the optimization runs with different parameters. The results are shown as 1st quartile, median and 3rd quartile for the population of 100 optimizations for 2nd November 2015.

	Total Cost			Real Cost		
Parameters	**1st Quart.**	**Median**	**3rd Quart.**	**1st Quart.**	**Median**	**3rd Quart.**
@1800	5 328	5 517	5 734	4 455	4 707	4 876
UP@1800	4 232	4 478	4 882	3 685	3 859	4 239
UP@3600	3 947	4 061	4 203	3 447	3 571	3 704
UP@4500	3 866	3 929	4 043	3 374	3 463	3 556
0.35;UP@3600	3 903	3 970	4 120	3 442	3 527	3 645
0.50;UP@3600	3 917	3 994	4 133	3 464	3 557	3 652

online optimization does ten parallel optimization runs, and the best of the ten results is chosen as the optimum. Following binary distribution, there is a 94.4 % chance, that at least one of the ten runs is within the first quartile.

The calculation performance numbers are shown in Table 16. CPU time required by the optimization is approximately linearly proportional to the number of iterations when other parameters are kept the same. Preempting saves typically about 55 % of time step simulations, and the savings increase slightly as the number of iterations rises. Rising of the r parameter makes preemption slightly less efficient and likewise increases the CPU time required. This is likely because the solutions vary more, and the greater proportion of the solution candidates are close to the current best solution.

Table 15. Total cost and real cost savings using different parameters. The 1st quartile results are compared with the baseline efficiency.

	Savings [€]		Savings [%]	
Parameters	**Total**	**Real**	**Total**	**Real**
@1800	-242	-342	-4.7 %	-8.3 %
UP@1800	854	428	16.8 %	10.4 %
UP@3600	1 139	666	22.4 %	16.2 %
UP@4500	1 220	739	24.0 %	18.0 %
0.35;UP@3600	1 183	671	23.3 %	16.3 %
0.50;UP@3600	1169	649	23.0 %	15.8 %

While it is not possible to optimize the system hourly, it is more than feasible to perform the optimization once every three to six hours, which would still provide sufficiently rapid response to the changes happening in the system.

Table 16. Average computational time required for an optimization run and the percentage of time steps saved by preemption using different parameters

Parameters	Mean CPU Time [h]	Preemptions
@1800	0.8	46 %
UP@1800	1.3	55 %
UP@3600	2.5	58 %
UP@4500	3.0	59 %
0.35;UP@3600	2.7	56 %
0.50;UP@3600	2.7	53 %

It was found out that the optimizer gives significantly better results, when the search is allowed to temporarily accept results worse than the current best. Aggressive preemption using the cost estimation had a negative impact on the cost savings. Using 3600 iterations yielded much better results than 1800 iterations, while the computation time was still below 2.5 h. By increasing the number of iterations to 4500 further improves the results. Preemption saves about 50 % to 60 % of hydraulic simulator time steps compared with the full 24 h evaluation, and thus reduces the optimization time to half.

The best results and overall performance was given by enabling the upward trend and disabling aggressive estimation. Using simple initial solution generation and 3600 iterations, gives 1st quartile result 3947 € for total and 3447 € for real costs, and median result 4061 € for total and 3447 € for real costs. Compared with the historical values, 5086 € and 4113 €, the 1st quartile savings are 1139 €

or 22 % and 666 € or 16 %. The yearly savings potential in the real costs would be more than 200 000 €. Using 4500 iterations makes the costs still 80 € lower and bumps the savings percentages to 24 % and 18 %. Besides the lower real costs, the quality of service is better than the manual solution. In particular the minimum water tower capacities are higher in the optimized solutions.

Increasing the value of the relative perturbation parameter r from the default and recommended value of 0.20 to 0.35, or further to 0.50, gives only slightly better results.

It is worth noting that the optimizer returned always better results than the baseline when temporary worsening of the results was allowed and the number of iterations was at least 3600.

4.5.2 Sustained performance

Sustained performance of the optimization framework was analyzed by performing ten different optimizations for 2nd to 15th November 2015 period using twelve-hour intervals for the optimization runs. Based on the results of optimizing a single day, the relative perturbation parameter r value was set to 0.35, the number of iterations to 4500, and the results were allowed to temporarily worsen.

The results are shown in Table 17 and graphically in Figure 29. Total baseline cost for the two-week period was 77 272 € (total cost) and 61 991 € (real cost). Optimized costs are 57 410 € and 49 780 €, which are 25.70 % and 19.70 % lower than the baseline. The CPU time required for the optimization was on average 2.0 h.

The optimizer gave consistently better results compared with the historical baseline performance.

4.6 Online operative application

As of October 2017, the optimization framework is being installed to a server in the Tampere Water utility's office network. Optimization process is started automatically once every six hours by a scheduled task.

Ten optimization runs are performed parallel to each other at once. The Best solution of the ten is chosen. Doing multiple optimizations and choosing the best ensures high probability of obtaining the best possible result.

Because the chance for an optimization result being in the 1st quartile (among the best 25 %) is 25 %, $X \sim B(10, 0.25)$, the probability for at least one of ten optimization runs being in 1st quartile is 94.4 % and being in 2nd quartile is 99.9 %.

Table 17. Results of the sustained performance runs

		Total Cost [€]			Real Cost [€]		
Start	**CPU [h]**	**1st Quart.**	**Median**	**3rd Quart.**	**1st Quart.**	**Median**	**3rd Quart.**
2015-11-02 00:00	1.9	3 908	3 921	3 956	3 398	3 454	3 538
2015-11-02 12:00	2.1	3 931	4 043	4 183	3 302	3 444	3 600
2015-11-03 00:00	2.1	4 441	4 545	4 673	3 923	4 028	4 093
2015-11-03 12:00	2.1	4 088	4 153	4 201	3 551	3 576	3 655
2015-11-04 00:00	2.1	4 283	4 398	4 882	3 765	3 892	4 314
2015-11-04 12:00	2.2	4 056	4 191	4 372	3 530	3 651	3 807
2015-11-05 00:00	2.2	4 441	4 544	5 274	3 895	4 009	4 526
2015-11-05 12:00	2.1	3 903	4 091	4 140	3 364	3 502	3 577
2015-11-06 00:00	2.2	4 430	4 513	4 690	3 947	4 069	4 218
2015-11-06 12:00	1.9	3 740	3 825	3 947	3 253	3 324	3 376
2015-11-07 00:00	2.0	4 061	4 178	4 298	3 583	3 699	3 907
2015-11-07 12:00	2.1	3 426	3 657	3 881	2 922	3 112	3 249
2015-11-08 00:00	2.3	4 073	4 180	4 353	3 561	3 682	3 810
2015-11-08 12:00	2.3	3 974	4 151	4 280	3 298	3 545	3 694
2015-11-09 00:00	1.7	4 396	4 474	4 571	3 746	3 861	4 094
2015-11-09 12:00	2.2	3 872	4 091	4 210	3 292	3 467	3 521
2015-11-10 00:00	1.9	4 677	4 829	5 120	4 138	4 307	4 484
2015-11-10 12:00	1.8	3 993	4 052	4 376	3 392	3 558	3 730
2015-11-11 00:00	2.0	4 646	4 834	4 935	4 072	4 293	4 393
2015-11-11 12:00	2.0	4 162	4 215	4 344	3 594	3 674	3 767
2015-11-12 00:00	2.0	4 454	4 570	5 326	3 923	4 060	4 629
2015-11-12 12:00	2.0	3 925	4 053	4 167	3 363	3 454	3 569
2015-11-13 00:00	2.0	4 599	4 838	5 543	4 079	4 419	4 721
2015-11-13 12:00	1.0	3 650	4 011	4 263	3 093	3 361	3 697
2015-11-14 00:00	2.0	4 070	4 276	4 474	3 622	3 780	3 957
2015-11-14 12:00	1.9	3 510	3 660	3 932	2 969	3 090	3 399
2015-11-15 00:00	2.0	4 222	4 285	4 334	3 669	3 788	3 910
2015-11-15 12:00	1.8	3 888	3 939	4 027	3 317	3 406	3 456
Average	2.0	4100.7	4232.8	4455.4	3555.7	3696.7	3881.9
Sum	56.4	57 410	59 259	62 376	49 780	51 754	54 346

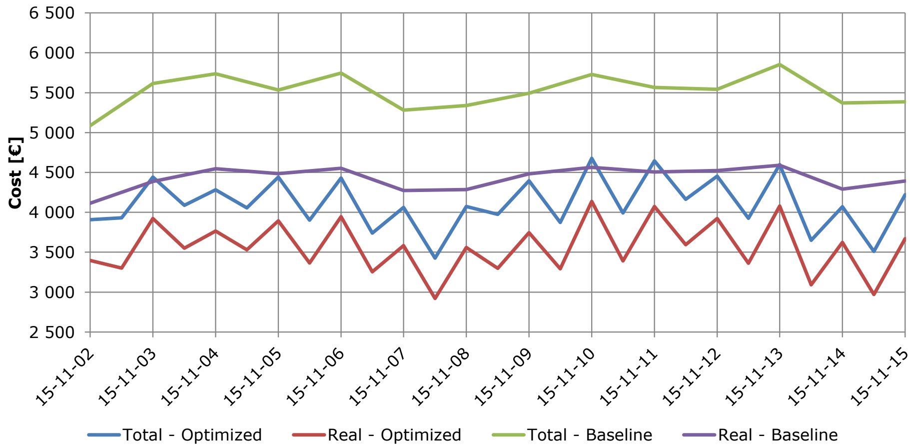


Figure 29. Sustained performance of the optimizer 2nd–15th November 2015 compared with baseline costs

Doing ten optimizations practically ensures that the obtained solution is always better than the median.

The actual historical costs are stored in the utility's open database. Likewise, the optimal costs calculated by the optimizer will be saved in the same database, so that it is possible to analyze, how well the optimizer performs and to better assess the savings potential.

4.7 Conclusions

The Tampere Water utility's network including the city of Tampere and the municipality of Pirkkala was modeled fully. The model includes every pump, pump motor and variable-speed drive in the system. Raw water extraction and treatment processes were also modeled, along the logic controlling them.

The obtained optimization results show that the optimization framework presented in the thesis works as intended. The savings potential varies from 7645 € to 12 211 € for the analyzed two-week period, which results in yearly real cost savings potential of 214 000 € to 342 000 € (12.3 % to 19.7 %). Even cold start performance can result in 18 % or 739 $\frac{€}{\mathrm{d}}$ real cost savings.

The optimization time, 2.0 h on average, is reasonable for near real-time use. The implementation proved to be robust, and integration into the SCADA system lowers the usage barrier and provides the operators with a familiar user interface.

5 CONCLUSIONS AND DISCUSSION

5.1 Summary of work

The aim of this thesis research was to develop a general framework for near real-time whole-cost optimization for operation of a water supply system (WSS) containing elevated storage, variable-speed driven pumps and multiple water sources.

The major goal for the framework was to use a hydraulic model as accurate as possible, and take into account every pumping that happens in the system and each aspect that affects the pumping efficiency and energy consumption, including raw water extraction and conveyance, which were usually left out in the earlier research. The focus was not only on the cost of energy, but also on water production costs.

The goal was achieved, as such a system was developed and successfully tested on a real, large-scale network in a case-study. The case-study shows that it is possible to use a non-simplified full-scale hydraulic model and include raw water extraction, conveyance and treatment in the near real-time operational optimization.

The developed optimization framework makes a 24-hour demand estimate for each demand measurement area and pressure zone, finds optimal flow and pressure settings for every station, and finally, optimal pump frequencies for all pumps in the system over the 24-hour optimization horizon, so that energy and water production costs are minimized while ensuring good quality of service. The optimization happens near real-time.

The optimization time is kept reasonable by a novel optimization problem formulation, considerably reducing the search-space, and by using two-step optimization, i.e. first calculating the global optimum for all possible working points on pump battery level offline, and then focusing the real-time optimization on the optimal flows from the stations using the pre-computed optimal pump combinations and frequencies for each battery.

The global optimization is performed using the novel meta-heuristic optimization algorithm MHD-DDS and utilizing the enhanced EPANET simulator for the objective function and constraint evaluation.

The case studies presented in Publication I and Publication II show that optimizing just the internal functioning of pumping stations can result in savings of 5 % to 8 %. Further, the full-scale high-level optimization results in 20 % savings in energy and 29 % savings in total costs, including the penalties in the case study presented in this dissertation (see Chapter 4) and in Publication V and Publication VI. The total yearly savings potential in the production and distribution costs in the case-study was 342 000 €. The computational time required by the optimization was on average 2.0 h.

5.2 Conclusions

The questions this thesis research set to answer were:

1. What components affecting energy use are typically missing from the operational optimization problem solutions, and how can they be included?
2. Can near real-time optimization be performed using a full-scale, all-pipe network model, including raw water extraction, conveyance and treatment, and an accurate pump energy model?

Both questions were properly addressed, and the case-study shows that the proposed methodology works and yields better results than the operators are currently able to achieve. However, it is still required to compare the methodology to other optimization methods and use benchmark networks.

Full-scale, non-simplified models can be used even in a near real-time setting, when the proposed problem formulation and enhancements are applied to the EPANET simulator.

The proposed optimization framework is the most complete presented in the literature, including all energy usage components of pump train, and the water production and distribution. Thus, the work can provide a baseline against which to compare other more computationally efficient methodologies.

The completion of this study raises new research questions: how significant it is to use a method as accurate as proposed here, and how different the optimization results would be using a simpler model.

5.3 Thesis contributions

The dissertation provides contributions in three main areas: optimal operation of variable-speed driven pumps, water supply system modeling, and global optimization of a water supply system as a whole.

Contributions to optimal operation of variable speed driven pumps are:

1. General and accurate model for pump drive train energy usage and efficiency

2. Method to globally pre-optimize operation of the pumps working in parallel in a pump battery using exhaustive search. The method supports both variable speed and fixed-speed pumps, and pumps with non-unique and non-analytic pump characteristic curves,

Contributions to water supply system modeling and the EPANET simulator are:

1. Novel EPANET component for modeling of flow or pressure controlled batteries of variable-speed driven pumps
2. General and accurate model for pump drive train energy usage and efficiency
3. Novel method to model arbitrary complex WSS control strategies in EPANET
4. Novel method for enabling parallel processing using EPANET without breaking the existing API
5. Various optimizations in EPANET to reduce time required for simulations.

Contributions to operational optimization of water supply system are:

1. Novel operational optimization problem formulation resulting in substantially reduced search space size
2. Inclusion of raw water extraction, conveyance and treatment in the system-wide optimization problem, and utilizing a non-simplified full-pipe system model in the optimization
3. Novel heuristic MINLP optimization algorithm, MHD-DDS, developed by fusing and modifying slightly continuous DDS and discrete HD-DDS
4. A complete, generally usable framework for optimizing water supply and distribution systems with variable-speed pumping either online or offline.

5.4 Future work recommendations

The performance of the MHD-DDS algorithm and problem formulation presented here should be compared with other problem formulations and commonly used algorithms, such as particle swarm optimization, genetic algorithm, and ant colony optimization. The case studies should include Tampere and other water supply systems, including the commonly used benchmark networks.

This would show how much of the performance is due to the problem formulation, especially due to the reduced search space size, and how much is due to the MHD-DDS algorithm being efficient.

The results obtained from the optimization still require more careful analysis, as to how the optimization utilizes different sources and stations and where exactly the savings come from. The actual implementation phase is also still an ongoing project and separate from this research work. Part of the implementation phase budget will be dedicated on more through results analysis and on solution implementability analysis.

More careful tuning of the penalty and optimization algorithm parameters could result in greater savings and better performance. Optimizing the penalty parameters would be an interesting line of research. Some studies focus on the automatic optimization of the penalty parameters (meta-optimization), such as [299], [12], [74], and [210]. These methods could be included in the framework quite easily.

Optimization against the spot energy pricing or a daily price pattern instead of a fixed energy price would be another interesting line of research. The use of dynamic electricity pricing and price forecasts [292] could result in a new level of energy cost savings. The inclusion of spot energy prices or some energy price forecast module in the framework would be quite straightforward, and could result in further savings in energy costs.

New developments in hydraulic simulation (see Chapter 2.5) can reduce the simulation time remarkably, and thus allow for more extensive optimization and better approximation of the global optimum in the same time-frame. The optimization framework allows using other hydraulic simulators than EPANET. Utilizing another simulator could yield better performance with regard to the computational time required, and could allow one to remediate some remaining limitations of EPANET.

There is only little research done on the actual effect of inaccuracies in the demand forecasts on the results in the operational optimization. Doing a comparison of select demand forecasting methods and the resulting optimization performance would be useful. It is also likely that the demand forecast model currently in use could be further improved.

One major improvement that could be investigated is the pressure-dependant leakage modeling. The presented case study includes fixed leakage component calculated for each measurement area and distributed to nodes proportional to pipe lengths, pipe diameters and average pressures. Online analysis of background leakage from demand measurements is not, however, a simple task, and especially calibrating the nodal emitter coefficients online without human intervention can be challenging, but could possibly yield greater savings.

Another interesting track for future research would be to analyze the realized long-term savings and how well in general the operators use controls suggested by the optimization frameworks. The optimization framework could also be easily altered to directly control the water supply system, after being used for long enough to prove its reliability. Direct model predictive optimal control would be especially beneficial if the operators do not follow the suggested optimal controls.

The future research could also explore how including a posteriori multi-objective optimization, for example, presenting costs versus some resilience index, affects the choices done by the operators. This would enable the operators make conscious choices between the costs and resilience and would not need them to blindly trust the optimization results.

Water supply resilience and reliability [231, 306, 248] often conflict with energy optimization goals [295]. The methodology presented here include reliability aspects such as pressure and water tower capacity and volume penalties, and because besides lowering real costs, the optimization also lowers the penalties it can be assumed, that the resilience is on a better level after optimization. Nonetheless, it would be interesting to compare various resilience and hydraulic performance metrics [22] along with energy balance [255, 237] of the optimized solutions with the historical performance of the system. Some reliability metrics could be included as a multi-objective goal besides the costs and penalties, and the choice of the exact solution to be implemented would be left to the operator.

Many of the developments, especially those done in the EPANET simulator and pump energy modeling, have much wider applicability besides the optimization discussed in this dissertation. Hopefully, much of the developments could be contributed to Open Water Analytics open source project to benefit and to be improved by other researchers and EPANET users.

This thesis research forms a good basis for future research and commercialization. The whole development forms an integrated system that can be easily tested and extended upon. The large-scale case-study shows that the chosen methodology holds potential. Several lines of research available can be pursued to further improve the methodology.

BIBLIOGRAPHY

[1] Epanet Java. URL http://baseform.com/np4/epanetTool.html.

[2] Pypl popularity of programming language – september 2016. URL http://pypl.github.io/PYPL.html.

[3] Tiobe index for september 2016. URL http://www.tiobe.com/tiobe-index.

[4] *Linker and Libraries Guide.* Redwood Shores, California, 2010.

[5] *Determining Electric Motor Load and Efficiency.* U.S. Department of Energy, 2010.

[6] *Guidelines for Drinking-water Quality.* World Health Organization, Geneva, 4th edition, 2011. ISBN 978 92 4 154815 1. URL http://apps.who.int/iris/bitstream/10665/44584/1/9789241548151{_}eng.pdf.

[7] *SE/SL Pumps.* Bjerringbro, Denmark, 2015.

[8] Java Platform Standard Edition 8 Documentation, 2016. URL http://docs.oracle.com/javase/8/docs/.

[9] Java Platform, Enterprise Edition (Java EE) 7 Release 7, 2016. URL http://docs.oracle.com/javaee/7/index.html.

[10] Hossam AbdelMeguid and Bogumil Ulanicki. Feedback Rules for Operation of Pumps in a Water Supply System Considering Electricity Tariffs. In *Water Distribution Systems Analysis 2010*, pages 1188–1205, Tucson, Arizona, 2010. American Society of Civil Engineers. ISBN 978-0-7844-1203-9. doi: 10.1061/41203(425)108. URL http://ascelibrary.org/doi/abs/10.1061/41203(425)108.

[11] Edo Abraham and Ivan Stoianov. Sparse Null Space Algorithms for Hydraulic Analysis of Large-Scale Water Supply Networks. *Journal of Hydraulic Engineering*, 142(3):1–13, 2016. doi: 10.1061/(ASCE)HY.1943-7900.0001089.

[12] M. H. Afshar and M. A. Mariño. A parameter-free self-adapting boundary genetic search for pipe network optimization. *Computational Optimization and Applications*, 37(1):83–102, 2007. ISSN 09266003. doi: 10.1007/s10589-007-9016-1.

[13] Enrique Alba, Gabriel Luque, and Sergio Nesmachnow. Parallel metaheuristics: Recent advances and new trends. *International Transactions in Operational Research*, 20(1):1–48, 2013. ISSN 09696016. doi: 10.1111/j.1475-3995.2012.00862.x.

[14] T De Almeida, Paula Fonseca, and Paolo Bertoldi. Energy-efficient motor systems in the industrial and in the services sectors in the European Union : characterisation , potentials , barriers and policies. 28:673–690, 2003. doi: 10.1016/S0360-5442(02)00160-3.

[15] M Alonso, Fernando Alvarruiz, David Guerrero, Vicente Herna, Pedro A Ruiz, Antonio M Vidal, Fernando Martı, and Juan Vercher. Parallel computing in water network analysis and leakage minimization. *Journal of Water Resources Planning and Management*, 126(4):251–260, 2000.

[16] E. Alperovits and U. Shamir. Design of optimal water distribution systems. *Water Resources Research*, 13(6):885–900, 1977. ISSN 19447973. doi: 10.1029/WR013i006p00885.

[17] Abdüsselam Altunkaynak, Mehmet Özger, and Mehmet Çakmakci. Water Consumption Prediction of Istanbul City by Using Fuzzy Logic Approach. *Water Resources Management*, 19(5):641–654, October 2005. ISSN 0920-4741. doi: 10.1007/s11269-005-7371-1. URL `http://link.springer.com/10.1007/s11269-005-7371-1`.

[18] Stefano Alvisi, Marco Franchini, and Alberto Marinelli. A short-term, pattern-based model for water-demand forecasting. *Journal of Hydroinformatics*, 9(1):39–50, January 2007. ISSN 14647141. doi: 10.2166/hydro.2006.016. URL `http://www.iwaponline.com/jh/009/jh0090039.htm`.

[19] F Martinez Alzamora, B Ulanicki, and E Salomons. Fast and Practical Method for Model Reduction of Large-Scale Water-Distribution Networks. *Journal of Water Resources Planning and Management*, 140:444–456, 2014. doi: 10.1061/(ASCE)WR.1943-5452.0000333.

[20] Aijun An, Ning Shan, Christine W. Chan, Nick Cercone, and Wojciech Ziarko. Discovering Rules for Water Demand Prediction : An Enhanced

Rough-set Approach. *Engineering Applications of Artificial Intelligence*, 9 (6):645–653, 1996.

[21] I. Montalvo Arango, J.S. Izquierdo, E.O.G. Campbell, and R. Pérez-García. Cloud-based Decision Making in Water Distribution Systems. *Procedia Engineering*, 89:488–494, 2014. ISSN 18777058. doi: 10.1016/j.proeng.2014.11.241. URL http://www.sciencedirect.com/science/article/pii/S187770581402356X.

[22] Sandro Artina, Andrea Bolognesi, Cristiana Bragalli, and Angela Marchi. Comparison among the best solutions of the optimal design of water distribution networks obtained with different algorithms. In *Conference on Computing and Control for the Water Industry*, 2011.

[23] R. Baños, C. Gil, J. Reca, and F.G. Montoya. A memetic algorithm applied to the design of water distribution networks. *Applied Soft Computing*, 10 (1):261–266, 2010. ISSN 15684946. doi: 10.1016/j.asoc.2009.07.010.

[24] Nadia Babaei, Massoud Tabesh, and Sara Nazif. Optimum reliable operation of water distribution networks by minimising energy cost and chlorine dosage. *Water SA*, 41(1):149–156, 2015. doi: http://dx.doi.org/10.4314/wsa.v41i1.18.

[25] K. S. Jinesh Babu and D. P. Vijayalakshmi. Self-Adaptive PSO-GA Hybrid Model for Combinatorial Water Distribution Network Design. *Journal of Pipeline Systems Engineering and Practice*, (February):35, 2012. ISSN 1949-1190. doi: 10.1061/(ASCE)PS.1949-1204.0000113.

[26] A.M. Bagirov, A.F. Barton, H. Mala-Jetmarova, A. Al Nuaimat, S.T. Ahmed, N. Sultanova, and J. Yearwood. An algorithm for minimization of pumping costs in water distribution systems using a novel approach to pump scheduling. *Mathematical and Computer Modelling*, 57(3-4):873–886, February 2013. ISSN 08957177. doi: 10.1016/j.mcm.2012.09.015. URL http://linkinghub.elsevier.com/retrieve/pii/S0895717712002543.

[27] Yun Bai, Pu Wang, Chuan Li, Jingjing Xie, and Yin Wang. Dynamic Forecast of Daily Urban Water Consumption Using a Variable-Structure Support Vector Regression Model. *Journal of Water Resources Planning and Management*, 141, 2015. doi: 10.1061/(ASCE)WR.1943-5452.0000457.

[28] M. Bakker, J.H.G. Vreeburg, K.M. van Schagen, and L.C. Rietveld. A fully adaptive forecasting model for short-term drinking water demand. *Environmental Modelling & Software*, 48:141–151, October 2013. ISSN 13648152.

doi: 10.1016/j.envsoft.2013.06.012. URL http://linkinghub.elsevier.com/retrieve/pii/S1364815213001576.

[29] Benjamín Barán, Christian Von Lücken, and Aldo Sotelo. Multi-objective pump scheduling optimisation using evolutionary strategies. *Advances in Engineering Software*, 36(1):39–47, 2005. ISSN 09659978. doi: 10.1016/j.advengsoft.2004.03.012.

[30] M. Behandish. Generalized GPU-based Artificial Neural Network Surrogate Model for Extended Period Hydraulic Simulation. In *World Environmental and Water Resources Congress 2013*, pages 716–731, 2013. ISBN 978-0-7844-1294-7. doi: 10.1061/9780784412947.069. URL http://ascelibrary.org/doi/abs/10.1061/9780784412947.069.

[31] M. Behandish and Z.Y. Wu. Concurrent Pump Scheduling and Storage Level Optimization Using Meta-models and Evolutionary Algorithms. *Procedia Engineering*, 70:103–112, 2014. ISSN 18777058. doi: 10.1016/j.proeng.2014.02.013. URL http://www.sciencedirect.com/science/article/pii/S1877705814000150.

[32] Stuart Bennett. The past of PID controllers. *Annual Reviews in Control*, 25:43–53, 2001.

[33] Michel A. Bernier and Bernard Bourret. Pumping Energy And Variable Frequency Drives. *ASHRAE Journal*, (December):1–4, 1999.

[34] Weiwei Bi, Graeme C Dandy, and M Asce. Optimization of Water Distribution Systems Using Online Retrained Metamodels. *Journal of Water Resources Planning and Management*, 140(10):1–10, 2014. doi: 10.1061/(ASCE)WR.1943-5452.0000419.

[35] Josef Bicik and Zoran Kapelan. Challenges in the implementation of a dss for real-time wds management. In *Conference on Computing and Control for the Water Industry*, 2011.

[36] C Biscos, M Mulholland, M-v Le Lann, C A Buckley, and C J Brouckaert. Optimal operation of water distribution networks by predictive control using MINLP. *Water SA*, 29(4):393–404, 2003.

[37] Fulvio Boano, Marco Scibetta, Luca Ridolfi, and Orazio Giustolisi. Water distribution system modeling and optimization: A case study. *Procedia Engineering*, 119(1):719–724, 2015. ISSN 18777058. doi: 10.1016/j.proeng.2015.

08.925. URL http://dx.doi.org/10.1016/j.proeng.2015.08.925.

[38] Dominic Boccelli. Announcement of an Open Source EPANET Initiative, 2015. URL http://community.wateranalytics.org/t/announcement-of-an-open-source-epanet-initiative/117.

[39] Jessica Bohórquez, Juan Saldarriaga, and Daniel Vallejo. Pumping pattern optimization in order to reduce WDS operation costs. *Procedia Engineering*, 119:1069–1077, 2015. ISSN 1877-7058. doi: 10.1016/j.proeng.2015.08.936. URL http://dx.doi.org/10.1016/j.proeng.2015.08.936.

[40] Andrea Bolognesi, Cristiana Bragalli, Angela Marchi, and Sandro Artina. Genetic Heritage Evolution by Stochastic Transmission in the optimal design of water distribution networks. *Advances in Engineering Software*, 41(5):792–801, 2010. ISSN 09659978. doi: 10.1016/j.advengsoft.2009.12.020. URL http://dx.doi.org/10.1016/j.advengsoft.2009.12.020.

[41] Gratien Bonvin, Sophie Demassey, Claude Le Pape, Nadia Maïzi, Vincent Mazauric, and Alfredo Samperio. A convex mathematical program for pump scheduling in a class of branched water networks. *Applied Energy*, 185:1702–1711, 2017. ISSN 03062619. doi: 10.1016/j.apenergy.2015.12.090. URL http://dx.doi.org/10.1016/j.apenergy.2015.12.090.

[42] A Borzi, E Gerbino, S Bovis, and M Corradini. Genetic Algorithms for water distribution network calibration: a real application. In *Conference on Computing and Control for the Water Industry*, 2005.

[43] A Borzi, P Cutore, and R Gueli. A Methodology for the Evaluation of Optimal Pumping System Operation Through Synthetic Indices. In *Water Distribution Systems Analysis 2008*, pages 532–542, Kruger National Park, South Africa, 2008.

[44] Peter Bounds, Jens Kahler, and Bogumil Ulanicki. Efficient energy management of a large-scale water supply system. *Civil Engineering and Environmental Systems*, 23(3):209–220, 2006. ISSN 1028-6608. doi: 10.1080/10286600600789409.

[45] Cristiana Bragalli, Claudia D'Ambrosio, Jon Lee, Andrea Lodi, and Paolo Toth. On the optimal design of water distribution networks: A practical MINLP approach. *Optimization and Engineering*, 13(2):219–246, 2012. ISSN 13894420. doi: 10.1007/s11081-011-9141-7.

[46] Malcolm Brandt. Energy Efficiency in the Water Industry, a Global Research Project. *Water Practice & Technology*, 6(2):1–2, July 2011. ISSN 1751231X. doi: 10.2166/wpt.2011.028. URL `http://www.iwaponline.com/wpt/006/wpt0060028.htm`.

[47] Bruno M Brentan, Edevar Luvizotto, Manuel Herrera, Joaquin Izquierdo, and Rafael Pérez-García. Hybrid regression models for near real-time urban water demand forecasting. *Journal of Computational and Applied Mathematics*, 2016. ISSN 03770427. doi: 10.1016/j.cam.2016.02.009. URL `http://dx.doi.org/10.1016/j.cam.2016.02.009`.

[48] Lehar M. Brion and Larry W. Mays. Methodology for Optimal Operation of Pumping Stations in Water Distribution Systems. *Journal of Hydraulic Engineering*, 117(11):1551–1569, 1991. ISSN 0733-9429. doi: 10.1061/(ASCE)0733-9429(1991)117:11(1551).

[49] D. R. Broad, G. C. Dandy, and H. R. Maier. Water Distribution System Optimization Using Metamodels. *Journal of Water Resources Planning and Management*, 131(3):172–180, 2005. ISSN 0733-9496. doi: 10.1061/(ASCE)0733-9496(2005)131:3(172).

[50] D. R. Broad, H. R. Maier, and G. C. Dandy. Optimal Operation of Complex Water Distribution Systems Using Metamodels. *Journal of Water Resources Planning and Management*, 136(4):433–443, 2010. ISSN 0733-9496. doi: 10.1061/(ASCE)WR.1943-5452.0000052.

[51] S. Bunn. Optimal pump scheduling for East Bay Municipal Utility District, Oakland CA, using the Derceto package. In *Conference on Computing and Control for the Water Industry*, 2005.

[52] S Bunn. Pump scheduling optimization in four us cities: Case studies. In *8th Annual Water Distribution Systems Analysis Symposium*, Cincinnati, Ohio, 2006.

[53] G. Burger, R. Sitzenfrei, M. Kleidorfer, and W. Rauch. Parallel flow routing in SWMM 5. *Environmental Modelling and Software*, 53:27–34, 2014. ISSN 13648152. doi: 10.1016/j.envsoft.2013.11.002.

[54] Ed Burns, editor. *JavaServer Faces Specification Version 2.2*, volume 436. Oracle Corporation, Redwood Shores, California, 2013.

[55] Charles Burt, Xianshu Piao, Franklin Gaudi, Bryan Busch, and NFN Taufik. Electric Motor Efficiency under Variable Frequencies and Loads. Technical report, California Polytechnic State University, San Luis Obispo, 2006.

[56] Charles Burt, Franklin Gaudi, and Xianshu Piao. Variable Frequency Drive (VFD) Considerations for Irrigation. In Albert J. Clemmens and Susan. S. Anderson, editors, *USCID Fourth International Conference*, pages 1049–1062, Sacramento, California, 2007. U.S. Committee on Irrigation and Drainage.

[57] E Cabrera, E Gómez, E Cabrera Jr., J Soriano, and V Espert. Energy Assessment of Pressurized Water Systems. *Journal of Water Resources Planning and Management*, 2014.

[58] Enrique Cabrera, MA Pardo, Ricardo Cobacho, and Enrique Cabrera Jr. Energy audit of water networks. *Journal of Water Resources Planning and Management*, 136(December):669–677, 2010. URL `http://ascelibrary.org/doi/abs/10.1061/(ASCE)WR.1943-5452.0000077`.

[59] Patrick Carribault, Marc Pérache, and Hervé Jourdren. Thread-local storage extension to support thread-based mpi/openmp applications. In *OpenMP in the Petascale Era*, pages 80–93. Springer, 2011.

[60] C Chang and J E van Zyl. Speeding up the stochastic analysis of water distribution systems using a compression heuristic. In *Water Distribution Systems Analysis 2010*, Tucson, Arizona, 2010.

[61] W.P. Cheng, T.C. Yu, and G. Xu. Real-Time Model of a Large-scale Water Distribution System. *Procedia Engineering*, 89:457–466, 2014. ISSN 18777058. doi: 10.1016/j.proeng.2014.11.212. URL `http://linkinghub.elsevier.com/retrieve/pii/S1877705814023273`.

[62] Carla Cherchi, Mohammad Badruzzaman, Joan Oppenheimer, Christopher M Bros, and Joseph G Jacangelo. Energy and water quality management systems for water utility's operations: A review. *Journal of environmental management*, 153C:108–120, feb 2015. ISSN 1095-8630. doi: 10.1016/j.jenvman.2015.01.051. URL `http://www.sciencedirect.com/science/article/pii/S0301479715000675`.

[63] Gunhui Chung, Do Guen Yoo, and Joong Hoon Kim. Optimal Water Quality Sensor Locations in Water Distribution Systems by Network Analysis and

Multi-Objective Genetic Algorithm. *Water Distribution Systems Analysis 2010*, pages 309–316, dec 2011. doi: 10.1061/41203(425)29. URL http://ascelibrary.org/doi/abs/10.1061/41203(425)29.

[64] Çağatay Çivici. *PrimeFaces User Guide 5.3*. PrimeTek Informatics, Ankara, Turkey, 1st edition, 2015. doi: 10.4337/9781782545583.00006.

[65] B. Coelho and A. Andrade-Campos. Efficiency achievement in water supply systems—A review. *Renewable and Sustainable Energy Reviews*, 30:59–84, February 2014. ISSN 13640321. doi: 10.1016/j.rser.2013.09.010. URL http://www.sciencedirect.com/science/article/pii/S1364032113006692.

[66] Congress of the United States of America. Energy Independence and Security Act of 2007, 2007.

[67] Luis Henrique Magalhães Costa, Bruno de Athayde Prata, Helena M. Ramos, and Marco Aurélio Holanda de Castro. A Branch-and-Bound Algorithm for Optimal Pump Scheduling in Water Distribution Networks. *Water Resources Management*, 30(3):1037–1052, 2016. ISSN 0920-4741. doi: 10.1007/s11269-015-1209-2. URL http://link.springer.com/10.1007/s11269-015-1209-2.

[68] Edson Costa Bortoni, Roberto Alves Almeida, and Augusto Nelson Carvalho Viana. Optimization of parallel variable-speed-driven centrifugal pumps operation. *Energy Efficiency*, 1(3):167–173, April 2008. ISSN 1570-646X. doi: 10.1007/s12053-008-9010-1. URL http://link.springer.com/10.1007/s12053-008-9010-1.

[69] Hardy Cross. Analysis of flow in networks of conduits or conductors. 1936.

[70] Maria da Conceição; Cunha and Joaquim Sousa. Water Distribution Network Design Optimization: Simulated Annealing Approach. *Journal of Water Resources Planning and Management*, 125:215–221, 1999.

[71] Maria da Conceição Cunha and Luisa Ribeiro. Tabu search algorithms for water network optimization. *European Journal of Operational Research*, 157(3):746–758, sep 2004. ISSN 03772217. doi: 10.1016/S0377-2217(03)00242-X. URL http://www.sciencedirect.com/science/article/pii/S037722170300242X.

[72] Graeme Dandy, Andrew Wilkins, and Hayley Rohrlach. A Methodology for Comparing Evolutionary Algorithms for Optimising Water Distribution Systems. *Water Distribution Systems Analysis 2010*, pages 786–798, dec

2011. doi: 10.1061/41203(425)73. URL `http://ascelibrary.org/doi/abs/10.1061/41203(425)73`.

[73] Anibal T De Almeida, Fernando JTE Ferreira, and João AC Fong. Standards for efficiency of electric motors. *Industry Applications Magazine, IEEE*, 17 (1):12–19, 2011.

[74] W. A. De Landgraaf, A. E. Eiben, and V. Nannen. Parameter calibration using meta-algorithms. *2007 IEEE Congress on Evolutionary Computation, CEC 2007*, pages 71–78, 2007. doi: 10.1109/CEC.2007.4424456.

[75] Linda DeMichiel and Bill Shannon. *Java Platform, Enterprise Edition (Java EE) Specification, v7*. Oracle Corporation, Redwood Shores, California, 2015.

[76] J Deuerlein, S Elhay, A R Simpson, and M Asce. Fast Graph Matrix Partitioning Algorithm for Solving the Water Distribution System Equations. *Journal of Water Resources Planning and Management*, 142(1):1–11, 2016. doi: 10.1061/(ASCE)WR.1943-5452.

[77] Jochen Deuerlein and Andreas Wolters. Online hydraulic simulation as operative tool for improving water network security. In *Conference on Computing and Control for the Water Industry*, 2011.

[78] Jochen Deuerlein, Angus R. Simpson, and Egbert Gross. The Never Ending Story of Modeling Control-Devices in Hydraulic Systems Analysis. *Water Distribution Systems Analysis 2008*, pages 1–12, March 2009. doi: 10.1061/41024(340)72. URL `http://ascelibrary.org/doi/abs/10.1061/41024(340)72`.

[79] Jochen W Deuerlein. Decomposition Model of a General Water Supply Network Graph. *Journal of Hydraulic Engineering*, 134(6):822–832, 2008.

[80] Kegong Diao, Zhengji Wang, Gregor Burger, Chien-hsun Chen, and Wolfgang Rauch. Software Speedup of water distribution simulation by domain decomposition. *Environmental Modelling and Software*, 52:253–263, 2014. ISSN 1364-8152. doi: 10.1016/j.envsoft.2013.09.025. URL `http://dx.doi.org/10.1016/j.envsoft.2013.09.025`.

[81] F Dipierro, S Khu, D Savic, and L Berardi. Efficient multi-objective optimal design of water distribution networks on a budget of simulations using hybrid algorithms. *Environmental Modelling Software*, 24(2):202–213, 2009. ISSN 13648152. doi: 10.1016/j.envsoft.2008.06.008. URL `http://dx.doi.org/10.1016/j.envsoft.2008.06.008`.

[82] Emmanuel A Donkor, Thomas A Mazzuchi, Refik Soyer, and J Alan Roberson. Urban Water Demand Forecasting : Review of Methods and Models. *Journal of Water Resources Planning and Management*, 140(February):146–159, 2014. doi: 10.1061/(ASCE)WR.1943-5452.

[83] Ulrich Drepper. ELF Handling For Thread-Local Storage. Technical report, RedHat Inc., Raleigh, North Carolina, 2013.

[84] Arne Stolbjerg Drud. CONOPT – A Large-Scale GRG Code. *ORSA Journal on Computing*, 6(2):207–216, 1994. ISSN 1091-9856. doi: 10.1287/ijoc.6.2.207. URL http://joc.journal.informs.org/cgi/doi/10.1287/ijoc.6.2.207.

[85] European Commission. Commission Regulation (EC) No 640/2009 of 22 July 2009 implementing Directive 2005/32/EC of the European Parliament and of the Council with regard to ecodesign requirements for electric motors (Text with EEA relevance), 2009.

[86] European Parliament and Council of the European Union. Directive 2005/32/EC of the European Parliament and of the Council of 6 July 2005 establishing a framework for the setting of ecodesign requirements for energy-using products and amending Council Directive 92/42/EEC and Directives 96/57/EC and 2000/55/EC of the European Parliament and of the Council, 2005.

[87] MM Eusuff and KE Lansey. Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and ...*, (June):210–225, 2003. URL http://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-9496(2003)129:3(210).

[88] F Fallside, B Sc, D Ph, C Eng, and P F Perry. Hierarchical optimisation of a water-supply network. *Institute of Eletrical Engineers*, 122(2):202–208, 1975. ISSN 00189146. doi: 10.1049/iipi.1975.0021.

[89] R Farmani, D A Savic, and G A Walters. Benchmark problems for design and optimisation of water distribution systems. In *Advances in Water Supply Management*, pages 249–256. 2003.

[90] R Farmani, P Ingeduld, D A Savic, G A Walters, Z Svitak, and Jan Berka. Hydraulic water quality and realtime control model of South West Moravian regional water supply system. In *Conference on Computing and Control for the Water Industry*, number 5, 2005.

[91] Raziyeh Farmani and Jonathan A. Wright. Self-Adaptive Fitness Formulation for Constrained Optimization. *IEEE Transactions on Evolutionary Computation*, 7(5):445–455, 2003. ISSN 1089778X. doi: 10.1109/TEVC.2003.817236.

[92] Raziyeh Farmani, Jonathan A Wright, Dragan A Savic, and Godfrey A Walters. Self-Adaptive Fitness Formulation for Evolutionary Constrained Optimization of Water Systems. *Journal of Computing in Civil Engineering*, 19 (2):212–216, 2005. ISSN 08873801. doi: 10.1061/(ASCE)0887-3801(2005)19:2(212).

[93] Farshid Felfelani and Reza Kerachian. Municipal water demand forecasting under peculiar fluctuations in population : a case study of Mashhad , a tourist city. *Hydrological Sciences Journal*, 2016. doi: 10.1080/02626667.2015.1027208.

[94] Michael Foord and Christian Muirhead. *IronPython in action*. Manning Publications Co., 2009. ISBN 978-1933988337.

[95] N. Ganidi and B. Holden. Real time control of water distribution systems using a multi-criteria decision-support tool for optimal water network management - a case study. *Proceedia Engineering*, 89:495–501, 2014.

[96] Zong Woo Geem. Particle-swarm harmony search for water network design. *Engineering Optimization*, 41(4):297–311, 2009. ISSN 0305-215X. doi: 10.1080/03052150802449227.

[97] Zong Woo Geem, Joong Hoon Kim, and G V Loganathan. A new heuristic optimization algorithm: harmony search. *Simulation*, 76(2):60–68, 2001. doi: 10.1177/003754970107600201.

[98] Srbislav Genić, Ivan Arandjelović, Petar Kolendić, Marko Jarić, Nikola Budimir, and Vojislav Genić. A Review of Explicit Approximations of Colebrook ' s Equation. *FME Transactions*, 39(2):67–71, 2011. ISSN 14512092.

[99] A. M. Georgescu, C. I. Cosoiu, S. Perju, S. C. Georgescu, L. Hasegan, and A. Anton. Estimation of the Efficiency for Variable Speed Pumps in EPANET Compared with Experimental Data. *Procedia Engineering*, 89:1404–1411, 2014. ISSN 18777058. doi: 10.1016/j.proeng.2014.11.466. URL `http://www.sciencedirect.com/science/article/pii/S1877705814025818`.

[100] Sanda-Carmen Georgescu, Radu Popa, and Andrei-Mugut Georgescu. Pumping Stations Scheduling for a Water Supply System with Multiple

Tanks. *University "Politehnica" of Bucharest Scientific Bulletin, Series D: Mechanical Engineering*, 72(3):129—-140, 2010.

[101] Bissan Ghaddar, Joe Naoum-Sawaya, Akihiro Kishimoto, Nicole Taheri, and Bradley Eck. A Lagrangian decomposition approach for the pump scheduling problem in water networks. *European Journal of Operational Research*, 241(2):490–501, 2015. ISSN 03772217. doi: 10.1016/j.ejor.2014.08.033. URL http://linkinghub.elsevier.com/retrieve/pii/S0377221714007103.

[102] M. Ghiassi, H. Saidane, and D.K. Zimbra. A dynamic artificial neural network model for forecasting time series events. *International Journal of Forecasting*, 21(2):341–362, April 2005. ISSN 01692070. doi: 10.1016/j.ijforecast.2004.10.008. URL http://linkinghub.elsevier.com/retrieve/pii/S0169207004001116.

[103] Carlo Giacomello, Zoran Kapelan, and Matteo Nicolini. Fast Hybrid Optimisation Method for Effective Pump Scheduling. *Journal of Water Resources Planning and Management*, 193:175–183, 2013. ISSN 0733-9496. doi: 10.1061/(ASCE)WR.1943-5452.0000239.

[104] Orazio Giustolisi, Daniele Laucelli, Luigi Berardi, and Dragan A Savi. Computationally Efficient Modeling Method for Large Water Network Analysis. *Journal of Hydraulic Engineering*, 138(4):313–326, 2012. doi: 10.1061/(ASCE)HY.1943-7900.0000517.

[105] J. F. Gülich. Effect of Reynolds Number and Surface Roughness on the Efficiency of Centrifugal Pumps. *Journal of Fluids Engineering*, 125:670–679, 2003. ISSN 00982202. doi: 10.1115/1.1593711.

[106] Christos Gogos, Panayiotis Alefragis, and Efthymios Housos. Public Enterprise Water Pump Scheduling System. In *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference*, Catania, Italy, 2005.

[107] G Greenaway, R Guanlao, N Bayda, and Q Zhang. Water distribution systems demand forecasting with pattern recognition. In *8th Annual Water Distribution Systems Analysis Symposium*, 2006.

[108] M. Guidolin, P. Burovskiy, Z. Kapelan, and D. a. Savić. CWSNET: An Object-Oriented Toolkit for Water Distribution System Simulations. *Water Distribution Systems Analysis 2010*, pages 1–13, December 2010. doi: 10.

1061/41203(425)2. URL http://ascelibrary.org/doi/abs/10.1061/41203(425)2.

[109] M Guidolin, D Savic, Z Kapelan, and O Giustolisi. High performance hydraulic simulations with epanet on graphics processing units. In *the 9th International Conference on Hydroinformatics, HIC 2010*, 2010.

[110] Michele Guidolin, Dragan A Savic, and Zoran Kapelan. Computational performance analysis and improvment of the demand-driven hydraulic solver for the cwsnet library. In *Conference on Computing and Control for the Water Industry*, 2011.

[111] I. Gupta, J.K. Bassin, A. Gupta, and P. Khanna. Optimization of water distribution system. *Environmental Software*, 8(2):101–113, 1993. ISSN 02669838. doi: 10.1016/0266-9838(93)90020-I.

[112] Indrani Gupta, A Gupta, and P Khanna. Genetic algorithm for optimization of water distribution systems. *Environmental Modelling & Software*, 14(5):437–446, 1999. ISSN 13648152. doi: 10.1016/s1364-8152(98)00089-9. URL http://www.sciencedirect.com/science/article/pii/S1364815298000899.

[113] Mehrdad Hakimi-Asiabar, Seyyed Hassan Ghodsypour, and Reza Kerachian. Deriving operating policies for multi-objective reservoir systems: Application of Self-Learning Genetic Algorithm. *Applied Soft Computing Journal*, 10(4):1151–1163, 2010. ISSN 15684946. doi: 10.1016/j.asoc.2009.08.016.

[114] Seyed Saeed Hashemi, Massoud Tabesh, and Behzad Ataeekia. Ant-colony optimization of pumping schedule to minimize the energy cost using variable-speed pumps in water distribution networks. *Urban Water Journal*, 11(5):335–347, July 2013. ISSN 1573-062X. doi: 10.1080/1573062X.2013.795233. URL http://dx.doi.org/10.1080/1573062X.2013.795233.

[115] Sam Hatchett, Dominic Boccelli, Jim Uber, Terra Haxton, Robert Janke, Amy Kramer, Amy Matracia, and Srinivas Panguluri. How Accurate is a Hydraulic Model? *Water Distribution Systems Analysis 2010*, pages 1379–1389, dec 2011. doi: 10.1061/41203(425)123. URL http://ascelibrary.org/doi/abs/10.1061/41203(425)123.

[116] Sam Hatchett, James Uber, Dominic Boccelli, Terra Haxton, Robert Janke, Amy Kramer, Amy Matracia, and Srinivas Panguluri. Real-time distribution

system modeling: development, application, and insights. In *Conference on Computing and Control for the Water Industry*, 2011.

[117] Manuel Herrera, Luís Torgo, Joaquín Izquierdo, and Rafael Pérez-García. Predictive models for forecasting hourly urban water demand. *Journal of Hydrology*, 387(1-2):141–150, June 2010. ISSN 00221694. doi: 10.1016/j.jhydrol.2010.04.005. URL `http://linkinghub.elsevier.com/retrieve/pii/S0022169410001861`.

[118] Chatree Homwongs, Tep Sastri, and Josepth. W. III Foster. Adaptive Forecasting of Hourly Municipal Water Consumption. *Journal of Water Resources Planning and Management*, 120(6):888–905, 1994.

[119] Robert Hooke and T. A. Jeeves. “Direct Search” Solution of Numerical and Statistical Problems. *Journal of the ACM*, 8(2):212–229, 1961. ISSN 00045411. doi: 10.1145/321062.321069.

[120] Lily a. House-Peters and Heejun Chang. Urban water demand modeling: Review of concepts, methods, and organizing principles. *Water Resources Research*, 47(W05401), may 2011. ISSN 0043-1397. doi: 10.1029/2010WR009624. URL `http://www.agu.org/pubs/crossref/2011/2010WR009624.shtml`.

[121] G Hovstadius. Life-cycle strategy for pumps improves cost structure. *World Pumps*, (413):30–32, 2001. ISSN 02621762 (ISSN). doi: 10.1016/S0262-1762(01)80061-4. URL `http://www.scopus.com/inward/record.url?eid=2-s2.0-0035246763{&}partnerID=40{&}md5=d94078808b01e8f6f55d3a4c95a7c5c0`.

[122] David Ibarra and Josep Arnal. Parallel Programming Techniques Applied to Water Pump Scheduling Problems. *Journal of Water Resources Planning and Management*, 140, 2014. doi: 10.1061/(ASCE)WR.1943-5452.0000439.

[123] IEC60034-31. IEC60034-30:2008, 2008.

[124] IEC60034-31. IEC60034-31:2009, 2009.

[125] P Ingeduld. Online analysis with epanet. In *Water Distribution Systems Analysis 2006*, Cincinnati, Ohio.

[126] ISO9899:2011. ISO/IEC 9899:2011 - Information technology – Programming languages – C, 2011.

[127] Ashu Jain and Lindell E. Ormsbee. Short-Term Water Demand Forecast Modeling Techniques - Conventional Methods Versus AI. *Journal AWWA*, 94(7):64–72, 2002.

[128] Ashu Jain, Ashish Kumar Varshney, and Umesh Chandra Joshi. Short-term water demand forecast modelling at IIT Kanpur using artificial neural networks. *Water Resources Management*, 15(5):299–321, 2001. ISSN 09204741. doi: 10.1023/A:1014415503476.

[129] M. Jamasb, M. Tabesh, and M. Rahimi. Calibration of EPANET Using Genetic Algorithm. *Water Distribution Systems Analysis 2008*, pages 1–9, mar 2009. doi: 10.1061/41024(340)77. URL `http://ascelibrary.org/doi/abs/10.1061/41024(340)77`.

[130] Derek G Jamieson, Uri Shamir, Fernando Martinez, and Marco Franchini. Conceptual design of a generic, real-time, near-optimal control system for water-distribution networks. *Journal of Hydroinformatics*, 9(1):3–14, 2007. doi: 10.2166/hydro.2006.013.

[131] Eric Jendrock, Ricardo Cervera-Navarro, Ian Evans, Kim Haase, and William Markito. *The Java EE 7 Tutorial.* Addison-Wesley Professional, 2014.

[132] P. M. R. Jonkergouw, S.-T. Khu, Z. S. Kapelan, and D. A. Savic. Water Quality Model Calibration under Unknown Demands. *Journal of Water Resources Planning and Management*, 134(4):326–336. ISSN 0733-9496@e1943-5452. URL `http://cedb.asce.org/cgi/WWWdisplay.cgi?0805554`.

[133] L Jourdan, D W Corne, D Savic, and G Walters. LEMMO: Hybridising rule indication and NSGA II for multi-objective water systems design. *Conference on Computing and Control for the Water Industry*, pages 4–9, 2005.

[134] Paul W. Jowitt and George Germanopoulos. Optimal Pump Scheduling in Water-Supply Networks. *Journal of Water Resources Planning and Management*, 118(4):406–422, July 1992. ISSN 0733-9496. doi: 10.1061/(ASCE)0733-9496(1992)118:4(406). URL `http://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-9496(1992)118:4(406)`.

[135] Josh Juneau, Jim Baker, Victor Ng, Leo Soto, and Frank Wierzbicki. *The Definitive Guide to Jython: Python for the Java Platform.* 2010. ISBN 978-1-4302-2528-7.

[136] Donghwi Jung, Doosun Kang, Mingu Kang, and Byungseop Kim. Real-time pump scheduling for water transmission systems: Case study. *KSCE Journal of Civil Engineering*, 2014. ISSN 1226-7988. doi: 10.1007/s12205-014-0195-x. URL `http://link.springer.com/10.1007/s12205-014-0195-x`.

[137] J Kahler and B Ulanicki. DISCUSSION OF NETWORK DESIGN AND SCHEDULING PROBLEMS. In *Water Distribution Systems Analysis 2006*, Cincinnati, Ohio, 2006.

[138] Doosun Kang and Kevin Lansey. Multi-Objective Optimal Design of Dual Water Distribution Network. *Water Distribution Systems Analysis 2010*, pages 810–821, December 2010. doi: 10.1061/41203(425)75. URL `http://ascelibrary.org/doi/abs/10.1061/41203(425)75`.

[139] Durmus Kaya, E. Alptekin Yagmur, K. Suleyman Yigit, Fatma Canka Kilic, a. Salih Eren, and Cenk Celik. Energy efficiency in pumps. *Energy Conversion and Management*, 49(6):1662–1673, 2008. ISSN 01968904. doi: 10.1016/j.enconman.2007.11.010.

[140] O Kiselychnyk, M Bodson, and H Werner. Overview of energy efficient control solutions for water supply systems. *Transactions of Kremenchuk State Polytechnic University*, 3(56):40–46, 2009. URL `http://www.kdu.edu.ua/statti/2009-3-1(56)/40.PDF`.

[141] Joshua D Knowles. ParEGO: A Hybrid Algorithm With On-Line Landscape Approximation for Expensive Multiobjective Optimization Problems. *Ieee Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

[142] M. Koor, A. Vassiljev, and T. Koppel. Optimization of pump efficiencies with different pumps characteristics working in parallel mode. *Advances in Engineering Software*, 000:1–8, 2015. ISSN 18735339. doi: 10.1016/j.advengsoft.2015.10.010. URL `http://dx.doi.org/10.1016/j.advengsoft.2015.10.010`.

[143] Margus Koor, Anatoli Vassiljev, and Tiit Koppel. Optimal pump count prediction algorithm for identical pumps working in parallel mode. In *Conference on Computing and Control for the Water Industry*, 2013.

[144] Wojciech Kurek and Avi Ostfeld. Multi-objective optimization of water quality, pumps operation, and storage sizing of water distribution systems. *Journal of environmental management*, 115:189–97, jan 2013. ISSN 1095-8630. doi: 10.1016/j.jenvman.2012.11.030. URL `http://www.ncbi.nlm.nih.gov/pubmed/23262407`.

[145] K E Lansey. The evolution of optimizing water distribution system applications. In *Water Distribution Systems Analysis 2006*, Cincinnati, Ohio, 2006.

[146] Kevin E. Lansey and Larry W. Mays. Optimization model for water distribution system design. *Journal of Hydraulic Engineering*, 115(10):1401–1418, 1989.

[147] Sheng Liang. *The Java Native Interface Programmer's Guide and Specification*, volume 56. Addison-Wesley, Reading, MA, 1st edition, 1999. ISBN 0201325772. doi: 10.1088/0031-9155/56/23/002. URL `http://www.ncbi.nlm.nih.gov/pubmed/22053245`.

[148] Min-der Lin, Yu-hsin Liu, Gee-fon Liu, and Chien-wei Chu. Scatter search heuristic for least-cost design of water distribution networks. *Engineering Optimization*, 39(7):857–876, 2007. doi: 10.1080/03052150701503611.

[149] M. López-Ibáñez, D. T. Prasad, and B. Paechter. Parallel Optimisation of Pump Schedules with a Thread-Safe Variant of EPANET Toolkit. *Water Distribution Systems Analysis 2008*, pages 1–10, March 2009. doi: 10.1061/41024(340)40. URL `http://ascelibrary.org/doi/abs/10.1061/41024(340)40`.

[150] Manuel López-Ibáñez. *Operational Optimisation of Water Distribution Networks*. PhD thesis, Edinburgh Napier University, 2009.

[151] Manuel López-Ibáñez, T. Devi Prasad, and Ben Paechter. Ant colony optimization for optimal control of pumps in water distribution networks. *Journal of Water Resources Planning and Management*, 134(4):337–346, 2008. doi: 10.1061/?ASCE?0733-9496?2008?134:4?337?CE. URL `http://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-9496(2008)134:4(337)`.

[152] Manuel López-Ibáñez, T Devi Prasad, and Ben Paechter. Representations and evolutionary operators for the scheduling of pump operations in water distribution networks. *Evolutionary computation*, 19(3):429–467, 2011.

[153] Manuel López-Ibáñez, T D Prasad, and B Paechter. Optimal pump scheduling: Representation and multiple objectives. In *Conference on Computing and Control for the Water Industry*, 2005.

[154] Mark Lutz. *Learning python*. O'Reilly Media, 5th edition, 2013. ISBN 978-1449355739.

[155] E Jr. Luvizotto, M C Cavichia, P Vatavuk, and J G P Andrade. Nonmatrix Gradient Method for the Simulation of Water Distribution Networks. *Journal of Water Resources Planning and Management*, 139(4):433–439, 2013. doi: 10.1061/(ASCE)WR.1943-5452.

[156] Gerald Weinberg Lyle N. Hoag. Pipeline network analysis by electronic digital computer. *Journal (American Water Works Association)*, 49(5):517–524, 1957. ISSN 0003150X, 15518833. URL http://www.jstor.org/stable/41254856.

[157] Zhenjun Ma and Shengwei Wang. Energy efficient control of variable speed pumps in complex building central air-conditioning systems. *Energy and Buildings*, 41(2):197–205, 2009. ISSN 03787788. doi: 10.1016/j.enbuild.2008.09.002.

[158] J. Machell, S. R. Mounce, and J. B. Boxall. Online modelling of water distribution systems: a uk case study. *Drinking Water Engineering and Science*, pages 280–294, 2009.

[159] John Machin. xlrd documentation, 2017. URL http://xlrd.readthedocs.io/en/latest/.

[160] Günther Mäckle, Dragan A. Savić, and Godfrey A. Walters. Application of Genetic Algorithms to Pump Scheduling for Water Supply. In *Genetic Algorithms in Engineering Systems: Innovations and Applications*, number 414, pages 400–405, Sheffield, UK, 1995. IET.

[161] Holger R. Maier, Angus R. Simpson, Aaron C. Zecchin, Wai Kuan Foong, Kuang Yeow Phang, Hsin Yeow Seah, and Chan Lim Tan. Ant Colony Optimization for Design of Water Distribution Systems. *Journal of Water Resources Planning and Management*, 129(3):200–209, May 2003. ISSN 0733-9496. doi: 10.1061/(ASCE)0733-9496(2003)129:3(200). URL http://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-9496(2003)129:3(200).

[162] Michael Mair, Robert Sitzenfrei, Christian Urich, Manfred Kleidorfer, and Wolfgang Rauch. Performance improvement with parallel numerical model simulations in the field of urban water management. In *International Conference on Hydroinformatics HIC 2012*, Hamburg, Germany, 2012.

[163] Yasaman Makaremi, Ali Haghighi, and Hamid Reza Ghafouri. Optimization of Pump Scheduling Program in Water Supply Systems Using a Self-Adaptive NSGA-II; a Review of Theory to Real Application. *Water Resources Management*, 31(1283):1283–1304, 2017. ISSN 0920-4741. doi:

10.1007/s11269-017-1577-x. URL http://link.springer.com/10.1007/s11269-017-1577-x.

[164] A. Marchi, A. R. Simpson, and N. Ertugrul. Assessing variable speed pump efficiency in water distribution systems. *Drinking Water Engineering and Science*, 5(1):15–21, July 2012. ISSN 1996-9465. doi: 10.5194/dwes-5-15-2012. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84881113706&partnerID=tZOtx3y1.

[165] Angela Marchi and Angus R. Simpson. Correction of the EPANET Inaccuracy in Computing the Efficiency of Variable Speed Pumps. *Journal of Water Resources Planning and Management*, 139(4):456–459, July 2013. ISSN 0733-9496. doi: 10.1061/(ASCE)WR.1943-5452.0000273. URL http://ascelibrary.org/doi/abs/10.1061/%28ASCE%29WR.1943-5452.0000273.

[166] Angela Marchi, Angus R Simpson, and Nhu Cuong Do. Optimal Scheduling of Variable Speed Pumps in Water Distribution. In *OzWater 2012*, Sydney, Australia, 2012.

[167] Angela Marchi, Angus R. Simpson, and Martin F. Lambert. Optimization of Pump Operation Using Rule-Based Controls in EPANET2: New ETTAR Toolkit and Correction of Energy Computation. *Journal of Water Resources Planning and Management*, page 04016012, 2016. ISSN 0733-9496. doi: 10.1061/(ASCE)WR.1943-5452.0000637. URL http://ascelibrary.org/doi/abs/10.1061/{%}28ASCE{%}29WR.1943-5452.0000637.

[168] F Martínez, V Bou, V Hernández, F Alvarruiz, and J M Alonso. ANN architectures for simulating water distribution networks. In *Conference on Computing and Control for the Water Industry*, 2005.

[169] Fernando Martínez, Vicente Hernández, José Miguel Alonso, Zhengfu Rao, and Stefano Alvisi. Optimizing the operation of the Valencia water-distribution network. *Journal of Hydroinformatics*, 9(1):65, 2007. ISSN 14647141. doi: 10.2166/hydro.2006.018.

[170] G. McCormick and R. S. Powell. Optimal Pump Scheduling in Water Supply Systems with Maximum Demand Charges. *Journal of Water Resources Planning and Management*, 129(5):372–379, 2003. ISSN 0733-9496. doi: 10.1061/(ASCE)0733-9496(2003)129:5(372).

[171] G McCormick and R S Powell. Derivation of near-optimal pump schedules for water distribution by simulated annealing. *Journal of the Operational Research Society*, 55(0):728–736, 2004. ISSN 0160-5682. doi: 10.2307/4102019.

[172] S. Mohan and K. S. Jinesh Babu. Optimal Water Distribution Network Design with Honey-Bee Mating Optimization. *Journal of Computing in Civil Engineering*, 24(1):117–126, 2010. ISSN 0887-3801. doi: 10.1061/(ASCE)CP.1943-5487.0000018.

[173] Idel Montalvo, Joaquín Izquierdo, Rafael Pérez, and Michael M. Tung. Particle Swarm Optimization applied to the design of water supply systems. *Computers and Mathematics with Applications*, 56(3):769–776, 2008. ISSN 08981221. doi: 10.1016/j.camwa.2008.02.006.

[174] William Nagel. Embedding python in your c programs. *Linux journal*, 142:8, 2006.

[175] Vilas Nitivattananon, Elaine C. Sadowski, and Rafael G. Quimpo. Optimization of Water Supply System Operation. *Journal of Water Resources Planning and Management*, 122(5):374–384, sep 1996. ISSN 0733-9496. doi: 10.1061/(ASCE)0733-9496(1996)122:5(374). URL `http://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-9496(1996)122:5(374)`.

[176] Mateus Ricardo Nogueira Vilanova and José Antônio Perrella Balestieri. Energy and hydraulic efficiency in conventional water supply systems. *Renewable and Sustainable Energy Reviews*, 30:701–714, feb 2014. ISSN 13640321. doi: 10.1016/j.rser.2013.11.024. URL `http://linkinghub.elsevier.com/retrieve/pii/S1364032113007739`.

[177] F. Keizo Odan, L. F. Ribeiro Reis, and Z. Kapelan. Use of Metamodels in Real-Time Operation of Water Distribution Systems. *Procedia Engineering*, 89:449–456, 2014. ISSN 18777058. doi: 10.1016/j.proeng.2014.11.211. URL `http://linkinghub.elsevier.com/retrieve/pii/S1877705814023261`.

[178] Frederico Keizo Odan, Luisa Fernanda, and Ribeiro Reis. Hybrid Water Demand Forecasting Model Associating Artificial Neural Network and Fourier Series. In *Water Distribution Systems Analysis 2010*, Tucson, Arizona, 2010.

[179] Frederico Keizo Odan, Luisa Fernanda Ribeiro Reis, and Zoran Kapelan. Real-Time Multiobjective Optimization of Operation of Water Supply Systems. *Journal of Water Resources Planning and Management*,

page 04015011, feb 2015. ISSN 0733-9496. doi: 10.1061/(ASCE)WR.1943-5452.0000515. URL http://ascelibrary.org/doi/abs/10.1061/(ASCE)WR.1943-5452.0000515.

[180] Aidan O'Dwyer. *Handbook of PI and PID Controller Tuning Rules*, volume 26. Imperial College Press, London, 2006. ISBN 9781848162426. doi: 10.1109/MCS.2006.1580157.

[181] I. Okeya, Z. Kapelan, C. Hutton, and D. Naga. Online Modelling of Water Distribution System Using Data Assimilation. *Procedia Engineering*, 70(0):1261–1270, 2014. ISSN 18777058. doi: 10.1016/j.proeng.2014.02.139. URL http://linkinghub.elsevier.com/retrieve/pii/S1877705814001416.

[182] A Oliva and G Araújo. Speeding Up thread-local storage access in dynamic libraries. *Proceedings of the GCC Developers' Summit 2006*, pages 159–178, 2006. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-77954725082{&}partnerID=40{&}md5=fdcfe522d5dacaa746dde6ced8797be4.

[183] Pawel Olszewski. Genetic optimization and experimental verification of complex parallel pumping station with centrifugal pumps. *Applied Energy*, 178:527–539, 2016. ISSN 03062619. doi: 10.1016/j.apenergy.2016.06.084. URL http://dx.doi.org/10.1016/j.apenergy.2016.06.084.

[184] L E Ormsbee. The History of Water Distribution Network Analysis: The Computer Age. In *8th Annual Water Distribution Systems Analysis Symposium*, pages 1–6, Cincinnati, Ohio, 2006.

[185] Lindell E. Ormsbee and Kevin E. Lansey. Optimal Control of Water Supply System Pumping Systems. *Journal of Water Resources Planning and Management*, 120:237–252, 1994.

[186] Lindell E. Ormsbee and Srinivasa L. Reddy. Nonlinear Heuristic for Pump Operations. *Journal of Water Resources Planning and Management*, 121(4):302–309, 1995.

[187] A Ostfeld and A Tubaltzev. An ant colony non-dimensional algorithm for water distribution systems optimal design and operation. In *Conference on Computing and Control for the Water Industry*, pages 2–7, Exeter, UK, 2005.

[188] Avi Ostfeld and Shani Salomons. A hybrid genetic - Instance based learning algorithm for CE-QUAL-W2 calibration. *Journal of Hydrology*, 310(1-4): 122–142, 2005. ISSN 00221694. doi: 10.1016/j.jhydrol.2004.12.004.

[189] Sonja Ostojin, Stephen R Mounce, David Singerton, and Joby B Boxall. Operational control of sewer pumping stations to minimize energy costs considering efficiency. In *Conference on Computing and Control for the Water Industry*, 2011.

[190] E Pacchin, F Gagliardi, S Alvisi, and M Franchini. A comparison of short-term water demand forecasting models. In *Computing and Control for Water Industry*, Sheffield, UK, 2017.

[191] Daniel Paluszczyszyn, Piotr Skworcow, and Bogumil Ulanicki. Online simplification of water distiribution network models. In *Conference on Computing and Control for the Water Industry*, 2011.

[192] Daniel Paluszczyszyn, Piotr Skworcow, and Bogumil Ulanicki. Online simplification of water distribution network models for optimal scheduling. *Journal of Hydroinformatics*, 15(3):652–665, 2013. doi: 10.2166/hydro.2013.029.

[193] Daniel Paluszczyszyn, Piotr Skworcow, and Bogumil Ulanicki. Modelling and simulation of water distribution systems with quantised state system methods. *Procedia Engineering*, 119(1):554–563, 2015. ISSN 18777058. doi: 10.1016/j.proeng.2015.08.908. URL `http://dx.doi.org/10.1016/j.proeng.2015.08.908`.

[194] M. F. K. Pasha and K. Lansey. Optimal Pump Scheduling by Linear Programming. *World Environmental and Water Resources Congress 2009: Great Rivers*, (2006):38–38, 2009. doi: 10.1061/41036(342)38. URL `http://link.aip.org/link/ASCECP/v342/i41036/p38/s1&Agg=doi`.

[195] M. F. K. Pasha and K. Lansey. Strategies for Real Time Pump Operation for Water Distribution Systems. In *Water Distribution Systems Analysis 2010*, pages 1456–1469, Reston, VA, dec 2011. American Society of Civil Engineers. ISBN 978-0-7844-1203-9. doi: 10.1061/41203(425)130. URL `http://ascelibrary.org/doi/abs/10.1061/41203(425)130`.

[196] M. Fayzul. K. Pasha and Kevin Lansey. Strategies to Develop Warm Solutions for Real-Time Pump Scheduling for Water Distribution Systems. *Water Resources Management*, 28(12):3975–3987, July 2014. ISSN 0920-4741.

doi: 10.1007/s11269-014-0721-0. URL http://link.springer.com/10.1007/s11269-014-0721-0.

[197] Santiago Pericas-Geertsen and Marek Potociar, editors. *JAX-RS: Java™ API for RESTful Web Services 2.0.* Oracle Corporation, Redwood Shores, California, 2013.

[198] T. Devi Prasad and Tiku T. Tanyimboh. Entropy Based Design of "Anytown" Water Distribution Network. *Water Distribution Systems Analysis 2008*, pages 1–12, mar 2009. doi: 10.1061/41024(340)39. URL http://ascelibrary.org/doi/abs/10.1061/41024(340)39.

[199] A. Preis, T. Obaid, M. Allen, M. Iqbal, and A.J. Whittle. Real-time hydraulic modelling of a water distribution system in Singapore. In *Water Distribution System Analysis Conference*, 2010.

[200] E. Price and A. Ostfeld. Optimal water system operation using graph theory algorithms. *Procedia Engineering*, 89:502–508, 2014. ISSN 18777058. doi: 10.1016/j.proeng.2014.11.245. URL http://dx.doi.org/10.1016/j.proeng.2014.11.245.

[201] E. Price and A. Ostfeld. Practical approach to water system optimal operation. *Procedia Engineering*, 70:1362–1368, 2014. ISSN 18777058. doi: 10.1016/j.proeng.2014.02.150. URL http://dx.doi.org/10.1016/j.proeng.2014.02.150.

[202] Eyal Price and Avi Ostfeld. Headloss successive linearization for optimal operation of water distribution systems. In *Conference on Computing and Control for the Water Industry*, 2011.

[203] Eyal Price and Avi Ostfeld. Discrete Pump Scheduling and Leakage Control Using Linear Programming for Optimal Operation of Water Distribution Systems. *Journal of Hydraulic Engineering*, 140(6):1–16, 2014. doi: 10.1061/(ASCE)HY.1943-7900.0000864.

[204] Eyal Price and Avi Ostfeld. Graph Theory Modeling Approach for Optimal Operation of Water Distribution Systems. *Journal of Hydraulic Engineering*, 142(3):1–15, 2016. ISSN 9780784479865. doi: 10.1061/(ASCE)HY.1943-7900.0001099.

[205] V. Puleo, C.M. Fontanazza, V. Notaro, and G. Freni. Multi Sources Water Supply System Optimal Control: A Case Study. *Procedia Engineering*, 89:247–254, 2014. ISSN 18777058. doi: 10.1016/j.proeng.2014.11.184.

URL http://linkinghub.elsevier.com/retrieve/pii/S1877705814022991.

[206] V. Puleo, M. Morley, G. Freni, and D. Savić. Multi-stage linear programming optimization for pump scheduling. *Procedia Engineering*, 70:1378–1385, 2014. ISSN 18777058. doi: 10.1016/j.proeng.2014.02.152.

[207] R Puust, Z Kapelan, D Savic, and T Koppel. Probabilistic leak detection in pipe networks using the scem-ua algorithm. In *8th Annual Water Distribution Systems Analysis Symposium*, Cincinnati, Ohio, 2006.

[208] Raido Puust, Madis Maddison, and Janek Laanearu. Reviewing the effectiveness of gpu power when used for water network optimization problems. In *Conference on Computing and Control for the Water Industry*, 2011.

[209] Joseba Quevedo. Optimisation methods for the calibration of water network systems. In *Conference on Computing and Control for the Water Industry*, number 1974, pages 273–282, 2003.

[210] Darian Raad, Alexander Sinske, and Jan van Vuuren. Robust multi-objective optimization for water distribution system design using a meta-metaheuristic. *International Transactions in Operational Research*, 16 (5):595–626, 2009. ISSN 09696016. doi: 10.1111/j.1475-3995.2009.00705.x. URL http://gateway.library.qut.edu.au/login?url=http://search.ebscohost.com/login.aspx?direct=true{&}db=eoah{&}AN=19216617{&}site=ehost-live.

[211] Z Rao, J Wicks, and S West. Encoms - An energy cost minimisation system for real-time, operational control of water distribution networks. In *Conference on Computing and Control for the Water Industry*, 2005.

[212] Zhengfu Rao and Fernando Alvarruiz. Use of an artificial neural network to capture the domain knowledge of a conventional hydraulic simulation model. *Journal of Hydroinformatics*, 9(1):15, 2007. ISSN 14647141. doi: 10.2166/hydro.2006.014.

[213] Zhengfu Rao and Elad Salomons. Development of a real-time , near-optimal control process for water-distribution networks. *Journal of Hydroinformatics*, 9(1):25–37, 2007. doi: 10.2166/hydro.2006.015.

[214] Saman Razavi, Bryan A. Tolson, L. Shawn Matott, Neil R. Thomson, Angela MacLean, and Frank R. Seglenieks. Reducing the computational cost of automatic calibration through model preemption. *Water Resources Research*, 46(11):1–17, 2010. ISSN 00431397. doi: 10.1029/2009WR008957.

[215] Saman Razavi, Bryan A. Tolson, and Donald H. Burn. Review of surrogate modeling in water resources. *Water Resources Research*, 48(7), 2012. ISSN 00431397. doi: 10.1029/2011WR011527.

[216] Saman Razavi, Bryan A. Tolson, and Donald H. Burn. Numerical assessment of metamodelling strategies in computationally intensive optimization. *Environmental Modelling and Software*, 34:67–86, 2012. ISSN 13648152. doi: 10.1016/j.envsoft.2011.09.010. URL http://dx.doi.org/10.1016/j.envsoft.2011.09.010.

[217] Juan Reca and Juan Martínez. Genetic algorithms for the design of looped irrigation water distribution networks. *Water Resources Research*, 42(5):1–9, 2006. ISSN 00431397. doi: 10.1029/2005WR004383.

[218] Juan Reca, Alfonso García-Manzano, and Juan Martínez. Optimal Pumping Scheduling for Complex Irrigation Water Distribution Systems. *Journal of Water Resources Planning and Management*, 140(5):630–637, 2014. ISSN 0733-9496. doi: 10.1061/(ASCE)WR.1943-5452.0000360. URL http://ascelibrary.org/doi/abs/10.1061/{%}28ASCE{%}29WR.1943-5452.0000360.

[219] L Reis and T Pinheiro. Combined genetic algorithm-linear programming (GA-LP) procedure applied to the operation of the Fortaleza water supply reservoir system (Brazil). In *Conference on Computing and Control for the Water Industry*, 2005.

[220] RIL. *RIL 124-2 Vesihuolto II*. Suomen rakennusinsinöörien liitto RIL ry, 2004.

[221] M. Romano, Z. Kapelan, and D. Savić. Automated Detection of Pipe Bursts and Other Events in Water Distribution Systems. *Journal of Water Resources Planning and Management*, 140(4):457––467, 2014.

[222] Lewis A Rossman. An Overview of EPANET Version 3.0. In *Water Distribution Systems Analysis 2010*, pages 14–18, Tucson, Arizona, 2011. ISBN 978-0-7844-1203-9. doi: doi:10.1061/41203(425)3. URL http://ascelibrary.org/doi/abs/10.1061/41203{%}28425{%}293.

[223] Lewis A. Rossman and J. E. van Zyl. The Open Sourcing of EPANET. In *Water Distribution Systems Analysis 2010*, Tucson, Arizona, December 2010. American Society of Civil Engineers. ISBN 978-0-7844-1203-9. doi: 10.1061/41203(425)4. URL http://ascelibrary.org/doi/abs/10.1061/41203(425)4.

[224] Lewis A. Rossman and J. E. Van Zyl. The Open Sourcing of EPANET. In *Water Distribution Systems Analysis 2010*, Tucson, Arizona, December 2010. American Society of Civil Engineers. ISBN 978-0-7844-1203-9. doi: 10.1061/41203(425)4. URL `http://ascelibrary.org/doi/abs/10.1061/41203(425)4`.

[225] Lewis A Rossmann. The epanet programmer's toolkit for analysis of water distribution systems. In *ASCE 29th Annual Water Resources Planning and Management Conference*, pages 39–48, 1999.

[226] Lewis A. Rossmann. *EPANET 2 Users manual*. U.S. Environmental Protection Agency, 2000.

[227] R Saidur. A review on electrical motors energy use and energy savings. *Renewable and Sustainable Energy Reviews*, 14:877–898, 2010. doi: 10.1016/j.rser.2009.10.018.

[228] R Saidur, S Mekhilef, M B Ali, A Safari, and H A Mohammed. Applications of variable speed drive (VSD) in electrical motors energy savings. *Renewable and Sustainable Energy Reviews*, 16(1):543–550, 2012. ISSN 1364-0321. doi: 10.1016/j.rser.2011.08.020. URL `http://dx.doi.org/10.1016/j.rser.2011.08.020`.

[229] A B A Sakarya and Larry W. Mays. Optimal Operation of Water Distribution Pumps Considering Water Quality. *Journal of Water Resources Planning and Management*, 126(4):210–220, 2000.

[230] J Saldarriaga, D Páez, L López, and O Díaz. A new Integer Linear Programming based Design Methodology for Water Distribution systems. In *World Environmental and Water Resources Congress 2014*, pages 577–590, 2014.

[231] J G Saldarriaga, A Bernal, and S Ochoa. Optimized Design of Water Distribution Network Enlargements Using Resilience and Dissipated Power Concepts. In J E van Zyl, A A Ilemobade, and H E Jacobs, editors, *Water Distribution Systems Analysis 2010*, pages 298–312, Kruger National Park, South Africa, 2008.

[232] Juan Saldarriaga and Silvia Takahashi. Multi-objective water distribution system design using an expert algorithm. In *Conference on Computing and Control for the Water Industry*, 2011.

[233] Juan Saldarriaga, Laura Nieto, Ivonne Navarro, and Cesar Niño. Calibration Process of a Chlorine Decay Model: The Case Study of Bogota. In *Water Distribution Systems Analysis 2010*, Tucson, Arizona, 2010.

[234] Elad Salomons, Alexander Goryashko, Uri Shamir, Zhengfu Rao, and Stefano Alvisi. Optimizing the operation of the Haifa-A water-distribution network. *Journal of Hydroinformatics*, 9(1):51–64, 2007. doi: 10.2166/hydro.2006.017.

[235] I. Sârbu and I Borza. Energetic optimization of water pumping in distribution systems. *Periodica polytechnica ser. mech. eng.*, 42(2):141–152, 1998. URL http://www.pp.bme.hu/me/1998_2/me1998_2_05.html.

[236] Dragan A Savic, Godfrey A Walters, and Martin Schwab. Multiobjective Genetic Algorithms for Pump Scheduling in Water Supply. *Evolutionary Computing*, pages 227–235, 1997.

[237] Sonja Saviranta. *Energy Audit of Finnish Water Distribution Networks.* Master's thesis, Tampere University of Technology, 2015. URL http://urn.fi/URN:NBN:fi:tty-201511171722.

[238] István Selek, József Gergely Bene, and Csaba Hős. Optimal (short-term) pump schedule detection for water distribution systems by neutral evolutionary search. *Applied Soft Computing*, 12(8):2336–2351, aug 2012. ISSN 15684946. doi: 10.1016/j.asoc.2012.03.045. URL http://www.sciencedirect.com/science/article/pii/S1568494612001433.

[239] Uri Shamir. *Water Distribution Systems Analysis.* IBM, 1973.

[240] Uri Shamir and Elad Salomons. Optimal Real-Time Operation of Urban Water Distribution Systems Using Reduced Models. *Journal of Water Resources Planning and Management*, 134(2):181–185, 2008. ISSN 0733-9496. doi: 10.1061/(ASCE)0733-9496(2008)134:2(181).

[241] H Shi and Y Fengqi. Energy Optimization of Water Supply System Scheduling: Novel MINLP Model and Efficient Global Optimization Algorithm. *AIChE Journal*, 62(12):4277–4296, 2016. doi: 10.1002/aic.15332.

[242] Calvin Siew and Tiku T Tanyimboh. Penalty-Free Multi-Objective Evolutionary Optimization Approach for Water Distribution Systems. In *Water Distribution Systems Analysis 2010*, Tucson, Arizona, 2010.

[243] Angus R. Simpson. Modeling of Pressure Regulating Devices: The Last Major Problem to be Solved in Hydraulic Simulation Angus R. Simpson. In *WRPMD'99: Preparing for the 21st Century*, number March, pages 1–9, 1999. ISBN 0784404305. doi: 10.1061/40430(1999)40.

[244] Angus R. Simpson and Angela Marchi. Evaluating the Approximation of the Affinity Laws and Improving the Efficiency Estimate for Variable Speed Pumps. *Journal of Hydraulic Engineering*, 139 (12):1314–1317, 2013. ISSN 0733-9429. doi: 10.1061/(ASCE)HY.1943-7900.0000776. URL http://ascelibrary.org/doi/abs/10.1061/(ASCE)HY.1943-7900.0000776.

[245] Angus R Simpson, Graeme C Dandy, and Laurence J Murphy. Genetic algorithms compared to other techniques for pipe optimization. *Journal of Water Resources Planning and Management*, 120(4):423–443, 1994.

[246] P. Skworcow, D. Paluszczyszyn, B. Ulanicki, R. Rudek, and T. Belrain. Optimisation of pump and valve schedules in complex large-scale water distribution systems using GAMS modelling language. *Procedia Engineering*, 70: 1566–1574, 2014. ISSN 18777058. doi: 10.1016/j.proeng.2014.02.173. URL http://dx.doi.org/10.1016/j.proeng.2014.02.173.

[247] Piotr Skworcow, Hossam Abdelmeguid, Bogumil Ulanicki, and Peter Bounds. Optimal pump scheduling with pressure control aspects : Case studies. In *Integrating Water Systems: Proceedings of the 10th International Conference on Computing and Control in the Water Industry*, pages 113–119, 2010. ISBN 9780415548519.

[248] Davide Soldi, Antonio Candelieri, and Francesco Archetti. Resilience and vulnerability in urban water distribution networks through network theory and hydraulic simulation. *Procedia Engineering*, 119(1):1259–1268, 2015. ISSN 18777058. doi: 10.1016/j.proeng.2015.08.990. URL http://dx.doi.org/10.1016/j.proeng.2015.08.990.

[249] D. Steffelbauer and D. Fuchs-Hanusch. OOPNET: An object-oriented EPANET in Python. *Procedia Engineering*, 119:710–718, 2015. ISSN 18777058. doi: 10.1016/j.proeng.2015.08.924. URL http://linkinghub.elsevier.com/retrieve/pii/S1877705815025941.

[250] R. A. Strub, L. Bonciani, C. J. Borer, M. V. Casey, S. L. Cole, B. B. Cook, J. Kotzur, H. Simon, and M. A. Strite. Influence of the Reynolds Number on the Performance of Centrifugal Compressors. *Journal of Turbomachinery*, 109:541–544, 1987. ISSN 0889504X. doi: 10.1115/1.3262145.

[251] Yung-Hsin Sun, William W-G Yeh, Nien-Sheng Hsu, and Peter W F Louie. Generalized Network Algorithm for Water-Supply-System Optimization. *Journal of Water Resources Planning and Management*, 121(5):392–398, 1995.

[252] Markus Sunela. Modelling water distribution network control system (in Finnish). Master's thesis, Tampere University of Technology, October 2010.

[253] M.I. Sunela and R. Puust. Simple visual tool to analyse pump battery efficiencies for various pump combinations. *Procedia Engineering*, 89:525–532, 2014. ISSN 1877-7058. doi: http://dx.doi.org/10.1016/j.proeng.2014.11.474. 16th Water Distribution System Analysis Conference.

[254] M.I. Sunela and R. Puust. A visual tool to calculate optimal control strategy for non-identical pumps working in parallel, taking motor and vsd efficiencies into account. *Water Science and Technology: Water Supply*, 15(5): 1115–1122, 2015.

[255] M.I. Sunela and R. Puust. Real time water supply system hydraulic and quality modeling – a case study. *Procedia Engineering*, 119:744–752, 2015.

[256] M.I. Sunela and R. Puust. Modeling water supply system control system algorithms. *Procedia Engineering*, 119:734–743, 2015.

[257] M. Tabesh, A. H. Asadiyami Yekta, and R. Burrows. An integrated model to evaluate losses in water distribution systems. *Water Resources Management*, 23(3):477–492, 2009. ISSN 09204741. doi: 10.1007/s11269-008-9284-2.

[258] Sarah Thorstensen. Energy Savings Analysis Generated by a Real Time Energy Management System for Water Distribution. In *Proceedings of the AWWA Pennsylvania/Chesapeake section conference*, Hershey, PA, 2007.

[259] E Todini. On the Convergence Properties of the Different Pipe Network Algorithms. In *8th Annual Water Distribution Systems Analysis Symposium*, pages 1–16, Cincinnati, Ohio, 2006.

[260] E. Todini and S. Pilati. A gradient method for the analysis of pipe networks. In *Computer Applications in Water Supply*, volume 1, pages 1–20, 1988.

[261] E. Todini, M. E. Tryby, Z. Y. Wu, and T .M. Walski. Direct Computation of Variable Speed Pumps for Water Distribution System Analysis. In *Computing and Control for Water Industry*, 2007.

[262] Ezio Todini and Lewis A Rossman. Unified Framework for Deriving Simultaneous Equation Algorithms for Water Distribution Networks. *Journal of Hydraulic Engineering*, 139(5):511–526, 2013. doi: 10.1061/(ASCE)HY.1943-7900.0000703.

[263] Ezio Todini, Michael E Tryby, Jack S Cook, Thomas M Walski, and Robert F Mankowski. Automatic parameter estimation extension for variable speed pumps, March 14 2006. US Patent 7,013,248.

[264] Bryan A. Tolson and Christine A. Shoemaker. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resources Research*, 43(1):1–16, 2007. ISSN 00431397. doi: 10.1029/2005WR004723.

[265] Bryan A. Tolson, Masoud Asadzadeh, Holger R. Maier, and Aaron Zecchin. Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization. *Water Resources Research*, 45(12):1–15, 2009. ISSN 00431397. doi: 10.1029/2008WR007673.

[266] Vu Nam Tran and Mietek A. Brdys. Optimizing control by robustly feasible model predictive control and application to drinking water distribution systems. *Artificial Neural Networks – ICANN 2009*, 5769:823–834, 2009. ISSN 03029743. doi: 10.1007/978-3-642-04277-5_83.

[267] B Ulanicki, A Zehnpfund, and F Martinez. Simplification of water distribution network models. In *Proc., 2nd Int. Conf. on Hydroinformatics*, pages 493–500. Balkema Rotterdam, Netherlands, 1996.

[268] B Ulanicki, P L M Bounds, and J P Rance. Using a GAMS modelling environment to solve network scheduling problems. *Measurement + Control*, 32:110–115, 1999. ISSN 00202940.

[269] B Ulanicki, J Kahler, and B Coulbeck. Modeling the efficiency and power characteristics of a pump group. *Journal of Water Resources Planning and Management*, (February):88–94, 2008. URL http://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-9496(2008)134:1(88).

[270] J. E. van Zyl and Y. Chang. Programming Styles for an Open Source EPANET Project. In *Water Distribution Systems Analysis 2010*, pages 29–36, Tucson, Arizona, 2011. American Society of Civil Engineers. ISBN 978-0-7844-1203-9. doi: 10.1061/41203(425)5. URL http://ascelibrary.org/doi/abs/10.1061/41203(425)5.

[271] J E van Zyl, J Borthwick, and A Hardy. OOTEN: An object-oriented programmers toolkit for Epanet. In *Advances in Water Supply Management (CCWI 2003)*, 2003.

[272] Jakobus E. van Zyl, Dragan A. Savic, and Godfrey A. Walters. Operational Optimization of Water Distribution Systems Using a Hybrid Genetic Algorithm. *Journal of Water Resources Planning and Management*, 130(2):160–170, March 2004. ISSN 0733-9496. doi: 10.1061/(ASCE)0733-9496(2004)130:2(160). URL http://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-9496(2004)130%3A2(160).

[273] Jakobus E van Zyl, Prabhat Kumar, and Mayank Gupta. Two-Point Linearization Method for the Analysis of. *Journal of Hydraulic Engineering*, 134(8):1176–1179, 2008.

[274] A. Vasan and Slobodan P. Simonovic. Optimization of Water Distribution Network Design Using Differential Evolution. *Journal of Water Resources Planning and Management*, 136(2):279–287, 2010. ISSN 0733-9496. doi: 10.1061/(ASCE)0733-9496(2010)136:2(279).

[275] A Vassiljev, T Koppel, and R Puust. Calibration of the model of an operational water distribution system. In *Conference on Computing and Control for the Water Industry*, 2005.

[276] Juan David Velásquez-Henao, Carlos Jaime Franco-Cardona, and Yris Olaya-Morales. A Review of DAN2 (Dynamic Architecture for Artificial Neural Networks) Model in Time Series Forecasting 1. *Ing. Univ. Bogotá*, 16(1):135–146, 2012.

[277] Bou Vicente, Fernando Martínez, and Pilar Conejos. Sca-red, a general purpose scada application for taking decisions in real time with the aid of a hydraulic model. In *Water Distribution Systems Analysis 2006*, Cincinnati, Ohio, 2006.

[278] Filipe Vieira and Helena M. Ramos. Optimization of operational planning for wind/hydro hybrid water supply systems. *Renewable Energy*, 34(3):928–936, mar 2009. ISSN 09601481. doi: 10.1016/j.renene.2008.05.031. URL http://www.sciencedirect.com/science/article/pii/S0960148108002292.

[279] Juha Viholainen. *ENERGY-EFFICIENT CONTROL STRATEGIES FOR VARIABLE SPEED DRIVEN PARALLEL PUMPING SYSTEMS BASED ON PUMP OPERATION POINT*. PhD thesis, Lappeenranta University of Technology, 2014.

[280] Juha Viholainen, Jussi Tamminen, Tero Ahonen, Jero Ahola, Esa Vakkilainen, and Risto Soukka. Energy-efficient control strategy for variable

speed-driven parallel pumping systems. *Energy Efficiency*, 6(3):495–509, 2013. ISSN 1570646X. doi: 10.1007/s12053-012-9188-0.

[281] Michael Volk. *Pump Characteristics and Applications*. Taylor & Francis Group, 2nd edition, 2005. ISBN 978-0-82472-755-0.

[282] Christian von Lücken, Benjamin Barán, and Aldo Sotelo. Pump Scheduling Optimization Using Asynchronous Parallel Evolutionary Algorithms. *CLEI Electorinic Journal*, 7(2), 2004.

[283] T. Walski, K. Zimmerman, M. Dudinyak, and P. Dileepkumar. Some Surprises in Estimating the Efficiency of Variable-Speed Pumps with the Pump Affinity Laws. In *World Water & Environmental Resources Congress*, pages 1–10, Philadelphia, 2003. URL http://ascelibrary.org/doi/abs/10.1061/40685(2003)137.

[284] T. M. Walski. Water distribution system analysis before digital computers. In *8th Annual Water Distribution Systems Analysis Symposium*, pages 1–8, 2006.

[285] Thomas M. Walski. Water distribution system analysis before digital computers. In *8th Annual Water Distribution Systems Analysis Symposium*, 2006.

[286] Thomas M Walski, E Downey Brill, Johannes Gessler, Ian C Goulter, Roland M Jeppson, Kevin Lansey, Han-lin Lee, Jon C Liebman, and Larry Mays. Battle of the Network Models: Epilogue. *Journal of Water Resources Planning and Management*, 113(2):191–203, 1987.

[287] Thomas M. Walski, Donald V. Chase, Dragan A. Savic, Walter Grayman, Stephen Beckwith, and Edmundo Koelle. *Advanced Water Distribution Modeling and Management*. Haestad Press, 1st edition, 2003. ISBN 0-9714141-2-2.

[288] Huanhuan Wang, Shuming Liu, Fanlin Meng, and Mingming Li. Gene expression programming algorithms for optimization of water distribution networks. *Procedia Engineering*, 37(Cems):359–364, 2012. ISSN 18777058. doi: 10.1016/j.proeng.2012.04.253. URL http://dx.doi.org/10.1016/j.proeng.2012.04.253.

[289] Jen Ya Wang, Tsui Ping Chang, and Jr Shian Chen. An enhanced genetic algorithm for bi-objective pump scheduling in water supply. *Expert Systems with Applications*, 36(7):10249–10258, 2009. ISSN 09574174. doi: 10.1016/j.eswa.2009.01.054. URL http://dx.doi.org/10.1016/j.eswa.2009.01.054.

[290] Jingsong Wang and Mietek A. Brdys. Optimal operation of water distribution systems under full range of operating scenarios. In *Proceedings of 6th UKACC International Control Conference*, Glasgow, 2003.

[291] Shengwei Wang and John Burnett. ISSN 13594311. doi: 10.1016/S1359-4311(00)00109-5.

[292] Rafał Weron and Adam Misiorek. Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models. *International Journal of Forecasting*, 24(4):744–763, 2008. ISSN 01692070. doi: 10.1016/j.ijforecast.2008.08.004.

[293] F. J. Wiesner. A New Appraisal of Reynolds Number Effects on Centrifugal Compressor Performance. *Journal of Engineering for Power*, 101(3):384, July 1979. ISSN 00220825. doi: 10.1115/1.3446586. URL `http://gasturbinespower.asmedigitalcollection.asme.org/article.aspx?articleid=1419801`.

[294] Peng Wu, Zhounian Lai, Dazhuan Wu, and Leqin Wang. Optimization Research of Parallel Pump System for Improving Energy Efficiency. *Journal of Water Resources Planning and Management*, pages 1–8, 2014. doi: 10.1061/(ASCE)WR.1943-5452.0000493.

[295] Wenyan Wu, Angus R Simpson, and Holger R Maier. Multi-Objective Genetic Algorithm Optimisation of Water Distribution Systems Accounting for Sustainability. In *Water Down Under 2008*, Adeline, Australia, 2008.

[296] Wenyan Wu, Angus R. Simpson, Holger R. Maier, and Angela Marchi. Incorporation of Variable-Speed Pumping in Multiobjective Genetic Algorithm Optimization of the Design of Water Transmission Systems. *Journal of Water Resources Planning and Management*, 138(5):543–552, sep 2012. ISSN 0733-9496. doi: 10.1061/(ASCE)WR.1943-5452.0000195. URL `http://ascelibrary.org/doi/abs/10.1061/(ASCE)WR.1943-5452.0000195`.

[297] Z. Y. Wu and E Todini. Extended Approach for Modelling Fixed-flow Variable Speed Pumps. In *Computing and Control for Water Industry*, 2007. ISBN 9780415454155.

[298] Zhen Yi Wu and Qi Zhu. Scalable parallel computing framework for pump scheduling optimization. In *World Environmental and Water Resources Congress 2009*, pages 1–11, 2009. doi: 10.1061/41036(342)

42. URL http://ascelibrary.org/doi/abs/10.1061/41036%28342%2942.

[299] Zheng Y. Wu and Angus R. Simpson. A self-adaptive boundary search genetic algorithm and its application to water distribution systems. *Journal of Hydraulic Research*, 40(2):191–203, 2002. ISSN 0022-1686. doi: 10.1080/00221680209499862.

[300] Zheng Yi Wu and A. A. Eftekharian. Parallel artificial neural network using CUDA-enabled GPU for extracting hydraulic domain knowledge of large water distribution systems. In *World Environmental and Water Resources Congress 2011*, pages 79–92, 2011.

[301] Zheng Yi Wu and Ian Lee. Lessons for parallelizing linear equation solvers and water distribution analysis. In *Conference on Computing and Control for the Water Industry*, 2011.

[302] Zheng Yi Wu and Paul Sage. Water loss detection via genetic algorithm optimization-based model calibration. In *8th Annual Water Distribution Systems Analysis Symposium*, Cincinnati, Ohio, 2006.

[303] Zheng Yi Wu, Qiben Wang, Swaroop Butala, and Tian Mi. Generalized framework for high performance infrastructure system optimization. In *Conference on Computing and Control for the Water Industry*, 2011.

[304] Zhenyu Yang and H Borsting. Energy efficient control of a boosting system with multiple variable-speed pumps in parallel. *Decision and Control (CDC), 2010 49th ...*, pages 2198–2203, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5717312.

[305] D. F. Yates, A. B. Templeman, and T. B. Boffey. The Computational Complexity of the Problem of Determining Least Capital Cost Designs for Water Supply Networks. *Engineering Optimization*, 7(March 2015):143–155, 1984. ISSN 0305-215X. doi: 10.1080/03052158408960635. URL http://www.tandfonline.com/doi/abs/10.1080/03052158408960635.

[306] A Yazdani and P Jeffrey. Robustness and vulnerability analysis of water distribution networks using graph theoretic and complex network principles. In *Water Distribution Systems Analysis 2010*, Tucson, Arizona, 2010.

[307] G. Yu, R. S. Powell, and M. J H Sterling. Optimized pump scheduling in water distribution systems. *Journal of Optimization Theory and Applications*, 83(3):463–488, 1994. ISSN 00223239. doi: 10.1007/BF02207638.

[308] Mehmet Ali Yurdusev and Mahmut Firat. Adaptive neuro fuzzy inference system approach for municipal water consumption modeling: An application to izmir, turkey. *Journal of Hydrology*, 365(3-4):225 – 234, 2009. ISSN 0022-1694. doi: DOI: 10.1016/j.jhydrol.2008.11.036. URL `http://www.sciencedirect.com/science/article/B6V6C-4V3HFJ6-2/2/c2022765477b9431c6491806f434ee69`.

[309] U. Zessler and U. Shamir. Optimal Operation of Water Distribution Systen. *Journal of Water Resources Planning and Management*, 115:735–752, 1989.

[310] Jingjing Zhang, Rengao Song, Nageshwar R Bhaskar, and Mark N French. Short-term water demand forecasting: a case study. In *8th Annual Water Distribution Systems Analysis Symposium*, pages 1–14, Cincinnati, Ohio, 2006.

[311] Hongbin Zhao, Chenguang Wu, Jinliang Gao, Yurun Liu, Qiang Wang, and Weiping Guan. Study and applications of on-line optimal operation in typical urban water supply system of china. In *Conference on Computing and Control for the Water Industry*, pages 2–7, 2005.

[312] Feifei Zheng. Comparing the Real-Time Searching Behavior of Four Differential-Evolution Variants Applied to Water-Distribution-Network Design Optimization. *Journal of Water Resources Planning and Management*, 2015. doi: 10.1061/(ASCE)WR.1943-5452.0000534.

[313] Feifei Zheng, Angus R Simpson, Aaron C Zecchin, and Jochen W Deuerlein. A graph decomposition-based approach for water distribution network optimization. *Water Resources Research*, 49:2093–2109, 2013. doi: 10.1002/wrcr.20175.

[314] Feifei Zheng, Aaron C Zecchin, Angus R Simpson, M Asce, and Martin F Lambert. Noncrossover Dither Creeping Mutation-Based Genetic Algorithm for Pipe Network Optimization. *Journal of Water Resources Planning and Management*, 140:553–557, 2014. doi: 10.1061/(ASCE)WR.1943-5452.0000351.

[315] Feifei Zheng, Angus Simpson, and Aaron Zecchin. Environmental Modelling & Software Improving the efficiency of multi-objective evolutionary algorithms through decomposition: An application to water distribution network design. *Environmental Modelling and Software*, 69:240–252, 2015. ISSN 1364-8152. doi: 10.1016/j.envsoft.2014.08.022. URL `http://dx.doi.org/10.1016/j.envsoft.2014.08.022`.

[316] Guilin Zheng and Qi Huang. Energy Optimization Study of Rural Deep Well Two-Stage Water Supply Pumping Station. *IEEE Transactions on Control Systems Technology*, PP(99):1–9, 2015. doi: 10.1109/TCST.2015.2498140.

[317] S.L. Zhou, T.a. McMahon, A. Walton, and J. Lewis. Forecasting daily urban water demand: a case study of Melbourne. *Journal of Hydrology*, 236(3-4):153–164, sep 2000. ISSN 00221694. doi: 10.1016/S0022-1694(00)00287-0. URL `http://linkinghub.elsevier.com/retrieve/pii/S0022169400002870`.

[318] Xiaojun Zhou, David Yang Gao, and Angus R Simpson. Optimal Design of Water Distribution Networks by Discrete State Transition Algorithm. *Engineering Optimization*, 48(4):603–628, 2016. doi: 10.1080/0305215X.2015.1025775.

[319] Xiangtao Zhuan and Xiaohua Xia. Optimal operation scheduling of a pumping station with multiple pumps. *Applied Energy*, 104:250–257, 2013. ISSN 03062619. doi: 10.1016/j.apenergy.2012.10.028. URL `http://dx.doi.org/10.1016/j.apenergy.2012.10.028`.

CURRICULUM VITAE

Personal data

Name:	Markus Ilmari Sunela
Date of birth:	December 30th 1985
Place of birth:	Kurikka, Finland
Citizenship:	Finnish

Contact data

Address:	Rotkonraitti 6 B 13, 33720 TAMPERE, FINLAND
Phone:	+385 40 768 5165
E-mail:	markus.sunela@iki.fi

Education

2012–2017:	Tallinn University of Technology	PhD, Civil & Env. Eng.
2008–2010:	Tampere University of Technology	MSc, Water & Waste Mgmt.
2008–2009:	Tallinn University of Technology	MSc, exchange studies
2005–2008:	Tampere University of Technology	BSc, Water & Waste Mgmt.
2001–2004:	Kurikan lukio	Highschool

Language competence

Finnish:	Native speaker
English:	Fluent
Estonian:	Fluent
Swedish:	Fluent
Russian:	Average
Esperanto:	Fluent

Professional employment

2017– :	Fluidit Ltd.	CTO, Chairman
2016– :	Tampere University of Technology, Lab. of Pervasive Computing	Visiting Research Fellow
2015– :	Tampere University of Technology, Lab. of Chemistry and Bioeng.	Visiting Teacher
2014– 2017:	FCG Design and Engineering Ltd.	Team Leader
2011– 2014:	FCG Design and Engineering Ltd.	Senior Specialist
2010– 2011:	FCG Finnish Consulting Group Ltd.	Project Engineer
2006– 2010:	Norfello Inc.	Software Developer
2005:	Finnish Defence Forces, Western Command HQ	Systems Engineer
2005:	Finnish Defence Forces, Army Academy	Research Assistant
2004:	National Land Survey of Finland	Surveyor

Honors & certifications

2017:	Best Student Paper and Presentation, CCWI 2017
2016:	Young Consultant of the Year, Finalist
2013:	FEANI EUR ING License # 32440
2010:	SNIL Sholarship for Master of Science
2010:	Master's Degree with Honors
2008:	Bachelor's Degree with Honors

Co-supervised theses

2016:	Kyösti Vääräniemi: Master of Science Thesis
2016:	Kim Kurki: Master of Science Thesis
2015:	Tatu Salmivirta: Master of Science Thesis
2015:	Sonja Saviranta: Master of Science Thesis
2012:	Pekka Raukola: Master of Science Thesis

Independent courses

2012:	Optimal Solutions in Water Systems	Tallinn Univ. of Technology
2011:	Urban Water Systems: Interactions and integrating modeling, planning and management	Tallinn Univ. of Technology

Research activities

2016–:	Cybersecurity of critical infrastructure
2015–:	Online operational optimization
2014–:	Online network modeling
2012–:	Pump energy modeling and optimization
2011–:	Water supply and distribution system and sewerage energy analysis
2011–2013:	Locating leaks and closed valves in networks
2010–:	Modeling Water supply and distribution systems and sewerage and their control systems

ELULOOKIRJELDUS

Isikuandmed

Nimi:	Markus Ilmari Sunela
Sünniaeg:	Detsember 30, 1985
Sünnikoht:	Kurikka, Soome
Kodakondsus:	Soomlane

Kontaktandmed

Aadress:	Rotkonraitti 6 B 13, 33720 TAMPERE, SOOME
Telefon:	+385 40 768 5165
E-post:	markus.sunela@iki.fi

Hariduskäik

2012–2017:	Tallinna Tehnikaülikool	doktor, ehitus ja keskkonnatehnika
2008–2010:	Tampere Tehnikaülikool	magister, vee- ja jäätmehaldustehnoloogia
2008–2009:	Tallinna Tehnikaülikool	magister, vahetusõppe
2005–2008:	Tampere Tehnikaülikool	bakalaureus, vee- ja jäätmehaldustehnoloogia
2001–2004:	Kurikka gümnaasium	keskharidus

Keelteoskus

Soome:	emakeel
Inglise:	kõrgtase
Eesti:	kõrgtase
Rootsi:	kõrgtase
Vene:	kesktase
Esperanto:	kõrgtase

Teenistuskäik

2017– :	Fluidit Ltd.	tehnoloogiajuht, nõukogu esimees
2016– :	Tampere Tehnikaülikool, Lausandmetöötluse lab.	külalisteadlane
2015– :	Tampere Tehnikaülikool, Keemia ja biotehnika lab.	külalisõpetaja
2014– 2017:	FCG Design and Engineering Ltd.	meeskonna juht
2011– 2014:	FCG Design and Engineering Ltd.	vanem spetsialist
2010– 2011:	FCG Finnish Consulting Group Ltd.	projektiinsener
2006– 2010:	Norfello Inc.	tarkvara arendaja
2005:	Soome kaitsevägi, Läänepiirkonna peakorter	süsteemiarhitekt
2005:	Soome kaitsevägi, Maaväe akadeemia	teadustöö assistent
2004:	Soome Maa-amet	maamõõtja

Tunnustused & sertifikaatid

2017:	Parim üliõpilase artikkel ja ettekanne, CCWI 2017
2016:	Aasta noor konsultant, finalist
2013:	FEANI EUR ING litsents # 32440
2010:	SNIL stipendium diplominseneridele
2010:	Magistrikraad *cum laude*
2008:	Bakalaureusekraadi *cum laude*

Kaasajuhendatud lõputööd

2016:	Kyösti Vääräniemi: magistritöö
2016:	Kim Kurki: magistritöö
2015:	Tatu Salmivirta: magistritöö
2015:	Sonja Saviranta: magistritöö
2012:	Pekka Raukola: magistritöö

Täiendusõpe

2012:	Optimal Solutions in Water Systems	Tallinn Tehnikaülikool
2011:	Urban Water Systems: Interactions and integrating modeling, planning and management	Tallinn Tehnikaülikool

Teadustegevus

2016–:	Kriitilise infrastruktuuri küberturvalisus
2015–:	Veevõrkude juhtimise reaalajas optimeerimine
2014–:	Veevõrkude reaalajas modelleerimine
2012–:	Pumbade energiatarbimise modelleerimine ja optimeerimine
2011–:	Veevarustus- ja reoveesüsteemide energiatarbimise analüüs
2011–2013:	Lekete ja suletud klappide otsing veevõrkudes
2010–:	Veevarustus- ja reoveesüsteemide ning nende juhtimissüsteemide modelleerimine

DISSERTATIONS DEFENDED AT TALLINN UNIVERSITY OF TECHNOLOGY ON *CIVIL ENGINEERING*

1. **Heino Mölder**. Cycle of Investigations to Improve the Efficiency and Reliability of Activated Sludge Process in Sewage Treatment Plants. 1992.

2. **Stellian Grabko**. Structure and Properties of Oil-Shale Portland Cement Concrete. 1993.

3. **Kent Arvidsson**. Analysis of Interacting Systems of Shear Walls, Coupled Shear Walls and Frames in Multi-Storey Buildings. 1996.

4. **Andrus Aavik**. Methodical Basis for the Evaluation of Pavement Structural Strength in Estonian Pavement Management System (EPMS). 2003.

5. **Priit Vilba**. Unstiffened Welded Thin-Walled Metal Girder under Uniform Loading. 2003.

6. **Irene Lill.** Evaluation of Labour Management Strategies in Construction. 2004.

7. **Juhan Idnurm**. Discrete Analysis of Cable-Supported Bridges. 2004.

8. **Arvo Iital**. Monitoring of Surface Water Quality in Small Agricultural Watersheds. Methodology and Optimization of monitoring Network. 2005.

9. **Liis Sipelgas**. Application of Satellite Data for Monitoring the Marine Environment. 2006.

10. **Ott Koppel**. Infrastruktuuri arvestus vertikaalselt integreeritud raudtee-ettevõtja korral: hinnakujunduse aspekt (Eesti peamise raudtee-ettevõtja näitel). 2006.

11. **Targo Kalamees**. Hygrothermal Criteria for Design and Simulation of Buildings. 2006.

12. **Raido Puust**. Probabilistic Leak Detection in Pipe Networks Using the SCEM-UA Algorithm. 2007.

13. **Sergei Zub**. Combined Treatment of Sulfate-Rich Molasses Wastewater from Yeast Industry. Technology Optimization. 2007.

14. **Alvina Reihan**. Analysis of Long-Term River Runoff Trends and Climate Change Impact on Water Resources in Estonia. 2008.

15. **Ain Valdmann**. On the Coastal Zone Management of the City of Tallinn under Natural and Anthropogenic Pressure. 2008.

16. **Ira Didenkulova**. Long Wave Dynamics in the Coastal Zone. 2008.

17. **Alvar Toode**. DHW Consumption, Consumption Profiles and Their Influence on Dimensioning of a District Heating Network. 2008.

18. **Annely Kuu**. Biological Diversity of Agricultural Soils in Estonia. 2008.

19. **Andres Tolli**. Hiina konteinerveod läbi Eesti Venemaale ja Hiinasse tagasisaadetavate tühjade konteinerite arvu vähendamise võimalused. 2008.

20. **Heiki Onton**. Investigation of the Causes of Deterioration of Old Reinforced Concrete Constructions and Possibilities of Their Restoration. 2008.

21. **Harri Moora**. Life Cycle Assessment as a Decision Support Tool for System optimisation – the Case of Waste Management in Estonia. 2009.

22. **Andres Kask**. Lithohydrodynamic Processes in the Tallinn Bay Area. 2009.

23. **Loreta Kelpšaitė**. Changing Properties of Wind Waves and Vessel Wakes on the Eastern Coast of the Baltic Sea. 2009.

24. **Dmitry Kurennoy.** Analysis of the Properties of Fast Ferry Wakes in the Context of Coastal Management. 2009.

25. **Egon Kivi**. Structural Behavior of Cable-Stayed Suspension Bridge Structure. 2009.

26. **Madis Ratassepp**. Wave Scattering at Discontinuities in Plates and Pipes. 2010.

27. **Tiia Pedusaar**. Management of Lake Ülemiste, a Drinking Water Reservoir. 2010.

28. **Karin Pachel**. Water Resources, Sustainable Use and Integrated Management in Estonia. 2010.

29. **Andrus Räämet**. Spatio-Temporal Variability of the Baltic Sea Wave Fields. 2010.

30. **Alar Just**. Structural Fire Design of Timber Frame Assemblies Insulated by Glass Wool and Covered by Gypsum Plasterboards. 2010.

31. **Toomas Liiv**. Experimental Analysis of Boundary Layer Dynamics in Plunging Breaking Wave. 2011.

32. **Martti Kiisa**. Discrete Analysis of Single-Pylon Suspension Bridges. 2011.

33. **Ivar Annus**. Development of Accelerating Pipe Flow Starting from Rest. 2011.

34. **Emlyn D. Q. Witt**. Risk Transfer and Construction Project Delivery Efficiency – Implications for Public Private Partnerships. 2012.

35. **Oxana Kurkina**. Nonlinear Dynamics of Internal Gravity Waves in Shallow Seas. 2012.

36. **Allan Hani**. Investigation of Energy Efficiency in Buildings and HVAC Systems. 2012.

37. **Tiina Hain**. Characteristics of Portland Cements for Sulfate and Weather Resistant Concrete. 2012.

38. **Dmitri Loginov**. Autonomous Design Systems (ADS) in HVAC Field. Synergetics-Based Approach. 2012.

39. **Kati Kõrbe Kaare**. Performance Measurement for the Road Network: Conceptual Approach and Technologies for Estonia. 2013.

40. **Viktoria Voronova**. Assessment of Environmental Impacts of Landfilling and Alternatives for Management of Municipal Solid Waste. 2013.

41. **Joonas Vaabel**. Hydraulic Power Capacity of Water Supply Systems. 2013.

42. **Inga Zaitseva-Pärnaste**. Wave Climate and its Decadal Changes in the Baltic Sea Derived from Visual Observations. 2013.

43. **Bert Viikmäe**. Optimising Fairways in the Gulf of Finland Using Patterns of Surface Currents. 2014.

44. **Raili Niine**. Population Equivalence Based Discharge Criteria of Wastewater Treatment Plants in Estonia. 2014.

45. **Marika Eik**. Orientation of Short Steel Fibers in Cocrete. Measuring and Modelling. 2014.

46. **Maija Viška**. Sediment Transport Patterns Along the Eastern Coasts of the Baltic Sea. 2014.

47. **Jana Põldnurk**. Integrated Economic and Environmental Impact Assessment and Optimisation of the Municipal Waste Management Model in Rural Area by Case of Harju County Municipalities in Estonia. 2014.

48. **Nicole Delpeche-Ellmann**. Circulation Patterns in the Gulf of Finland Applied to Environmental Management of Marine Protected Areas. 2014.

49. **Andrea Giudici**. Quantification of Spontaneous Current-Induced Patch Formation in the Marine Surface Layer. 2015.

50. **Tiina Nuuter**. Comparison of Housing Market Sustainability in European Countries Based on Multiple Criteria Assessment. 2015.

51. **Erkki Seinre**. Quantification of Environmental and Economic Impacts in Building Sustainability Assessment. 2015.

52. **Artem Rodin**. Propagation and Run-up of Nonlinear Solitary Surface Waves in Shallow Seas and Coastal Areas. 2015.

53. **Kaspar Lasn**. Evaluation of Stiffness and Damage of Laminar Composites. 2015.

54. **Margus Koor**. Water Distribution System Modelling and Pumping Optimization Based on Real Network of Tallinn. 2015.

55. **Mikk Maivel**. Heating System Efficiency Aspects in Low-Energy Residential Buildings. 2015.

56. **Kalle Kuusk**. Integrated Cost-Optimal Renovation of Apartment Buildings toward Nearly Zero-Energy Buildings. 2015.

57. **Endrik Arumägi**. Renovation of Historic Wooden Apartment Buildings. 2015.

58. **Tarvo Niine**. New Approach to Logistics Education with Emphasis to Engineering Competences. 2015.

59. **Martin Thalfeldt**. Total Economy of Energy-Efficient Office Building Facades in a Cold Climate. 2016.

60. **Aare Kuusik**. Intensifying Landfill Wastewater and Biodegradable Waste Treatment in Estonia. 2016.

61. **Mart Hiob**. The Shifting Paradigm of Spatial Planning in Estonia: The Rise of Neighbourhood Participation and Conservation of Built-up Areas through the Detailed Case Study of Supilinn, a Historic Suburb of Tartu City, Estonia. 2016.

62. **Martin Heinvee**. The Rapid Prediction of Grounding Behavior of Double Bottom Tankers. 2016.

63. **Bharat Maharjan**. Stormwater Quantity and Quality of Large Urban Catchment in Tallinn. 2016.

64. **Nele Nutt**. The Restoration of Nationally Protected Estonian Manor Parks in the Light of the Florence Charter. 2017.

65. **Üllar Alev**. Renovation and Energy Performance Improvement of Estonian Wooden Rural Houses. 2017.

66. **Simo Ilomets**. Renovation Need and Performance of Envelopes of Concrete Apartment Buildings in Estonia. 2017.

67. **Argo Kuusik**. Determining Biogas Yield from Industrial Biodegradable Waste. 2017.

68. **Katri Pindsoo**. Spatio-Temporal Changes in the Components of Extreme Water Levels on Estonian Coasts. 2017.