



CHALMERS
UNIVERSITY OF TECHNOLOGY



Optimizing Water Tank Levels Using Genetic Algorithms

Master's thesis in Sustainable Energy Systems

ALVAR WIKSTRÖM

DEPARTMENT OF ARCHITECTURE AND CIVIL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2025

www.chalmers.se

MASTER'S THESIS 2025: ACEX30
Master's Thesis in Sustainable Energy Systems
Supervisors: Behroz Haidarian and Glen Nivert
Examiner: Thomas Pettersson



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Architecture and Civil Engineering

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2025

Optimizing Water Tank Levels Using Genetic Algorithms

ALVAR WIKSTRÖM

© ALVAR WIKSTRÖM, 2025.

Supervisor: Behroz Haidarian and Glen Nivert, Kretslopp och vatten, Göteborg stad

Examiner: Thomas Pettersson, Chalmers University of Technology

Degree project report 2025

Department of Architecture and Civil Engineering

Chalmers University of Technology

SE-412 96 Gothenburg

Sweden Telephone

+46 31 772 1000

Optimizing Water Tank Levels Using Genetic Algorithms

ALVAR WIKSTRÖM

Department of Architecture and Civil Engineering

Chalmers University of Technology

Abstract

This thesis presents a practical optimization framework for energy-efficient pump scheduling in water distribution systems. It combines the practicality of rule-based control with the global-search power of genetic algorithms. A novel setpoint curve encoding scheme is introduced, in which daily tank-level targets are parameterized by a small set of meaningful coefficients (baseline, peak/dip timing and amplitude, and curvature descriptors). These key parameters are then optimized using a custom genetic algorithm, coupled with EPANET-driven hydraulic simulations. Constraint handling is managed by penalty functions for demand security, hydraulic feasibility, reservoir volume balance and pump maintenance. The framework is first demonstrated on the simplified NET-1 hydraulic network, providing insight on how to algorithm operate. The optimization algorithm is subsequently applied to a calibrated high-pressure zone (HPZ-G) of the Gothenburg water network, using historical operational data for model validation. Results indicate that the optimized setpoint curves can reduce energy cost, whilst still hydraulic and operational constraints. However, certain data gaps are identified which would need to be addressed to improve the model's validity.

Keywords: Genetic Algorithms, Pump scheduling, Optimization, EPANET, Energy Efficiency, Hydraulic simulation, Python, Water Distribution System, Setpoint curve, Water Tank Level.

Acknowledgements

I would like to express my sincere gratitude to my supervisors at Kretslopp och Vatten for the opportunity to undertake this thesis and for their continuous support, inspiration, and guidance provided throughout the process. I am also thankful for providing access to data and software resources that enabled the completion of this thesis.

I am also grateful to Andreas Hermanson at Kretslopp och Vatten for his assistance and guidance in the hydraulic modelling aspects.

Finally, I would like to thank Thomas Pettersson at Chalmers University of Technology for his supervision and for serving as an examiner.

Alvar Wikström, Gothenburg, June 2025

Contents

Abbreviations	IX
List of Figures	X
List of Tables.....	XII
1. Introduction.....	1
1.1 Purpose	2
1.2 Scope.....	2
1.3 Limitations and Delimitations	2
1.3.1 Demand representation	2
1.3.2 Control approach	2
1.3.3 Optimization method	2
1.3.4 Electricity price as a “pseudo” variable	3
1.4 Water distribution networks.....	3
1.4.1 Hydraulic pressure zones	3
1.4.2 Water tanks and reservoirs	4
1.4.3 Pressure levels.....	5
1.5 Demand curves/patterns	5
1.5.1 Water demand	5
1.5.2 Energy demand/production	6
2. Literature study.....	7
2.1 Optimization.....	7
2.1.1 Deterministic models	7
2.1.2 Metaheuristic models	7
2.2 Genetic Algorithms	8
2.2.1 Structure.....	8
2.2.2 Population and Initialization	9
2.2.3 Encoding.....	9
2.2.4 Fitness	9
2.2.5 Operators	9
2.3 Objective function.....	11
2.4 Decision variables	12
2.4.1 Pump status.....	12
2.4.2 Tank Trigger levels.....	12
2.5 Constraints	12
2.5.1 Constraints vs penalties	12
2.6 The use of hydraulic models	13
2.7 Real world implementation	13
2.8 Trigger level control	14

2.9	Setpoint-curve control	16
3.	Method.....	19
3.1	Software	20
3.1.1	EPANET	20
3.1.2	EPyT	20
3.1.3	DEAP	21
3.1.4	MIKE+	21
3.2	Hydraulic models	21
3.2.1	NET-1.....	21
3.2.2	HPZ-G.....	22
3.3	Validation of the HPZ-G hydraulic model.....	22
3.3.1	Pump curve calibrations.....	22
3.3.2	Hydraulic simulation validation.....	23
3.4	Timesteps.....	23
3.4.1	Hydraulic timestep	23
3.4.2	Reporting timestep.....	23
3.4.3	Rule timestep	23
3.4.4	Interpolation interval timestep	23
3.5	Objective function.....	24
3.5.1	Energy cost	24
3.5.2	Constraints	24
3.6	The Optimization algorithm	27
3.6.1	Key-parameter encoding.....	27
3.6.2	Pump trigger ranges variable	29
3.6.3	Mutation scheme	30
3.6.4	Initialization Scheme	32
3.6.5	Crossover Scheme	32
3.6.6	Elitism Strategy.....	33
3.6.7	Selection strategy.....	33
3.6.8	Parallelization	33
3.6.9	Hyper-parameter strategy.....	34
3.7	Scenario evaluations	34
3.7.1	NET-1.....	34
3.7.2	HPZ-G.....	35
4.	Data	36
4.1	Electricity price.....	36
4.2	Suction pressure from the general WDS.....	36
4.3	Hyperparameters	36

5.	Results	38
5.1	Validating the HPZ-G hydraulic model	38
5.1.1	Pump curve calibrations	38
5.1.2	Simulating current control scheme	40
5.1.3	Pump station 2	42
5.2	Net1.....	42
5.2.1	Parallelization	42
5.2.2	Simple optimization scenario	43
5.3	HPZ-G	44
5.3.1	Timestep Sensitivity Analysis.....	44
5.3.2	Main HPZ-G optimization results	46
5.3.3	Scenario 2022-11-07	46
5.3.4	Scenario 2022-11-13	47
6.	Discussion	49
6.1	Control scheme/Real network	49
6.1.1	Validity of hydraulic model	49
6.1.2	Validity of Energy data	49
6.1.3	Objective function development.....	49
6.2	Optimization algorithm	49
6.2.1	Shortcoming of the algorithm	49
6.2.2	Pump trigger ranges	50
6.2.3	Timestep variables	50
6.2.4	Parallelization	50
7.	Conclusion and further development	51
7.1	Conclusion	51
7.2	Further improvement for optimization model.....	51
7.3	Recommendation of the current control scheme.....	52
	References	XI
	Appendix	XIII
	A - Electricity prices	XIII
	B - Water Demand data	XIV
	C – General HPZ-G Run hyperparameters.....	XV
	D – PS-2 Results from validation of hydraulic model 2022-11-07.....	XVI
	E – Truncated Fourier series encoding.....	XVIII

Abbreviations

GA	Genetic Algorithm
HPZ	High Pressure Zone
TLC	Trigger Level Control
FSP	Fixed Speed Pumps
WDS	Water Distribution System
WST	Water Storage Tank
mWC	Meter Water Column
SCADA	Supervisory Control and Data Acquisition
KoV	Kretslopp och vatten, Göteborg stad
PTR	Pump Trigger Ranges

List of Figures

Figure 1-1 Illustration of simple boosted HPZ.....	3
Figure 1-2 Topographic map of Gothenburg.....	4
Figure 1-3 Typical residential are-based demand consumption pattern.	5
Figure 2-1 Overview of optimization schemes relevant for water pump optimization.....	8
Figure 2-2 Illustration of basic genetic algorithm terminology.....	9
Figure 2-3 Example of a simple single-point crossover operation.	10
Figure 2-4 Example of a simple mutation operation.....	10
Figure 2-5 Illustration of a tournament-based selection process.	11
Figure 2-6 Visualization of the solution space in a constrained optimization problem.....	13
Figure 2-7 Illustration of FTLs scheme, Adapted from (Quintiliani & Creaco, 2019)	14
Figure 2-8 Illustration of RFTLs scheme, Adapted from (Quintiliani & Creaco, 2019)	15
Figure 2-9 Illustration of VTLs scheme, Adapted from (Quintiliani & Creaco, 2019)	15
Figure 2-10 Illustration of RFLATs scheme, Adapted from (Quintiliani & Creaco, 2019)	16
Figure 2-11 Simple setpoint curve illustration.	16
Figure 2-12 Illustration of a set-point curve with individual pump-specific threshold ranges.	17
Figure 3-1 Flow diagram of complete optimization scheme.....	20
Figure 3-2 Hydraulic representation of the NET-1 network.....	21
Figure 3-3 Simplified schematic overview of the studied HPZ, referred to a “HPZ-G”	22
Figure 3-4 Illustration of the effect Interpolation interval timestep.....	24
Figure 3-5 Schematic overview of EPANET error/warning handling.....	26
Figure 3-6 Schematic overview of the encoding–decoding process used in the genetic algorithm.....	27
Figure 3-7 Visual representation of the key-parameters.	28
Figure 3-8 Example of pump threshold ranges (pt_var = 1).....	29
Figure 3-9 Pump threshold ranges after scaling by pt_var = 0.5.....	29
Figure 3-10 setpoint curve vs actual reservoir level, real-world data from 2022-11-07	30
Figure 3-11 Visual representation of the Peak-Dip separation constraint.....	32
Figure 3-12 Visual representation of the block-based crossover scheme.	32
Figure 3-13 Schematic overview of the parallel evaluation process using a “master-slave” method..	33
Figure 3-14 Tariff periods and initial (unoptimized) setpoint curve	34
Figure 5-1 Operating points for pumps in Pump Station 1 for the HPZ-G network.....	38
Figure 5-2 Recreated pump curves based on actual operating points in Pump Station 1	39
Figure 5-3 Operating points for pumps in Pump Station 1 for the HPZ-G network.....	39
Figure 5-4 Recreated pump curves based on actual operating points in pump station 2	40
Figure 5-5 Actual WTL measured in the simulation and from real-world data.....	40
Figure 5-6 Outgoing flow from PS-1 measured in the simulation and from real-operation data.....	41
Figure 5-7 Outgoing pressure from PS-1 measured in the simulation and from real-world data.	41
Figure 5-8 Active pumps in the simulation and during the real-world scenario.....	42
Figure 5-9 Runtime as a function of the number of processes used for parallel evaluation.	42
Figure 5-10 Evolution of reservoir level set-point curves over nine generations.	43
Figure 5-11 Comparison of initial and optimized set-point curves	44
Figure 5-12 Impact of reporting timestep on average runtime and energy cost.....	44
Figure 5-13 Impact of hydraulic timestep on average runtime and accuracy	45
Figure 5-14 Impact of interpolation timestep on average runtime and energy cost.....	45
Figure 5-15 Optimized reservoir level and base solution for the 24-hour time period: 2022-11-07....	46

Figure 5-16 Optimized reservoir level and base solution for the 24-hour time period: 2022-11-13.... 47
Figure B-1 Water demand per hour for 2022-11-07 XIV
Figure B-2 Water demand per hour for 2022-11-13 XIV
Figure D-1 Outgoing flow from PS-2 measured in the simulation and from real-operation data. XVI
Figure D-2 Outgoing flow from PS-2 measured in the simulation and from real-operation data. XVI
Figure D-3 Active pumps in the simulation and during the real-world scenario XVII

List of Tables

Table 2-1 Start and stop thresholds for pumps relative to the set-point curve	17
Table 3-1 Description of variables in the key-parameter encoding scheme.....	27
Table 3-2 Clipping bound for key parameters.....	31
Table 3-3 Timestep variables for "baseline scenario"	35
Table 4-1 General GA Settings.....	36
Table 4-2 Penalty variables	36
Table 4-3 Crossover weighting variables.....	37
Table 4-4 Timestep variables.....	37
Table 4-5 Mutation probability variables	37
Table 4-6 Mutation Clip bound variables	37
Table 5-1 Summary potential cost savings during the evaluated timeframe.	46
Table 5-2 Key-parameter encoded genotype for optimized setpoint curve.....	47
Table 5-3 Key-parameter encoded genotype for optimized setpoint curve.....	48
Table A-1 Electricity prices for 2022-11-07	XIII
Table A-2 Electricity price for 2022-11-13.....	XIII
Table C-1 Hyperparameters for general HPZ-G run.	XV

1. Introduction

The water distribution system (WDS) constitutes a vital component of critical drinking water infrastructure. Such systems inherently require high reliability due to clean water's foundational role in modern society. For most drinking water systems, raw and drinking water pumping constitutes most of the energy use (Tarek et al., 2022). Conditions such as topology and consumption patterns can have great effects on the required infrastructure and operating cost. Recent times also seem to suggest that prices and tariffs for electricity are on the rise (Hagman et al., 2024). The energy system is also turning more electrified and reliant on variable renewable energy sources. The demand for flexible, smart energy consumption is increasing. Recent periods of fluctuating prices of electricity have also further increased interest in consumption flexibility in WDS. It is important to note that the main objective of water distribution is quality and reliability.

A great deal of research has gone into optimizing the operation of water distribution to minimize its operational cost. The methodology of these studies can roughly be divided into rule-based control schemes and optimization algorithm-based methods (Giustolisi et al., 2013). A rule-based control scheme essentially means to use fixed control rules to control pumps. For example, a pump turns on when the water tank level is low, and off when it is full. It is a simple method to implement, and it responds in real time to current system status. It does however have limited optimization opportunities. Optimization-based methods use advanced algorithms to find the optimal schedule for operating pumps, often considering several different objectives. These objectives usually include energy cost, maintenance cost and reliability. These methods have the potential to provide more flexible and cost-effective scheduling compared to simple rule-based controls. These models are however: more complex, harder to implement and require accurate predictions of future demand, electricity tariffs and system response (Salomons & Housh, 2020).

This thesis aims to incorporate different aspects of both methods. The idea of a rule-based control optimization has been put into use before. For example, Marchi et al., (2017) implement and optimize flexible rule-based pump controls by including multiple conditions. These conditions were for example: tank levels, time-of-day, and pump status. This enhanced control logic is coupled with a genetic algorithm that tunes rule parameters to minimize energy costs while maintaining hydraulic stability in the system. Alvisi & Franchini, (2017) propose a trigger level-based strategy for pump control. Using a multi-objective genetic algorithm, they optimize these trigger levels to balance energy cost and the number of pump switches. While having some similarities with the approaches discussed in these articles, this thesis offers an alternative way to constrict and traverse the solution space. The optimization method proposed in this thesis is based on a setpoint curve control scheme. In the field of control theory, a setpoint curve refers to a predefined time-varying target value which a control system aims to maintain. Which in this specific scenario would be the level in the water tank, controlled by switching pumps on/off. This control scheme is described in detail in section 2.10.

The main novelty of this thesis is the setpoint curve specific optimization approach. In this optimization scheme the set-point curve is encoded by a few key parameters. New solutions are then generated through mutating these key-parameters which provide meaningful changes of the set-point curve. These changes are introduced based on a GA based optimization scheme. An extensive description of the optimization scheme is available in the method section.

The genetic algorithm is implemented using the DEAP evolutionary framework (Fortin et al., 2012), and candidate control curves are evaluated through EPANET simulation. The core idea of optimization is to reduce energy cost whilst respecting operational constraints such as reservoir bounds, pressure requirements, pump switching and demand security.

1.1 Purpose

The purpose of this thesis is to develop and evaluate an optimization framework for energy-efficient pump scheduling in water storage tank-based water distribution network. This approach aims to strike a practical balance between rule-based, interpretable control strategies and modern optimization techniques.

The proposed control scheme will be validated through simulation on a real-world water distribution network that has been calibrated using historical operational data. Performance will be assessed by comparing the optimized solutions generated by the genetic algorithm with the existing control strategy. This comparison will highlight potential improvements in energy efficiency, constraint handling, and adaptability introduced by the optimization-based approach.

1.2 Scope

The optimization framework is applied to a hydraulically isolated high-pressure zone within a larger water distribution system. This delimitation was made to reduce model complexity, limit simulation runtimes, and improve calibration accuracy given the available operational data. More broadly, raw water pumping and the energy required for water treatment are excluded from the analysis. As a result, the optimization focuses exclusively on a specific part of the clean water distribution stage, where electricity costs with pump operations are most significant and directly controllable.

1.3 Limitations and Delimitations

1.3.1 Demand representation

The consumption patterns used for simulations and evaluation are derived from historical operating data from a limited time period. As such the optimization results will be true for the investigated time period, but the result could vary significantly for time periods with different consumption patterns.

1.3.2 Control approach

This thesis deliberately focuses on optimizing a smooth set-point curve that implicitly controls when pumps turn on/off rather than performing explicit pump scheduling. This choice was mainly motivated by the desire to create an optimization scheme, fully compatible with the current control scheme. The set-point curve approach also enables a smoother incorporation of domain knowledge, used to generate a more efficient and targeted optimization process. More explicit pump scheduling has the potential of increased energy savings (Quintiliani & Creaco, 2019). However, it is generally harder to apply in practice as it substantially increases the complexity of the application. It often depends heavily on accurate short-term demand forecasts, which are not always available or reliable in practice (Salomons & Housh, 2020).

1.3.3 Optimization method

A wide range of optimization methods are relevant to pump scheduling, including linear programming based, mixed-integer models, and metaheuristics approaches. This thesis focuses on Genetic Algorithms (GAs), motivated by the demonstrated effectiveness in existing pump scheduling studies (Mala-Jetmarova et al., 2017). GAs is particularly well-suited for non-linear, non-convex, and

constraint-heavy problems (Nicklow et al., 2010), characteristics that are typical of real-world water systems (Marchi et al., 2017). The traditional mutation and crossover operators also synergies well with the encoding scheme utilized in this thesis. These operators are described in more detail in Chapter 2.

1.3.4 Electricity price as a “pseudo” variable

In this thesis, the electricity price is treated as a simplified proxy for the wider, more complex electricity tariff components. Rather than modeling the detailed structure of grid fees, capacity charges, power-based demand tariffs, and other contractual elements, the optimization scheme considers only the unit energy price. This was done partly since the specific tariff scheme paid by Kretslopp och vatten, Göteborg stad was not a central part of the research question, and the main objective was to create an optimization algorithm adapted for the specific control system. Focusing only on the energy price still captures the main operation incentives: shifting pump usage from high price/demand periods to reduce cost and grid load.

1.4 Water distribution networks

1.4.1 Hydraulic pressure zones

A hydraulic pressure zone (HPZ) can be defined as delimited part of the WDN for which obtains a certain pressure gradient (Salomons & Housh, 2020). These zones are thus connected to the general WDN through some type of pressure control device, such as a pumping station or a pressure reducing valve. HPZs are established based on elevation differences in the WDN, higher elevations require more pumping to reach desired pressure, and lower elevation might require Pressure reducing valves to not reach too high-pressure levels. In figure 1-1 a simple illustration of a boosted HPZ can be seen.

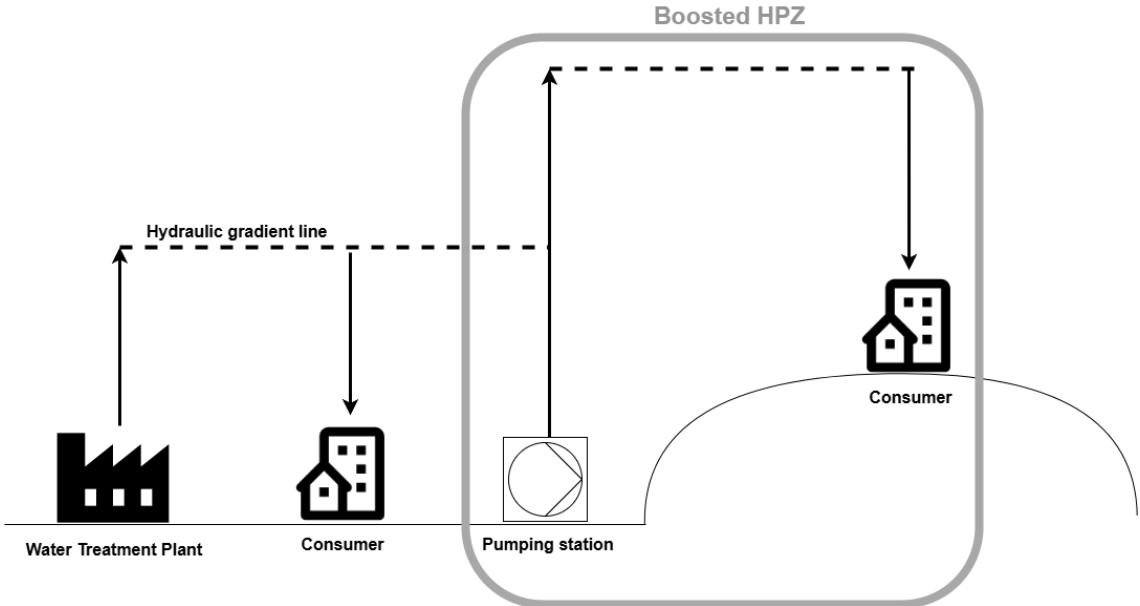


Figure 1-1 Illustration of simple boosted HPZ

The dotted line represents the Hydraulic Gradient Line (HGL), it’s a common concept in fluid dynamics and represents the pressure head and the elevation head. The figure presents an idealized

case that neglects friction losses, which would otherwise make the line more slanted. Larger WDN can contain multiple HPZs, the amount is mainly determined by topological conditions. Figure 1-2 presents an elevation map of the Gothenburg region. The distinct topographical variation clearly indicates the necessity of high-pressure zones in certain areas to maintain suitable pressure levels.

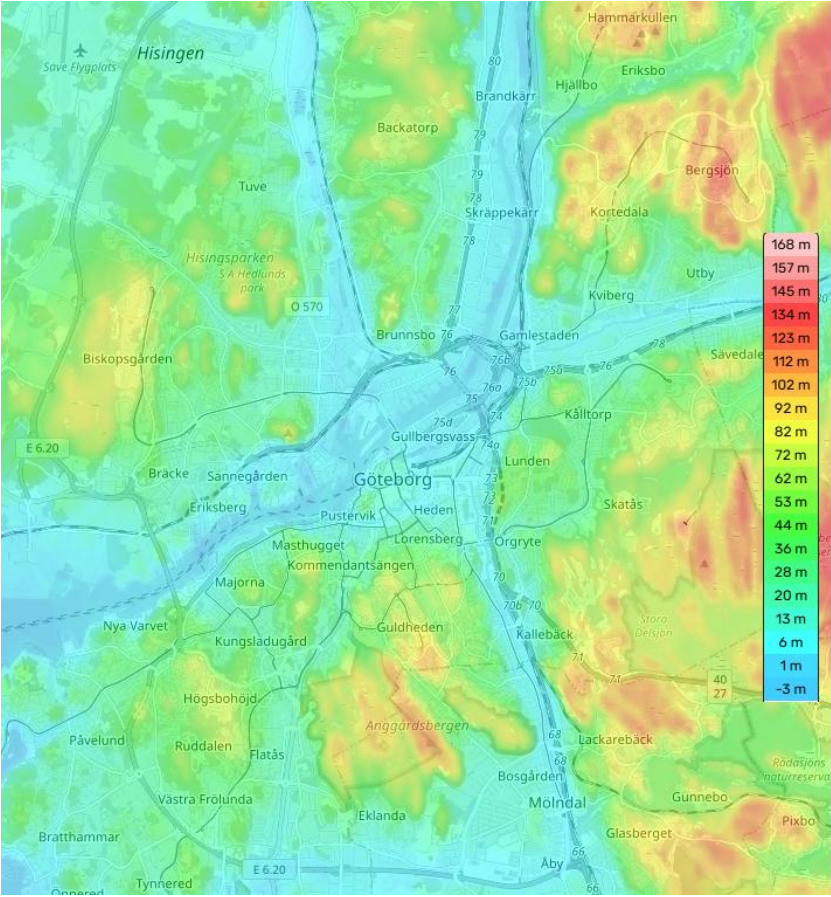


Figure 1-2 Topographic map of Gothenburg generated from TessaDEM data (v1.2) (TessaDEM, 2025)

A large amount of HPZs is operated with a Water Storage tank (WST), which offer more flexibility and reliability. HPZ can also be operated without storage, directly via pumping stations. These zones are referred to as “rigid” and are in most cases operated with variable speed pumps (VSP).

1.4.2 Water tanks and reservoirs

In the context of water distribution networks, the terms tank and reservoirs are often used quite interchangeably. However, in the context of simulation software (such as EPANET), a distinct difference exists between the two terms. A reservoir represents an infinite source or sink with a specified hydraulic head. The hydraulic head can be fixed or specified by a predetermined pattern. The flow from/to the reservoir is essentially determined by the pressure difference between the reservoir and the connected network. A tank on the other hand represents a finite-volume storage, where to volume depends on the inflow/outflow. The pressure head will thus calculate at every timestep, based on the tank volume. In this thesis the terms are used interchangeably.

The main purposes of WST are to provide increased reliability and flexibility. This is done mainly by storing water during low demand periods and supply it when demand is high. This cycling is usually

done on a daily time horizon, e.g. filling reservoir at night and emitting most of it during the day. Without WST, pumps in HPZ will have to run all the time, constantly adjusting the output based on the current demand. A WST enables limited and stable pump operation during high demand periods. WST also enable pumping to complete stop, while still maintaining pressure in the network. This is especially useful during maintenance and in case of pump failure.

1.4.3 Pressure levels

Correct management of high-pressure zones (HPZ) minimizes excessive pressure levels, thereby reducing pumping costs and lowering the risk of leakages and pipe bursts (Creaco et al., 2019). At the same time, it is essential to always ensure adequate pressure on customer nodes. Industry standards recommend a minimum pressure of 15 meters of water column (mWC) above the highest outlet at the connection site, while the maximum recommended pressure is 70 mWC (Svenskt Vatten, 2021).

1.5 Demand curves/patterns

In order to optimize pump scheduling a certain understanding of how the water demand varies over time is needed. In terms of energy optimization, it is also important to have a basic understanding of energy demand and production patterns.

1.5.1 Water demand

Consumption will vary based on time of year and temperature/weather, but the general shape of the consumption tends to stay the same. The type of consumer will also affect the shape of the demand. For example, an industry’s water consumption generally increases more during the day and can have less predictable patterns due to different water intensive processes. In this thesis the focus will be on a more residential based demand curve as this is the main type of consumer in investigated HPZ and is also representative of most of the demand in the Gothenburg Water distribution network.

In figure 1-3 a water consumption pattern can be seen for a typical weekday. Two demand peaks can be seen during the morning and afternoon, whilst a very low consumption can be seen during the night. During weekends, water consumption typically decreases and is slightly delayed.

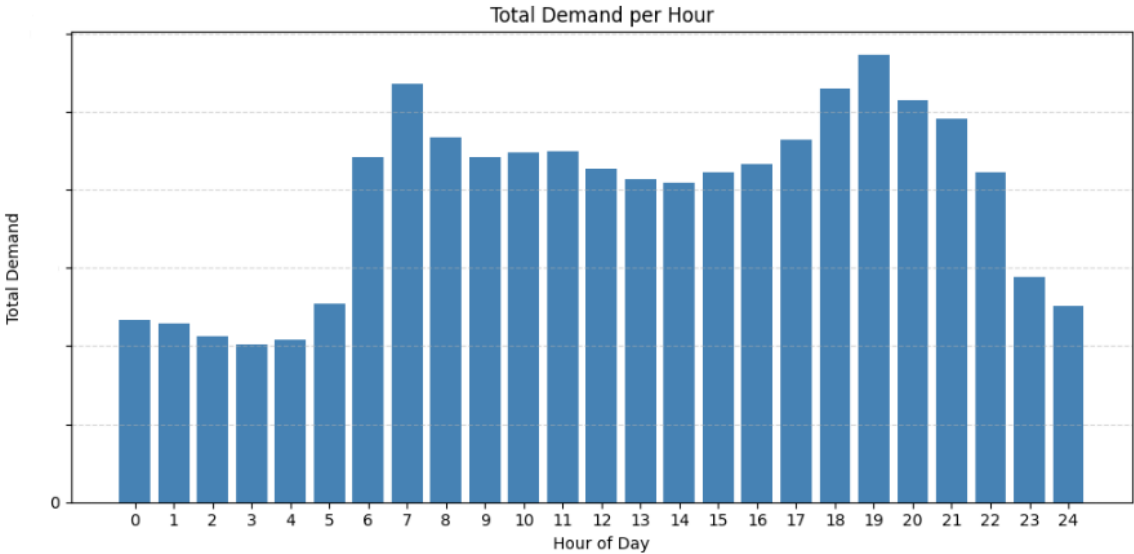


Figure 1-3 Typical residential are-based demand consumption pattern.

1.5.2 Energy demand/production

Electricity price tends to be a good surrogate for capturing both energy demand and production. The main pricing mechanism in Sweden is the day-ahead market run by Nord Pool (Hagman et al., 2024). Hourly prices are set the day before based on bidding by electricity producers and retailers. Even if a municipal electricity consumer often holds more fixed-price based electricity contract, there is still merit to considering real-time electricity prices. For example, grid fees, capacity charges, and power-based tariffs are generally not affected by a fixed electricity price. Also, by aligning operation with lower-priced periods, these types of consumers could assist in offloading the grid during high consumption periods, thus improving grid stability and efficiency (Hagman et al., 2024).

2. Literature study

2.1 Optimization

An optimization algorithm essentially is a method used to find the best solution from a set of potential solutions. What constitutes a best solution is based on single or multiple pre-determined objectives. When considering the choice of optimization method, it can be good to consider the “No Free Lunch Theorem”. This theorem essentially states that there is no overall superior optimization algorithm that can solve all problems, and that a method efficient for a specific set of problems will be outperformed by other algorithms on a different set of problems (Kramer, 2017). At its core, optimization is about finding the most suitable method for the problem at hand and adjusting the method to obtain a solution as efficiently and accurately as possible. In the context of WDS optimization, the optimization methods can generally be divided into two different categories: Deterministic model and heuristic/metaheuristic models.

2.1.1 Deterministic models

Deterministic models follow a systematic predictable process to determine results. For any given output the result from the model will always be the same. Examples of deterministic methods: Linear programming (LP), nonlinear programming (NLP), Mixed-Integer Linear Programming (MILP). These models are reliable, provide mathematically precise solutions and are efficient for well defined, structured problems (Sivanandam & Deepa, 2008). The drawback can however be that deterministic methods can struggle with large or complex networks due to computational limitations, and for non-convex problems they can end up in local optima (Balekelayi & Tesfamariam, 2017). These models are also arguably less suited of handling multi objective functions as the ability to manage complex trade-offs and explore diverse solutions is limited.

2.1.2 Metaheuristic models

Technologic advancements improving the efficiency of hydraulic simulations and computational speed in general, has made non-deterministic metaheuristic models more attractive (Mala-Jetmarova et al., 2017). These methods do not depend on gradients (derivatives) when searching for the optimal solution (Balekelayi & Tesfamariam, 2017). Instead, it is a higher-level procedure which aims to find a good enough solution based on some type of general heuristic method rather than a specific mathematic formula. A common attribute for metaheuristic models, especially in the field of WDS is that they are population based. This means they maintain and evolve a set of potential solutions simultaneously. They can also be applied to large-scale water networks with high-dimensional decision spaces (Janga Reddy & Nagesh Kumar, 2020). These models can balance exploration (searching the solution space) and exploitation (refining promising solutions).

It is naturally important to take the optimization model into consideration when defining the optimization objective. Deterministic models require more strict variable bounds and relationships. Metaheuristic models can handle more “soft” constraints, conflicting objective functions. In figure 2-1 an overview of optimization methods relevant to pump scheduling optimization can be seen.

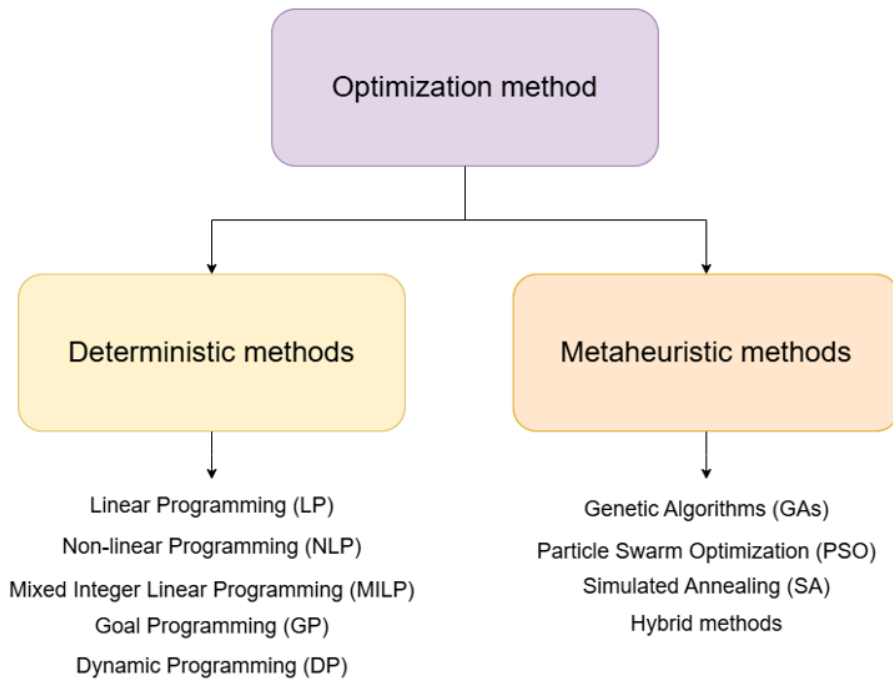


Figure 2-1 Overview of optimization schemes relevant for water pump optimization

2.2 Genetic Algorithms

Genetic algorithms are a type of metaheuristic method for solving optimization problems, inspired by the biological process of natural selection. It is part of the larger class of Evolutionary algorithms. These algorithms imitate how populations evolve over generations using mechanisms such as crossover, mutation, and selection. GA is well established in the field of pump scheduling optimization, and many different applications are available in literature (Mala-Jetmarova et al., 2017). A great advantage of GAs is that it does not rely on gradient information, making it more suitable for problems where the relationship between decision variables and the objective function are more complex.

2.2.1 Structure

In its simplest form a Genetic algorithm is built upon chromosomes which are represented by a string of binary values (1,0). These individual values are referred to as genes and can represent some type of action/operation, for example pump on/off statuses. A group of chromosomes is referred to as a population. In figure 2-2 a simple illustration of structures can be seen.

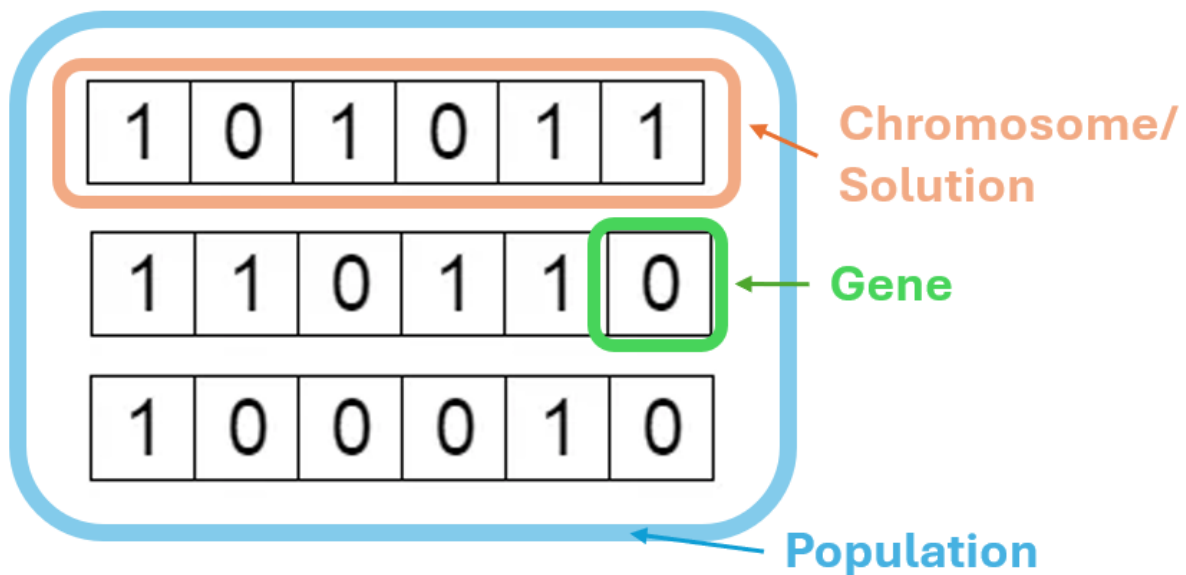


Figure 2-2 Illustration of basic genetic algorithm terminology

2.2.2 Population and Initialization

A population consists of a group of individuals, where each individual represents a solution. An important variable for GA is the size of the population. This variable will depend on the complexity of the problem and how the solution is structured (Sivanandam & Deepa, 2008). The initial population is often generated through some type of random initialization. For many problems the first population should ideally contain a wide array of solutions, to be able to capture the entire solution space. It is also possible to utilize heuristics to generate an initial population which has a higher average fitness value. This can be useful to obtain faster conversion rates but could also increase the risk of ending up in a local optimum. In the context of trigger levels, it is reasonable to use a heuristic initialization to obtain an initial solution which consists of mostly feasible solutions.

2.2.3 Encoding

For many problems, the actual solution is quite complex and needs to be discretized to be compatible with the different genetic operators. Expressing the solution as simpler data structure ensures that the model operates efficiently and accurately. Usually, the actual solution is referred to as the phenotype and the encoded representation of the solution is referred to as the genotype (Sivanandam & Deepa, 2008). To evaluate each solution, a conversion from genotype to phenotype is done as a part of the fitness function.

2.2.4 Fitness

The fitness functions essentially evaluate the quality of each solution, based on the desired objective/objectives. For multiple objective optimization the fitness function can be more complex to express. The fitness function is essentially equivalent to the objective function, which is discussed in the following section.

2.2.5 Operators

Operators are the mechanisms which determine the evolution of populations over multiple generations. In this section the general methodology for these operators will be covered, the specific operators used in the actual model and the motivation for using these are available in the method section.

Crossover

The operator which combines genetic material from two or more solution chromosomes (parents) to form a new solution (child) (Kramer, 2017). In its simplest form this can be done by a “single-point crossover” which can be seen in figure 2-3. A crossover point is selected somewhere along the chromosome. The string on each side of the point will come from different parents.

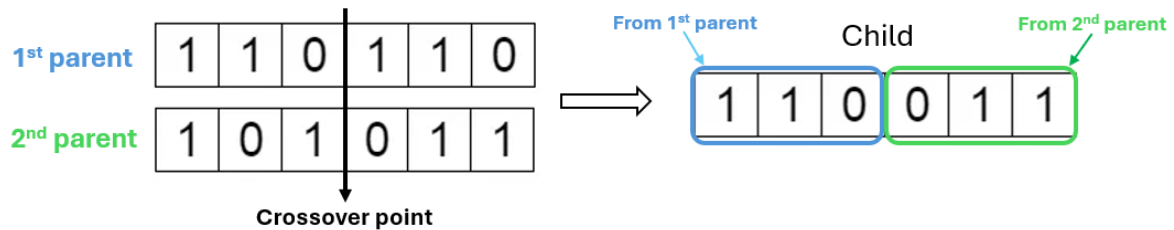


Figure 2-3 Example of a simple single-point crossover operation.

Mutation

The main purpose of mutations is to introduce genetic diversity through random changes of the populations. This in turn will prevent premature convergence and assist the algorithms in avoiding getting stuck in local optima. For a binary chromosome string a simple mutation could simply be randomly switching a gene from 0 to 1 (or vice versa). This is referred to as Bit-Flip mutation and can be seen in figure 2-4.

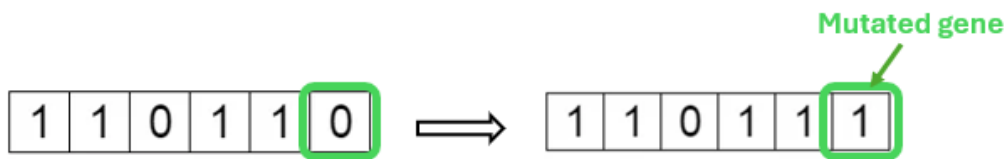


Figure 2-4 Example of a simple mutation operation.

Selection

Is the process which selects the best offspring solutions based on the previously mentioned fitness. This is essentially what decides which solutions are chosen for each new generation. Most selections employ some selection strategy based on fitness in combination with some type of randomness. A balance between exploration and exploitation is often desirable. A common selection method used is tournament selection. In this method n amount of solutions are chosen at random. The solution with the best fitness value is then selected for the next generation, whilst the others are discarded. A greater n increase exploitation. A tournament selection scheme can be seen in figure 2-5.

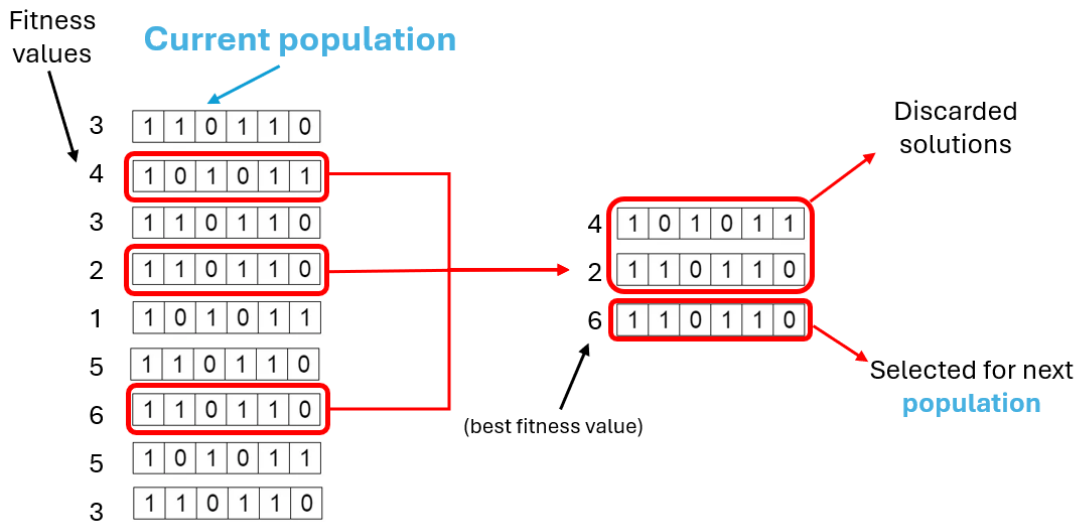


Figure 2-5 Illustration of a tournament-based selection process.

Termination

The condition for termination defines when the evolutionary algorithms should stop. In most cases the optimal solution is unknown, therefore the termination condition is rarely based on proximity to the optimal solution. Instead, the termination is often based on a predefined number of generations, or a lack of improvement from the convergence.

Heuristic choice of operators

While many of the GA operators can be applied to a range of different problems, the performance will increase drastically if operators design more specific tasks (Sivanandam & Deepa, 2008). As with any other optimization algorithm, utilizing information about the domain is an essential part of fine-tuning the model. Infeasible solutions can be discarded at an early stage of evaluation or even be not generated at all.

2.3 Objective function

The objective function is an essential part of any optimization algorithm, it essentially decides the aim of the optimization model. In the context of Pump scheduling optimization, the most common objectives are related to cost minimization, for example: minimization of Energy costs. Objective functions with multiple goals are referred to as multi-objective. Other objectives which can also be addressed are water quality constraints and security of supply. These types of objectives can be harder to quantify and is therefore often defined as constraints. Ideally all elements of an objective function should be expressed as the same unit, to obtain an optimal solution. In multi-objective optimization, the objectives can often be conflicting. If the objective function uses different units, the optimization results will be in the form of a pareto front. A pareto front represents a set of non-dominated solutions. Non-dominated essentially just means that each solution cannot be improved based on any objective, without worsening another objective (Kramer, 2017).

2.4 Decision variables

The controllable elements and actions of the system which are to be optimized by the model. These can both be in the form of binary- or continuous values.

2.4.1 Pump status

These are binary values which indicates if a pump is on or off. In this model the pump statuses are implicit variables as they are directly controlled by the trigger levels of the tanks. In this model all pumps are assumed to be Fixed Speed Pumps.

2.4.2 Tank Trigger levels

The tank trigger levels are the main controlling variable for the model. In its simplest form tank trigger levels are constant and set as the actual maximum/minimum levels of a tank (Quintiliani & Creaco, 2019). Dynamically changing trigger levels will vary throughout the day, changing the span of allowable reservoir levels.

2.5 Constraints

The limitations of the systems and requirements which must be fulfilled while performing the optimization are referred to as constraints. They can represent hydraulic constraints such as restriction of flow in pipes and mass balance in a tank. System constraints refer to limitations set by the specific system such as: max/min levels in reservoirs and required pressures for specific nodes. Operational constraints can be defined as an operation limit set by the system, for example: amount of pump switches. Certain elements of the objective function can express constraints.

2.5.1 Constraints vs penalties

In the context of optimization algorithms, particularly in Genetic algorithms, undesirable features of solutions can be handled in two main ways. One of the most basic methods is the death penalty, which essentially represent expressing something as a hard constraint. In a death penalty regime, any solution which violates any constraint is discarded (Kramer, 2017). For some problems, it can be favorable to immediately discard unfeasible solutions and not waste any resources on computing any derivatives of these undesirable solutions. An example could be violations which could lead to inaccurate results or crashes in the simulation's software. However, for a lot of problems the optimal solution will reside near the unfeasible region of solutions. As such there is a great incentive to extract information from these infeasible solutions (Sivanandam & Deepa, 2008). Penalty functions are a way to handle infeasible solutions as opposed to discarding them entirely. The fitness value is instead reduced based on the severity of the violation. To be effective, the penalty unit costs must be sufficiently high to guide the algorithm toward viable solutions, but not so high that they exclude potentially good alternatives (Abdelsalam & Gabbar, 2021). As with most other hyperparameters in genetic algorithms, the penalty factors are problem specific and need to be calibrated based on the structure of the optimization.

Figure 2-6 illustrates the somewhat abstract concept of a solution space. If infeasible solutions are simply discarded, those near the feasible region offer no value. By instead applying penalty functions, these nearby infeasible solutions can help guide the search toward feasible ones - a key advantage in stochastic optimization methods like Genetic Algorithms.

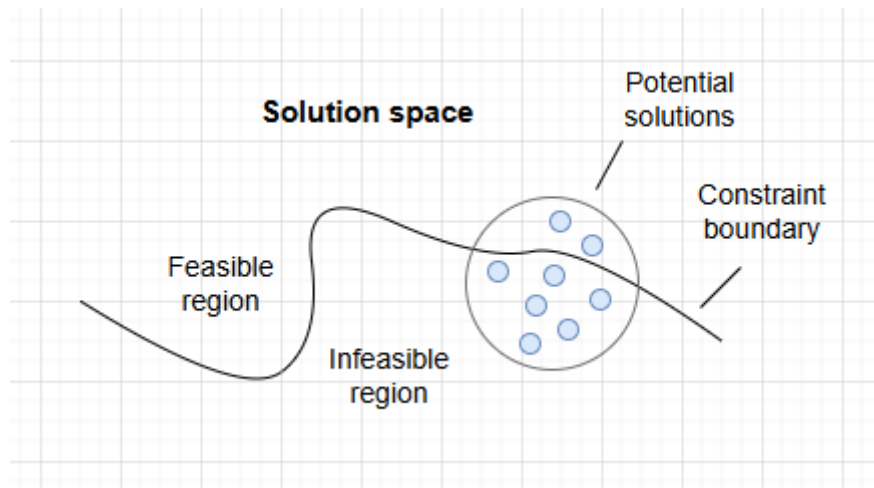


Figure 2-6 Visualization of the solution space in a constrained optimization problem.

2.6 The use of hydraulic models

In the context of pump scheduling optimization, EPANET hydraulic simulations are commonly used, though they can be applied in several different ways. The hydraulic model is used in combination with some type of optimization model. One way to incorporate these is to generate different pump schedules with the optimization algorithm and then using the hydraulic simulation to verify its feasibility in terms of mass balances and pressure requirements (Van Zyl et al., 2004). Some models will also operate iteratively to refine optimization model parameters each time step, which leads to more extensive computational effort.

Some methods will also exclude the use of hydraulic simulation tools such as EPANET, instead opting for integration of the hydraulic equations directly into the optimization framework (Janus et al., 2023; Thomas & Sela, 2024). These methods perform approximations and linearization to enable non-linear hydraulic constraints to be solved by more deterministic optimization solvers. These methods have the potential to provide guaranteed optimal solutions with lower computational processing. However, the drawback being that the simplification of the system naturally introduces inaccuracies of the model.

2.7 Real world implementation

As mentioned, a clear distinction must be made between an optimizing scheme for scheduling and for real time control. Metaheuristics optimizations schemes which utilize hydraulic simulations are not efficient enough to be applied in a real time scenario (Balekelayi & Tesfamariam, 2017).

Another challenge with real world implementation is related to data handling in Supervisory Control and Data Acquisition (SCADA) systems. Calculations done inside the SCADA are generally limited to simpler programming logic and might not be suitable for complex optimization algorithms. Therefore these calculations will have to be done outside the SCADA environment and then be re-entered into the system. This type of operations is not trivial, especially not if they rely on data/software from outside sources such as electricity prices or commercial optimization algorithms. The options are thus to provide a pump schedule or rely on simpler calculation which can be deployed on simpler hardware.

Another challenge with real world applications is that the optimization models sometimes might seem like a “black box”, i.e. limited insight in how the algorithm operates. It is important that the

responsible operator understands the control scheme. Control schemes for WDS are often based on heuristics methods which has been developed over many years, and it is generally hard to implement complete overhauls to automatize and optimize entire operation. A more reasonable approach is to utilize continues improvements, allowing operations and the surrounding infrastructure time to adapt to the new controls.

2.8 Trigger level control

In literature, the most common approach of optimize pump operation is trough explicit pump scheduling (Mala-Jetmarova et al., 2017). For pre-determined demand pattern, this type of scheduling would be able to achieve the absolute optimal solution. For non-determined demand patterns, a perfect prediction model would be required to reach an optimal solution, which would require constant updates and recalibrations based on the current demand (Alvisi & Franchini, 2017). Depending on the inaccuracy of the prediction model, the schedule obtained through explicit pump-scheduling could be suboptimal. Another downside of Pump scheduling as an optimization technique is its vast amounts of decision variables. For each time step, every individual pump on/off status would be a decision variable (Quintiliani & Creaco, 2019). Considering Variable speed pumps (VSP) would further increase the amount of decision variables.

To address these issues, a more robust control scheme is to control pumps on/off statuses based on water levels in the reservoir. Trigger level control (TLC) is an implicit control scheme which refers to setting specific water levels which automatically switch a pump on or off. In its simplest form two constant trigger levels are defined, an upper limit and a lower limit. When the reservoir reaches the upper limit the pump stops, and when it reaches the lower limit, the pumps start. This simple control scheme is referred to as Fixed Trigger Levels and can be seen in figure 2-7.

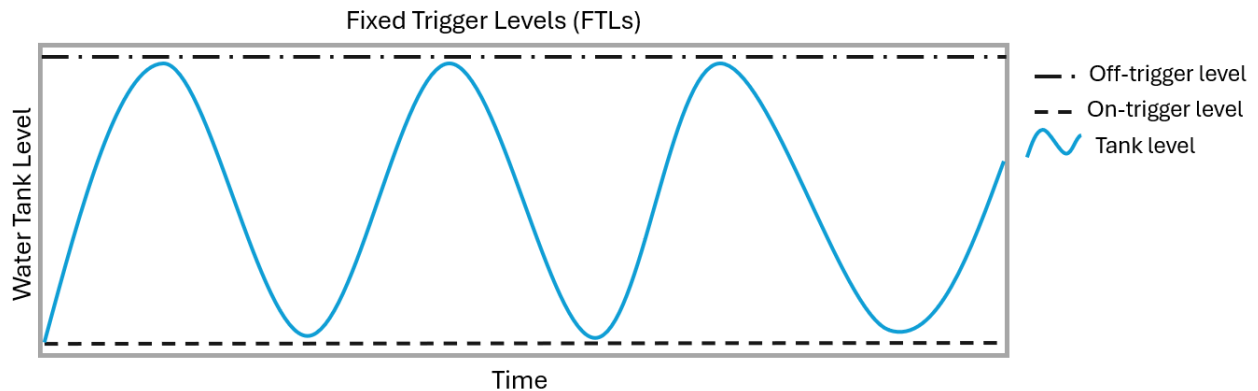


Figure 2-7 Illustration of FTLs scheme, Adapted from (Quintiliani & Creaco, 2019)

In the context of trigger level optimization, more advanced versions of FTL have been developed and evaluated. These models are mostly based on optimizing for electricity cost, based on different “peak” and “off-peak” time periods. One method is Reduced Fixed Trigger levels (RFTLs) (Marchi et al., 2017). In this scheme the off-peak off-trigger level is set to the maximum available level of the WST and the on-trigger level is the decision variable which is optimized. In the peak period the opposite true: on-trigger level is set to the minimum available level, and the off-level is the optimized parameter. The RFTLS scheme can be seen in figure 2-8.

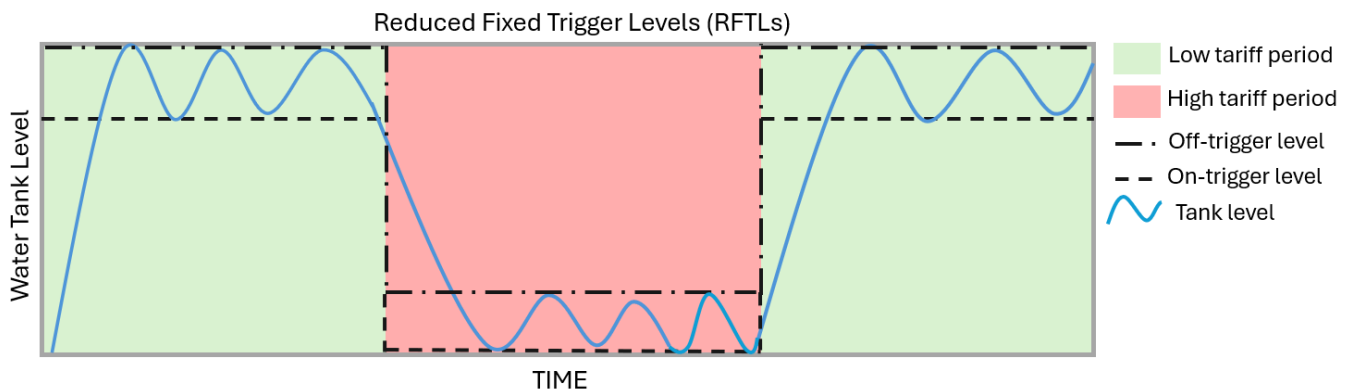


Figure 2-8 Illustration of RFTLs scheme, Adapted from (Quintiliani & Creaco, 2019)

A downside with RFTLs is that it does not guarantee to reach the highest allowable reservoir level at the end of the off-peak period, and the lowest allowable reservoir level at the end of the peak period (Alvisi & Franchini, 2017). It is possible to reduce the allowable tank range during each respective periods, but that would be greatly increase the amount of pump switches, which would be unfavourable from maintenance point of view. To address this issue Variable Trigger levels was introduced. Similarly to RFTLs the off-peak off-trigger value and the peak on-trigger value is set as maximum respectively minimum allowable tank level. The difference is that the other trigger levels are calculated with a power law, with time as the only variable. During the off-peak period the on-trigger level varies from the minimum tank level to some determined level T1 (slightly below maximum tank level). During the on-peak period the off-trigger level varies from the maximum tank level to some determined level T2 (slightly above minimum tank level). A simplification of this method was established in (Housh & Salomons, 2019) where a linear relationship was introduced for the VTLs which eliminates the need for optimizing the power law for each pump. The linearized version of VTLs can be seen in figure 2-9.

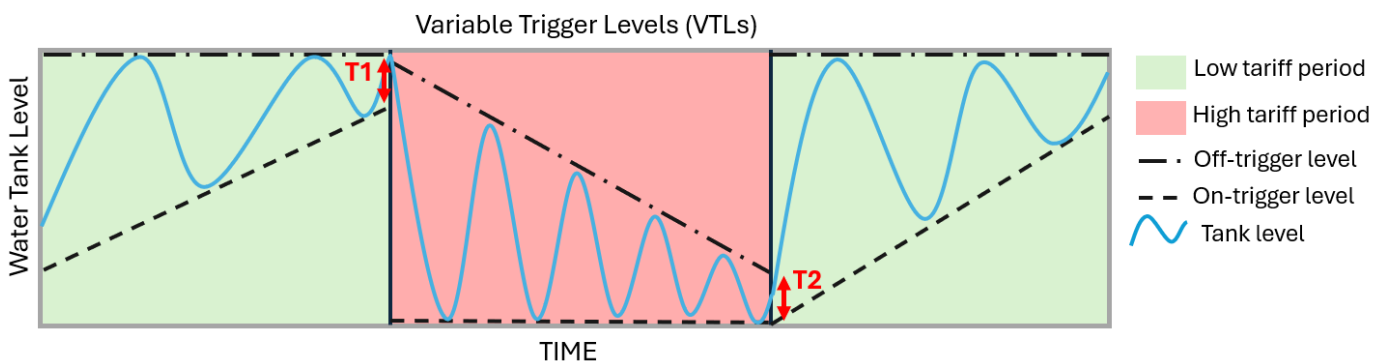


Figure 2-9 Illustration of VTLs scheme, Adapted from (Quintiliani & Creaco, 2019)

In (Quintiliani & Creaco, 2019) a further improved method is introduced which aims to address 1. the pump trigger levels temporal based change, and 2. The additional pump switches caused by tank-filling being affected even at large time periods from the actual tariff change. The main novelty of RFLATS is that trigger levels are optimized only close to the actual tariff change, creating an additional time slot in each respective tariff period. The method thus both achieves more constant

trigger levels and reduces the amount of pump switches. The RFLATs method can be seen in figure 3-10.

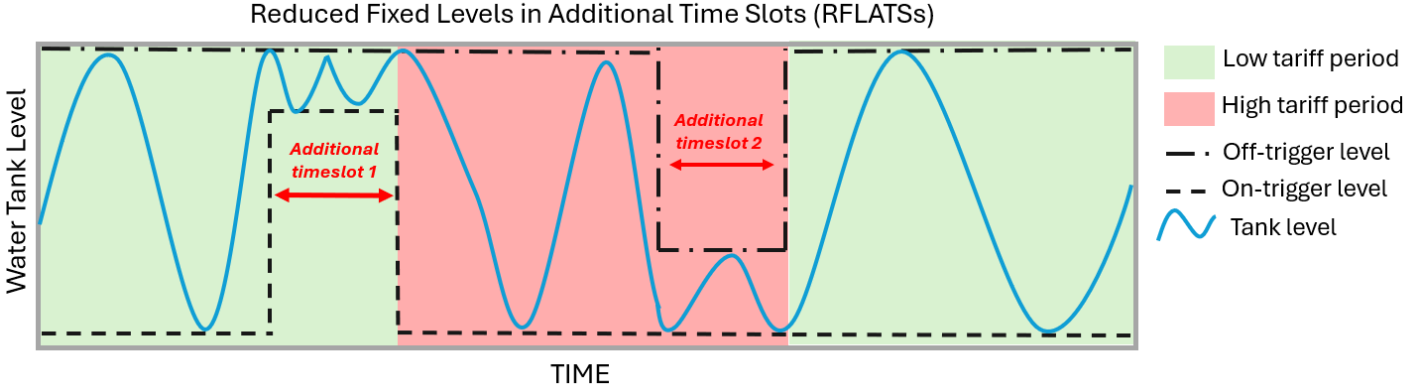


Figure 2-10 Illustration of RFLATs scheme, Adapted from (Quintiliani & Creaco, 2019)

2.9 Setpoint-curve control

In real world scenarios implementation of TLC is often done through the utilization of a set-point curve. In this context, the set-point curve indicates what the reservoir level should be at each time step. The setpoint curve control scheme is also what is used at KoV, all will be the main control scheme covered in the thesis. The values in the setpoint curve are usually defined on an hourly basis, and values in between are then interpolated. In figure 2-11 a simple example of a set-point curve can be seen.

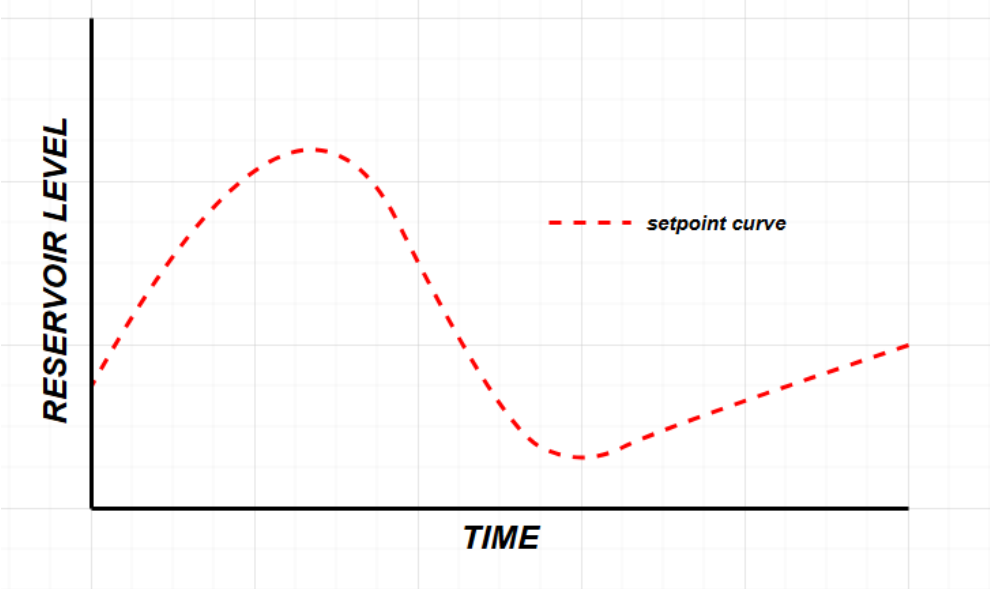


Figure 2-11 Simple setpoint curve illustration.

Most commonly this type of control scheme also requires a second set of parameters which represent the leeway for when pumps should turn on/off. In equation 1.1 the error value (e) is

expressed in terms of setpoint value (SP) and PV process value (PV). Here PV represents the actual reservoir level. The allowable range for the error is then to be expressed according to Upper Deadspann < e(t) < Lower Deadspann.

$$e(t) = SP(t) - PV(t) \tag{2.1}$$

Thus, the operating schedule based on a set-point curve consist of two sets of variables: the actual set-point curve and the allowable variance from said set-point curve. The allowable variance usually differs between pumps. So, a different number of pumps will be running depending on how much the actual value deviates from the set-point curve. The complete control implementation is illustrated by figure 2-12. Here a set-point curve with 2 different pumps can be seen. The specific pump trigger ranges (PTR) can be seen in table 2.1. Essentially can this type of scheme be seen as pump specific TLC which are constantly updating.

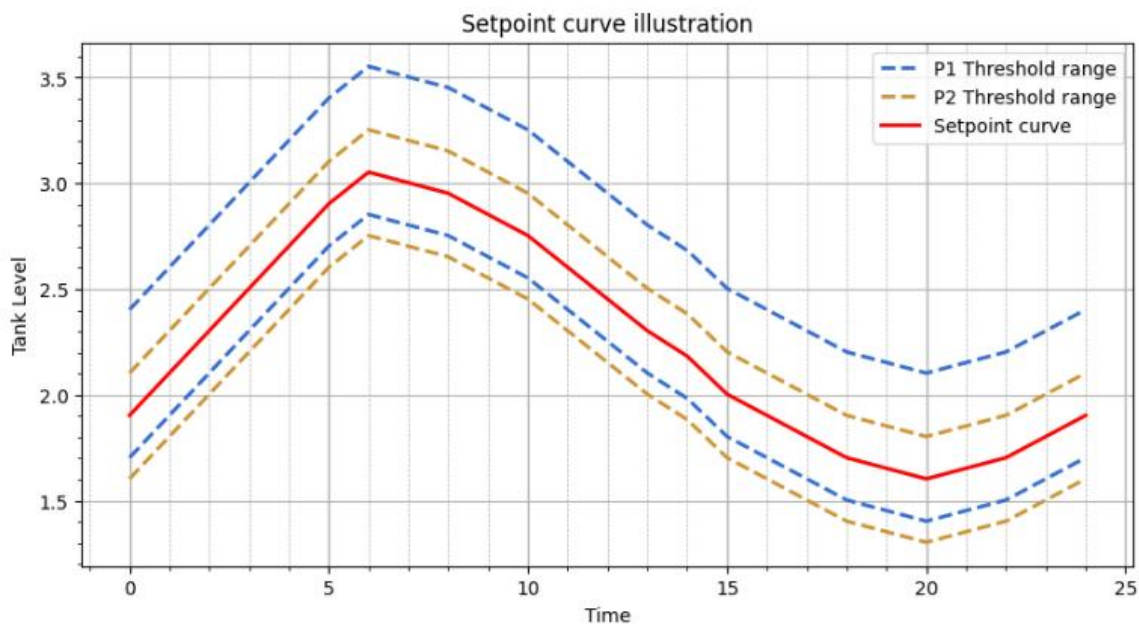


Figure 2-12 Illustration of a set-point curve with individual pump-specific threshold ranges.

Table 2-1 Start and stop thresholds for pumps relative to the set-point curve (Pump trigger ranges).

	Start	Stop
P1	0,2	-0,5
P2	0,3	-0,2

For WDS which contain one or multiple reservoirs, different variations of TLC are widely used in practice (Salomons & Housh, 2020). It is generally preferred to explicit pump scheduling as it offers a simpler control scheme with fewer decision variables. It is also more suitable for a decentralized

system where the control logic situated in the PLC of the WST or the connected pumping station. The general methodology is to generate and attempt to optimize these set-point curves offline, and then enter it into the SCADA system (Housh & Salomons, 2019).

It is important to note that there obviously are downsides to using a trigger level as opposed explicit pump scheduling. A trigger level approach essentially trades optimality for more reliable solution. The trigger-based approach lacks adaptability, and often a temporal offset occurs between the filling/emptying of the tank and the off-peak and peak tariff periods (Alvisi & Franchini, 2017). Which in the end results in smaller economic gain. Ultimately, the optimal control scheme for a given system will depend on the feasibility of dynamically adjusting the model, the frequency of system adjustments, the availability of relevant data, the system's sensitivity, and the extent of its fluctuations.

3. Method

Based on the literature review, no existing method was deemed a good fit based on the defined problem definition. The goal of this work is essentially to develop a model capable of generating setpoint curves from available data that can be directly implemented in the SCADA system. Existing methods fail to achieve the necessary balance between rule-based control and optimization-based approaches. Although certain trigger level-based strategies show promise, they lack sufficient temporal resolution to capture short-term dynamics effectively. Which is an essential requirement when optimizing setpoint curves. As previously discussed, a setpoint curve can essentially be viewed as a trigger-level scheme with shorter and more continuously varying trigger intervals.

Other optimization-based methods, while mathematically advanced, come with important limitations in this context. Many of these rely on detailed system models and require the problem to be reformulated for each specific network setup or tariff structure, making them difficult to reuse or adapt. They also often depend on precise predictions of future conditions and need a lot of adjustments to perform well. As the system becomes more complex, these methods can become too slow or difficult to manage. Moreover, they are not always able to produce the desired output of setpoint curves, which are needed by the existing control system.

To address these problems, a key-parameter encoded genetic algorithm was developed for integration with the current set-point curve-based control scheme. Each setpoint curve is represented by a concise set of parameters baseline level, peak/dip timing and amplitude, and three curvature descriptors that directly generate the smooth, continuous curves expected by the SCADA system. An iterative GA optimization, utilizing EPANET hydraulic simulations for evaluations, evolves these parameters to minimize energy costs, reduce pump cycling, and enforce water tank constraints. By combining the ease of rule-based curve deployment with the global-search capabilities of metaheuristic optimization, this method provides a flexible and practical solution to the defined problem. In figure 3-1 a complete schematic of the optimization scheme can be seen.

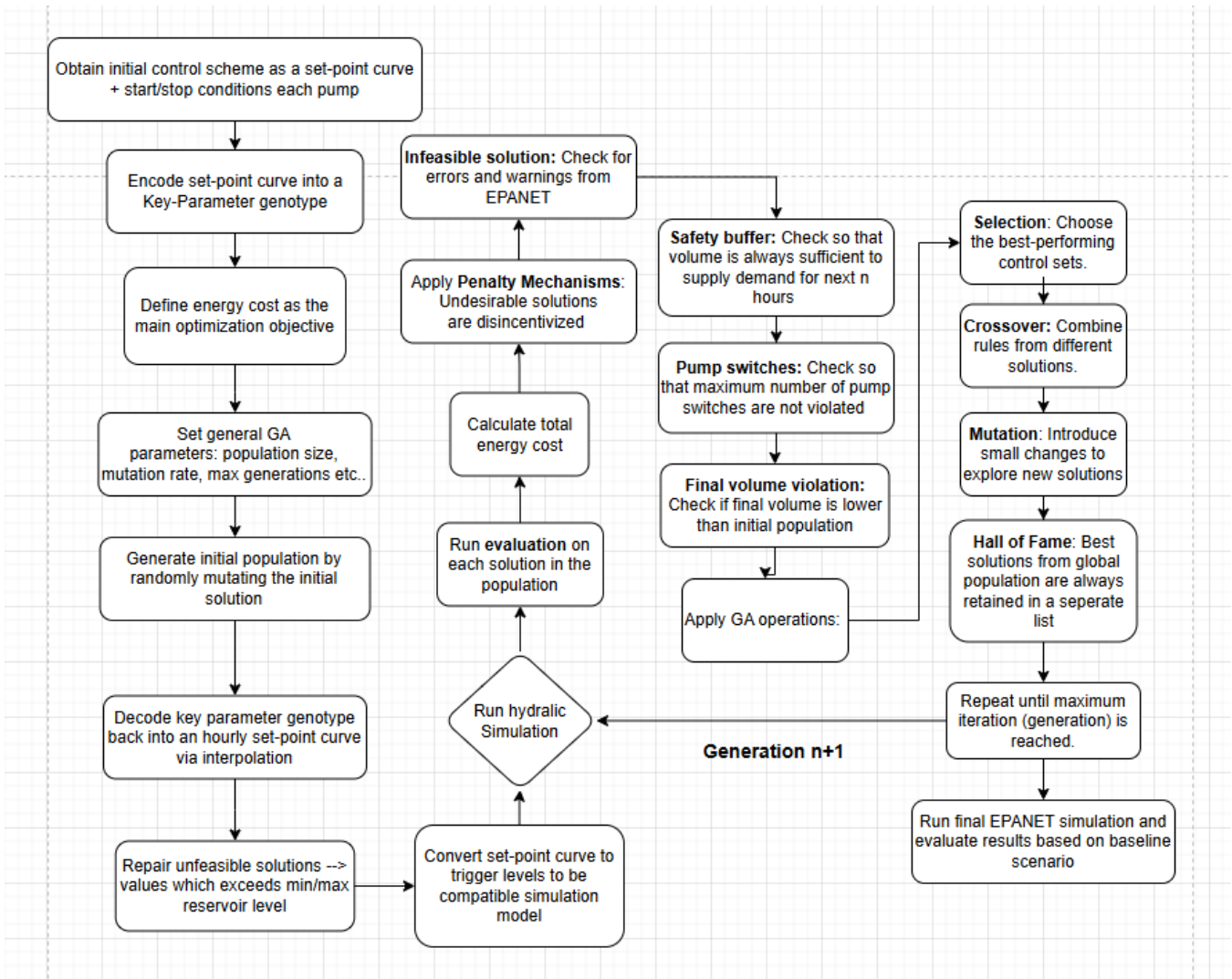


Figure 3-1 Flow diagram of complete optimization scheme.

3.1 Software

3.1.1 EPANET

EPANET is a commonly used hydraulic simulation software originally developed by the US Environment Protection Agency (EPA) in 1994 (Rossman et al., 2002). EPANET 2 is the latest major release and is available both as a free stand-alone software, and as an open-source toolkit. Some software (such as Mike+) utilizes EPANET as a basis for hydraulic calculations.

3.1.2 EPyT

EPyT is an open-source python package which enables a high-level python programming interface for EPANET (Kyriakou et al., 2023). It is essentially a wrapper for EPANET'S C based API. EPyT is designed to support automated modeling, simulation, and control of water distribution networks. Many previous articles covering similar in the field of WDS optimization utilizes the EPANET-MATLAB Toolkit, which has previously served as the main toolkit in linking the EPANET engine to code and computing environments.

3.1.3 DEAP

DEAP (Distributed Evolutionary Algorithms in Python) is a powerful, flexible, and open-source Python framework specifically designed for creating and experimenting with evolutionary algorithms (Fortin et al., 2012). It supports implementation of evolutionary computation techniques such as genetic algorithms, genetic programming, evolution strategies, particle swarm optimization and differential evolution.

3.1.4 MIKE+

MIKE+ is an integrated modeling platform developed by DHI for the simulation and management of water systems (DHI, 2024). It builds upon the EPANET engine, enhancing it with features such as GIS-based visualization, support for additional boundary conditions, and advanced control options. In this study, most modifications and calibrations to the HPZ-G model were performed within MIKE+, including adjustments to pump curves and pipe configurations to better align with observed SCADA data. MIKE+ was also used as a visualization tool, enabling detailed inspection of model behavior, such as flow patterns and pressure distribution across different network regions.

3.2 Hydraulic models

Two main hydraulic models were used in the process of developing the optimization model. The HPZ-G model was obviously the main focus of this thesis, and naturally the results will mostly relate to this hydraulic model. However, a much simpler network model, NET-1 was also used quite heavily during the development of the optimization model.

3.2.1 NET-1

The initial versions of the model were based on a simple small WDS, known as “NET-1”. This is a theoretical network used for testing and is one of the standard networks available in EPANET (Rossman et al., 2002). It includes a simple layout consisting of 10 interconnected nodes, 12 pipes, 1 reservoir, 1 pump and 1 WST. It also contains a simple predefined demand pattern. A visual representation of the Net1 model can be seen in figure 3-2.

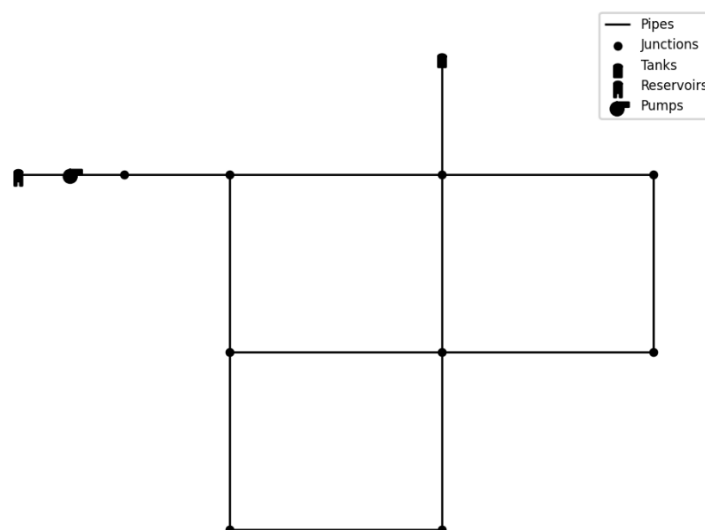


Figure 3-2 Hydraulic representation of the NET-1 network. Generated using EPyT (Kyriakou et al., 2023)

Even though the real-world implication based on result from the NET-1 model are limited. These results still provide meaningful insight of the optimization algorithm. For the more complex hydraulic model the optimization process can be harder to decipher given the complex hydraulic relationship, and other data inputs. The NET-1 result essentially provide simple demonstration of how optimization algorithm navigates in the solution space.

3.2.2 HPZ-G

The hydraulic model which represents the actual HPZ which exist inside the greater Gothenburg water distribution network. It contains 2 pumping stations with 3 pumps each and a single WST. The Actual real network is obviously more complex and containing around 5500 nodes and links each. The demand patterns also contain more complexity as it is based on actual measured consumption in the zone. The measured consumption from different consumer is aggregated in different demand nodes. The demand pattern for these nodes is then representable for all consumers in these aggregated points. Due to security reason the detailed layout of HPZ-G network cannot be shared. However, a simple illustration of the network structure can be seen in figure 3-3.

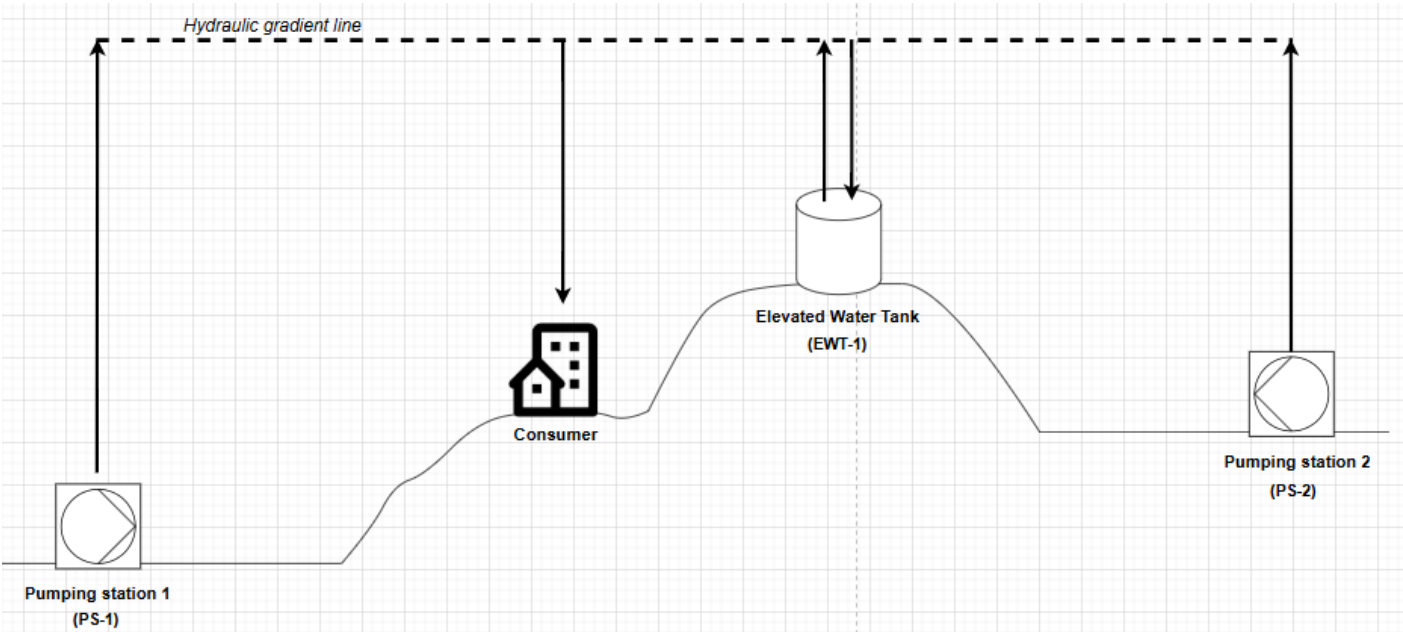


Figure 3-3 Simplified schematic overview of the studied HPZ, referred to a "HPZ-G"

3.3 Validation of the HPZ-G hydraulic model

To ensure the simulations accurately reflected the real network behavior, the model was validated using SCADA data from actual operations. The same control rules implemented in the SCADA system were applied to the model to replicate comparable operating conditions.

3.3.1 Pump curve calibrations

The pumps operating in HPZ are relatively old, and their efficiency has decreased over years of use. As a result, relying on manufacturer-provided efficiency data is not reliable (Bunn & Reynolds, 2009). Instead, SCADA data was used to develop updated pump curves for use in the simulation model. The data used for generating the new pump curves are based on the targeted simulation period. The relevant data which was used was P_{IU} (Pumps in use), Q (flow), P_{OUT} and P_{IN} (outgoing/ingoing pressure).

3.3.2 Hydraulic simulation validation

In this step, the simulation results were compared to measured data from the real network over the same time-period, utilizing the same control scheme setup. The complete seven-day period was simulated in consecutive 24-hour intervals.

3.4 Timesteps

Since the hydraulic model is the base of the optimization process, naturally the parameters will influence the results. In this section a brief explanation of some relevant timestep parameters will be presented, and their effect. In the result section a sensitivity analysis on how the different timesteps effects runtime is presented.

3.4.1 Hydraulic timestep

The hydraulic timestep determines the time interval between calculations of the hydraulic state of the network (Rossman et al., 2002). So essentially how often flows, pressures, tank levels etc. are updated. Generally speaking: a smaller hydraulic timestep provides a more realistic and accurate model, at the expense of longer computation time. Since the optimization scheme might require a lot of simulations for each iteration, this can have a very large effect on computational time. Therefore, for this project the hydraulic timestep variable will have to balance an accurate hydraulic simulation, and a sufficiently short runtime for the optimization loop.

3.4.2 Reporting timestep

Reporting timestep is the interval between which output results are generated, it does not have any effect on the hydraulic simulation (Rossman et al., 2002). The reporting timestep does however determine the temporal resolution of the optimization model. For example: if the interval is one hour, only 1 pumps switch could be registered during that period. The reporting timestep must be greater than or equal to the hydraulic timestep and should also be an integer multiple of it to prevent the introduction of unintended intermediate timesteps. This is due to a every output reporting period will always require a new hydraulic timestep (Marini et al., 2023).

3.4.3 Rule timestep

Is the time step used to check for changes in system status caused by the activation of rule-based controls between hydraulic time steps (Rossman et al., 2002). By default, it is set to one-tenth of the hydraulic time step. If a hydraulic event is triggered (such as a maximum reservoir level), EPANET will insert an additional hydraulic solution step to recalculate the new status of system. The rule timestep can enable the use of a longer hydraulic timestep while still maintaining a sufficiently accurate hydraulic system.

3.4.4 Interpolation interval timestep

The trigger interval timestep is a “artificial” timestep introduced to lessen computation strain on the model. When constructing the trigger levels in the SCADA-system every value in between is established through linear interpolation. These interpolations are updated every minute. When recreating this in the simulation every interpolation needs to be defined as a new control rule. Updating the interpolation every minute is not feasible for the simulation since it would massively increase the amount of control rule which had to be added, and thus significantly increase computational times. Instead, the Interpolation interval timestep decides how often the interpolation should be performed. In the model the timestep was chosen to be 15 minutes, as a good balance between runtime and accuracy. Validation of this parameter choice is provided in the result section. Figure 3-4 illustrates the variables effect on the setpoint-curve.

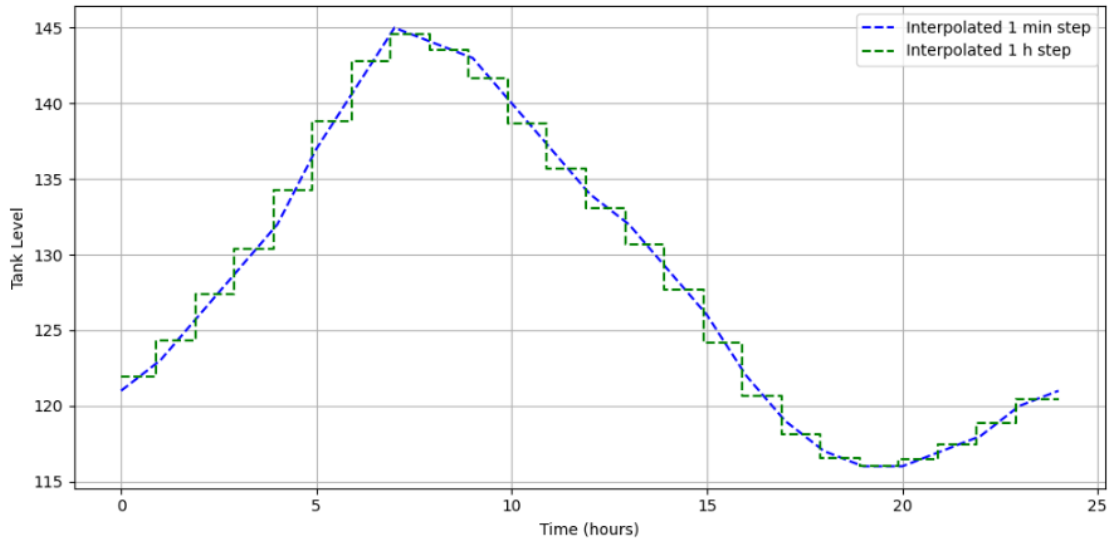


Figure 3-4 Illustration of the effect Interpolation interval timestep has on the shape of the setpoint curve.

3.5 Objective function

3.5.1 Energy cost

The driving force of this model will be to optimize the use of energy in a way which reduces the total cost. Minimizing energy cost can be divided into two parts: Reducing overall energy consumption and shifting consumption from high demand periods. It is important to note that these objectives can be conflicting, meaning that an optimal solution could increase the power consumption, if it is shifted to a lower demand period. The energy cost is calculated with by equation 3.1.

$$E = \sum_{i=1}^T \sum_{k=1}^{n_p} P_{i,k} \gamma_k \quad (3.1)$$

Where:

E = Total energy cost over the time horizon

n_p = Total number of pumps

T = Total number of time steps.

$P_{i,k}$ = Power consumption of pump k at time i

γ_k = Electricity tariff at time k

3.5.2 Constraints

Security of water supply

As mentioned, one of the main aims of a reservoir is to increase reliability. Therefore it is important to always have a certain volume in case of an emergency such as pump failure, power outage, major leakage or any other major failure. Equation 3.2 defines the constraint which is based on securing a volume which can supply the High-pressure Zone for n hours, based on expected demand. The

dynamic nature of the constraint means that a greater volume will have to be kept before high consumption periods, e.g. Morning and evenings, in comparison to low consumption periods, e.g. during night.

$$\text{pen}_{SSV} = \sum_{i=1}^T \delta_i \cdot \lambda_{SSV} \cdot \left(\sum_{j=0}^{H-1} \text{demand}_{i,j} - \text{volume}_i \right) \quad (3.2)$$

Where:

pen_{SSV} = Weighted penalty for security of supply violation.

H = “Emergency” duration in hours.

δ_i = Binary value ensuring penalty only activates when the rolling demand exceeds the stored volume.

λ_{SSV} = Penalty factor for security of water supply violation.

$\text{demand}_{i,j}$ = Current demand in timestep i in “lookahead horizon” j .

Final reservoir level

To retain a stable tank level between simulation horizons, the initial and final tank volume should ideally be minimized. Essentially, the stored volume inside the tank could be seen as potential energy. As such, unless the final reservoir level constraint is set, the optimal solution would converge towards solutions with lower final volumes. A penalty function is therefore needed to prevent this type of undesirable solution. Adding a hard constraint is undesirable since all solutions which obtain a final volume in the span between maximum and minimum, will at least be close to the feasible region. Instead, a penalty function based on the deviation from the final solution would be more suitable, as defined in Equation (3.3). This penalty is only applied when the final volume is lower than the initial volume. When the final volume is higher, no penalty is applied.

$$\text{pen}_{FFL} = \delta_{FFL} \cdot \lambda_{FFL} \cdot (V_{\text{init}} - V_{\text{final}}) \quad (3.3)$$

Where:

pen_{FFL} = Weighted penalty for final reservoir level violation.

V_{init} = Volume at initial timestep

V_{final} = Volume at final timestep

δ_{FFL} = Binary value ensuring penalty only activates when final volume exceeds initial volume.

λ_{FFL} = Penalty factor for final reservoir level violation

Maintenance cost

It can be important to consider how optimization models affect the cost of maintenance. This refers to the additional wear caused by pumps operating in an inefficient manner. To simplify and concretize this, some type of surrogate measure is often used to account for maintenance (Mala-

Jetmarova et al., 2017). In this thesis this surrogate measure will be the number of times pumps turn on/off, referred to as pump switches. An allowable amount of pump switches will be set, then any more pump switch will result in a penalty factor, as can be seen in equation 3.4.

$$\text{if } S_A > S_{th} \rightarrow \text{pen}_{PS} = (S_A - S_{th}) \cdot \lambda_{ps} \quad (3.4)$$

Where:

pen_{PS} = Weighted penalty for pump switch violations.

S_A = Actual amount of pump switches

S_{th} = Threshold for acceptable pump switches.

λ_{PS} = Penalty factor for exceeding the pump switch threshold

Hydraulic simulation constraints

The hydraulic feasibility of each solution is handled implicitly by hydraulic simulation (Rossman et al., 2002). These constraints include acceptable ranges for: pressures in nodes, water levels in reservoirs, etc. These constraints are not explicitly expressed in the optimization scheme and are instead considered by registering warnings and crashes from each simulation run. If a solution contains a critical warning, the fitness of the solution will not be evaluated, and the solution will return a poor fitness value. If a non-critical warning is obtained, a penalty is added to the evaluated fitness function. The Constraint/penalty process can be seen in figure 3-5.

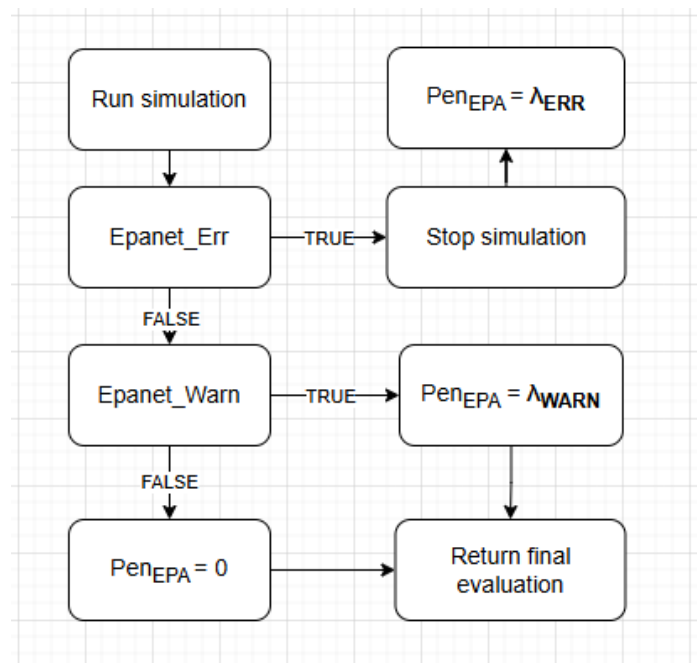


Figure 3-5 Schematic overview of EPANET error/warning handling.

3.6 The Optimization algorithm

3.6.1 Key-parameter encoding

As an alternative using the 24-value setpoint curve string directly in the optimization algorithm, key parameter encoding is used. In the genotype each gene represents a meaningful coefficient which is not dependent on the rest of the genes (as was the case with the 24 values string). The genotype consists of 10 parameters which defines some key characteristic about the set-point curve, the rest of the values are interpolated to recreate a new string of 24 values (set-point curve). The encoding/decoding process can be summarized in figure 3-6.

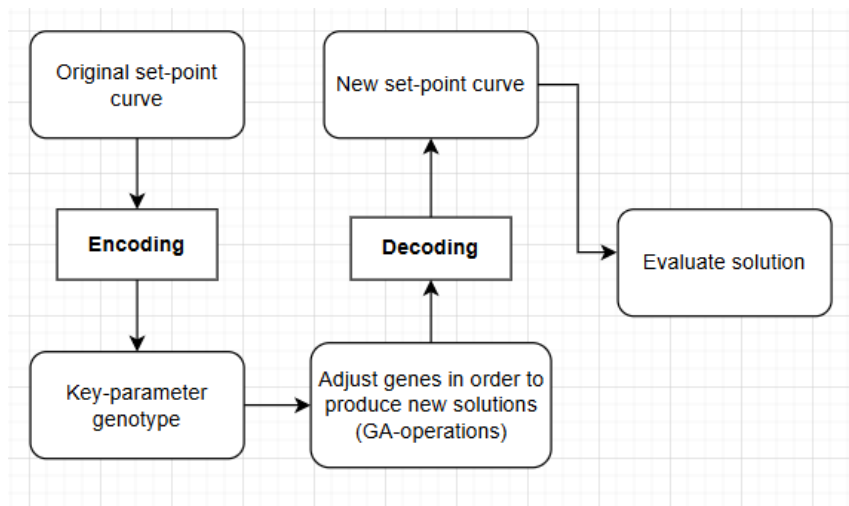


Figure 3-6 Schematic overview of the encoding–decoding process used in the genetic algorithm.

The key-parameters contain information about how long each peak/dip are, and when in time each of these apexes occur. The genotype also contains information regarding the concave/convex nature of the curve in between these extreme points and the initial/final level. For simplicity and continuity, the endpoints always balance out, i.e. initial level = final level. The endpoints are thus defined by a single variable. A complete list of all genes in the genotype can be seen in table 3-1.

Table 3-1 Description of variables in the key-parameter encoding scheme.

Parameter	Variable name	Description
Baseline Level	L_0	Defines the tank level at the start ($t = 0$) and end ($t = 24$) of the 24-hour cycle.
Peak Height	H_P	Maximum level reached during the daily cycle. Represents the apex of the curve.
Peak Timing	T_P	Time of day when the peak level occurs. Controls the horizontal position of the peak.
Peak Width	W_P	Duration of the peak, how long the curve stays elevated. Influences steepness and curvature.
Dip Timing	T_D	Time of day when the lowest reservoir level (trough) occurs.

Dip Height	H_D	Minimum level reached during the day. Typically, in the evening or night period.
Dip Width	W_D	Duration of the dip. Controls how sharp or extended the low-level period is.
Rise Curvature	CRV_1	Controls the shape of the rise from L_0 to H_P . Affects steepness and concavity.
Descent Curvature	CRV_2	Controls the shape of the fall from H_P to H_D . Refines the rounding of the descent.
Recovery Curvature	CRV_3	Controls the shape of the recovery from H_D to back to L_0 . Influences the tail of the curve.

The key-parameter encoding offers a lightweight, yet powerful method to generate new solutions. It focuses on the key shapes of the set-point curve to work well in combination with the genetic algorithm. It uses a conscious simplification of the solution space to focus on feasible solutions. The main issue is that the encoding can be too simplified and thus unable to find the true optimal curve. For example, some demand pattern contains multiple peaks which a curve with a single peak/dip might be unable to handle optimally. A possible solution could be to introduce more parameters, but at the cost of a more complex solution space. In the end it comes down to a compromise between optimization potential and complexity. Figure 3-7 each variable can be seen in an actual curve.

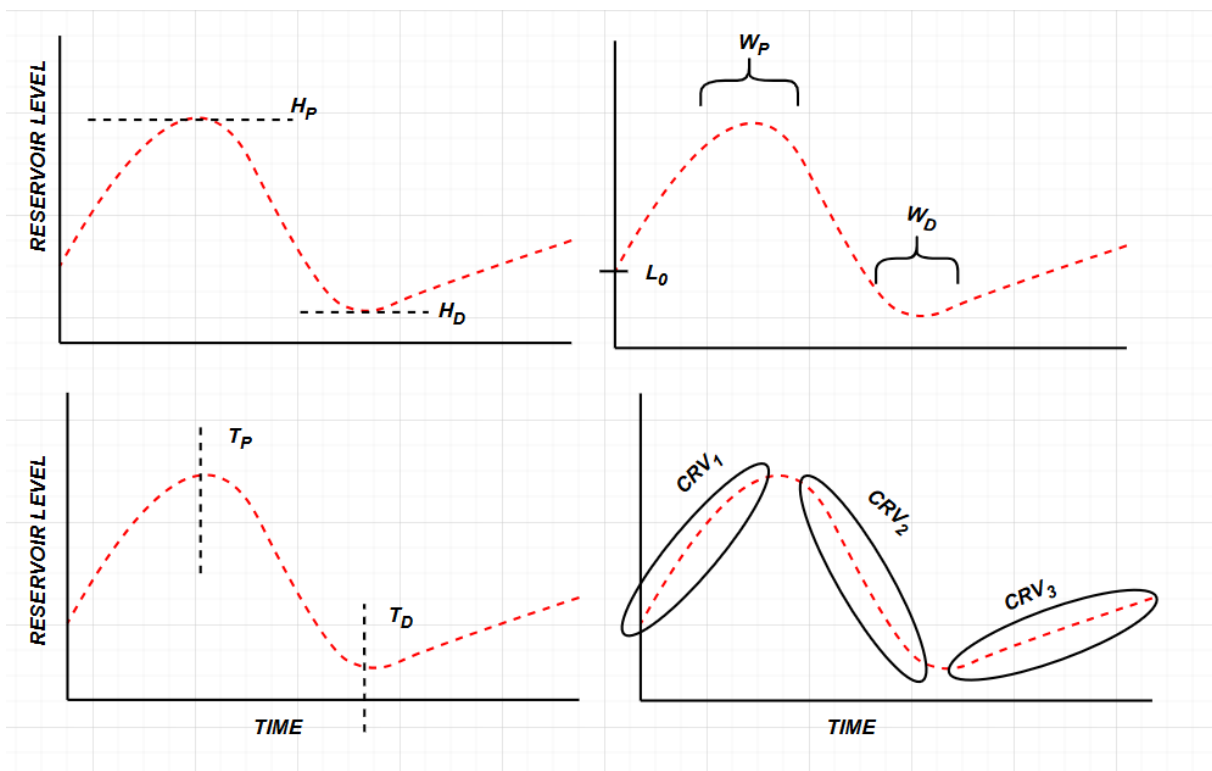


Figure 3-7 Visual representation of the key-parameters.

3.6.2 Pump trigger ranges variable

To capture variations in pump trigger ranges an additional variable was added to the genotype to represent the overall “strictness” of the trigger thresholds. For simplicity, this was implemented as a single scaling variable, referred to as “*pt_var*”. The pump trigger ranges are initially defined based on the baseline control scheme and then uniformly scaled by this factor. Figure 3-8 illustrates the original trigger ranges (*pt_var* = 1), while Figure 3-9 shows how the ranges are reduced when *pt_var* is set to 0.5. The differences between the figures illustrating how the pump trigger ranges become narrower around the set-point curve as the strictness parameter is reduced.

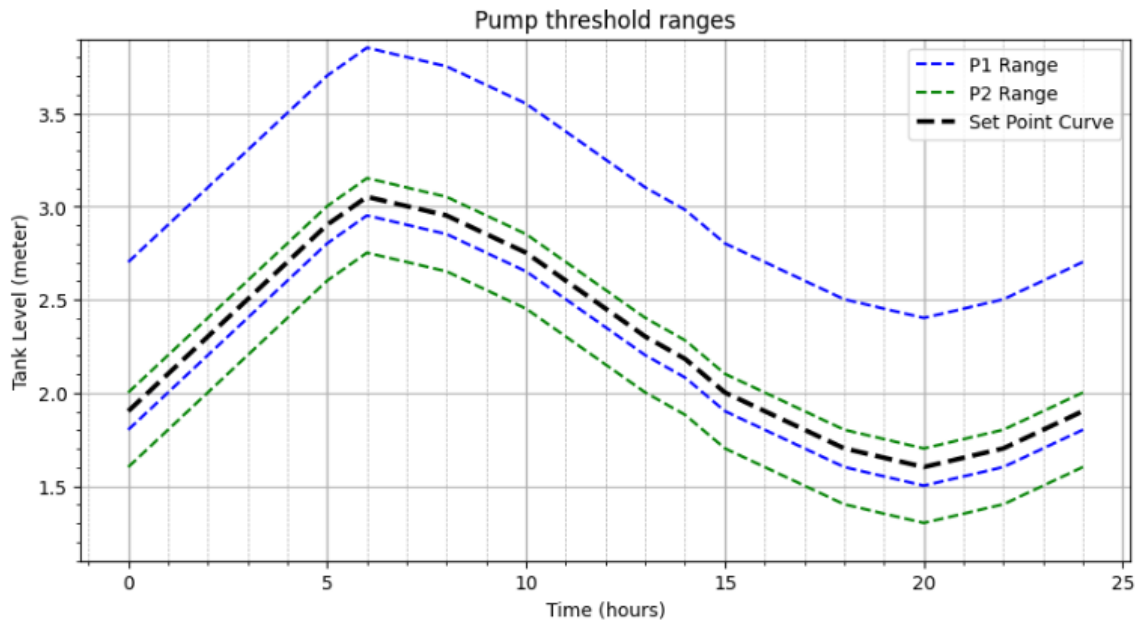


Figure 3-8 Example of pump threshold ranges (*pt_var* = 1)

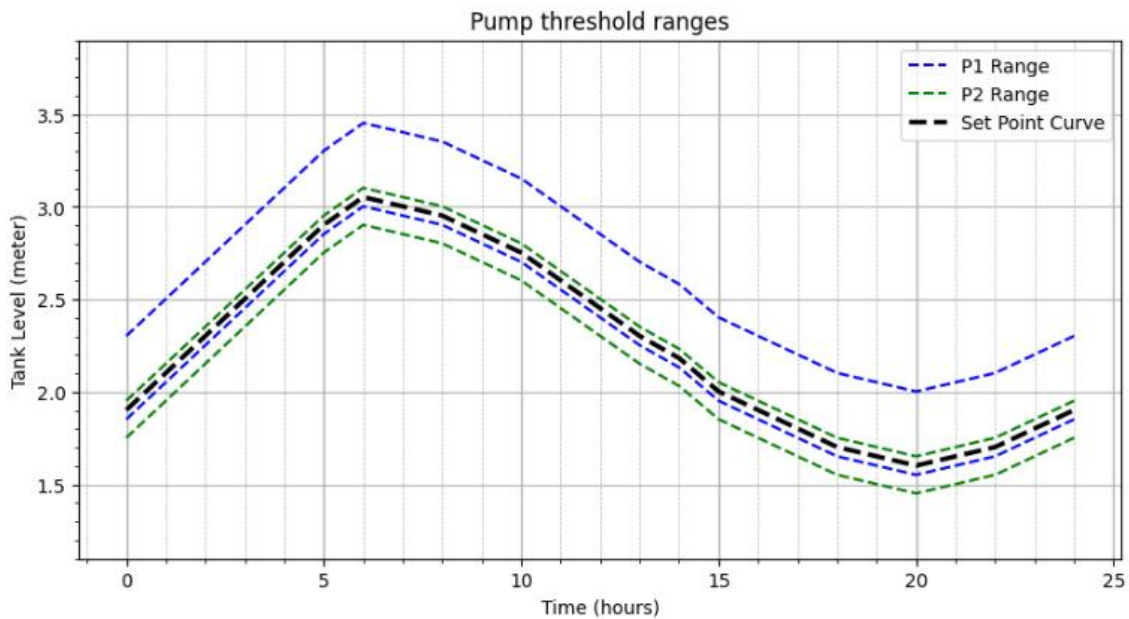


Figure 3-9 Pump threshold ranges after scaling by *pt_var* = 0.5

This variable was primarily introduced to address the systematic deviation often observed between the setpoint curve and the actual reservoir level in the existing control system. In practice, the reservoir level does not perfectly follow the defined setpoint but instead oscillates around it. This discrepancy arises mainly from the relatively wide pump trigger ranges that define the allowable deviation from the setpoint before a pump is switched on or off. In figure 3-10 this deviations is shown by plotting the actual measured reservoir level and the corresponding setpoint curve during the day 2022-11-07.

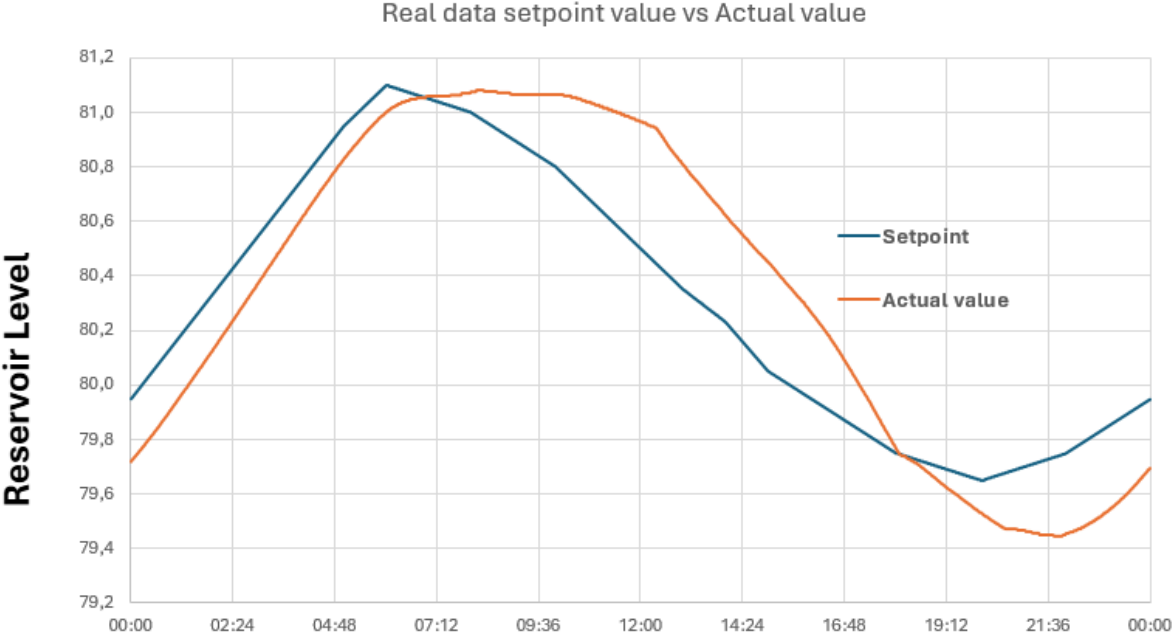


Figure 3-10 setpoint curve vs actual reservoir level, real-world data from 2022-11-07

3.6.3 Mutation scheme

The mutation operator utilized is a gene-wise probabilistic scheme. It includes a repair mechanism to ensure that all mutated individuals remain within feasible bounds (clipping). Since the initial population is generated from a near-optimal reference solution, the mutation scheme is intentionally biased toward local exploration. By employing parameter-specific mutation probabilities and step sizes, the approach offers fine-tuned control over search dynamics and enables specific adjustments based on insights from prior optimization runs. The mutation process is structured in three principal stages. Gene-Wise Probabilistic mutation, feasibility Clipping and peak-Dip Separation.

Gene-Wise Probabilistic mutation

First, each gene is assigned a mutation probability p_i . This probability is different from the global mutation parameter which determines if the mutation scheme should be triggered. If a given gene is selected for mutation, a uniformly distributed mutation-step $\delta_i \in [\Delta_i^{\min}, \Delta_i^{\max}]$ is sampled and added to the current gene value. The mutation can be seen in equation 3.5.

$$if r < p_i \rightarrow x_n = x_o + \delta_i \tag{3.5}$$

Where:

r = Random value between 1 and 0.

p_i = Mutation probability.

x_n = New gene value.

x_o = Old gene value.

δ_i = Mutation-step.

Δ_i^{\min} = Lower limit of mutation range.

Δ_i^{\max} = Upper limit of mutation range.

The mutation probabilities and step ranges are gene-specific and chosen based on the impact of each parameter. For instance, peak and dip timings may allow wider mutation ranges, while curvature parameters are adjusted more conservatively.

Feasibility Clipping

After mutation, each gene is clipped to its allowable domain to ensure physical feasibility. The bounds are defined based on operational limits of the reservoir (e.g., minimum and maximum water levels, valid time intervals, etc.). Clipping bounds can be found in table 3-2.

Table 3-2 Clipping bound for key parameters

Gene	Min	Max
L_0	79.65	81.1
H_P	79.65	81.1
T_P	0.0	24.0
W_P	0.0	24.0
T_D	0.0	24.0
H_D	79.65	81.1
W_D	0.0	24.0
CRV_1	-0.5	2.0
CRV_2	-0.5	0.5
CRV_3	-0.5	2.0

Peak-Dip Separation

A custom repair mechanism which enforces a minimum temporal separation between the peak and dip times. If the difference $T_D - T_P$ violates the specified threshold (e.g. 1 hour), the dip time is adjusted forward or the peak time backward, depending on their location within the simulation horizon. This correction prevents the overlapping of peak and dip features, which could otherwise lead to infeasible curve structures. Figure 3-11 illustrates the peak-dip separation.

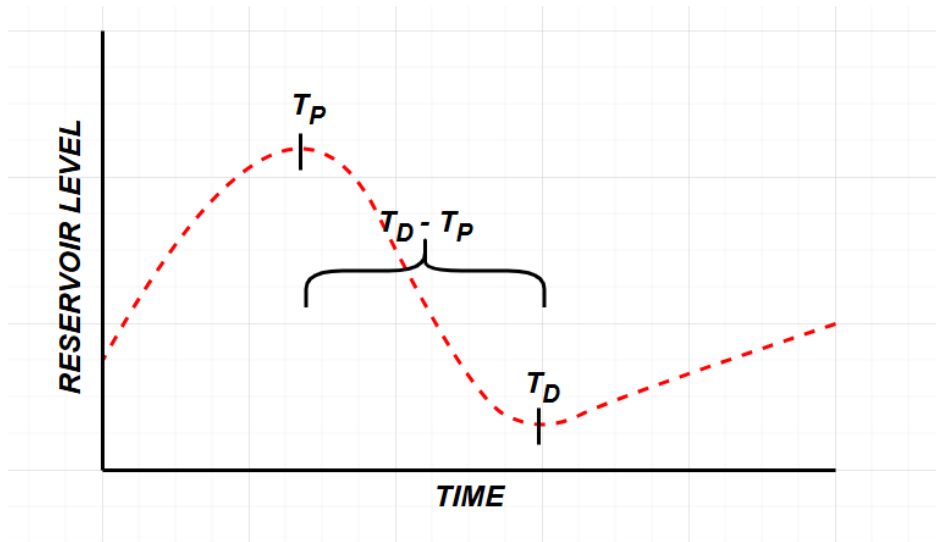


Figure 3-11 Visual representation of the Peak-Dip separation constraint

3.6.4 Initialization Scheme

The initialization approach employs a warm-start strategy in which all individuals are derived from a predefined base genotype, representing the current set-point curve used in the current control scheme. Injecting good solutions in the initial population has proven to be favorable in terms of search efficiency and reliability (Nicklow et al., 2010). Each individual in the initial population is generated by cloning this base solution and applying the existing mutation scheme to introduce localized variation. This ensures all initial candidates remain in the vicinity of a known feasible and operationally effective solution.

3.6.5 Crossover Scheme

The crossover strategy employs a block-wise recombination operator which is designed to preserve functional grouping within the genotype. Traditional crossover operators in genetic algorithms do not always retain the beneficial traits of the parent individuals, which can lead to lower-quality offspring and reduce the overall performance of the algorithm (Da Silva Brito et al., 2023). When crossover is applied, one of several predefined blocks, covering time-location parameters, height values, or curvature descriptors is selected and exchanged between parents. The likelihood of each block being chosen is governed by a weighting variable. Occasionally, a full two-point crossover is used to introduce greater diversity. The predefined block of the crossover scheme can be seen in figure 3-12.

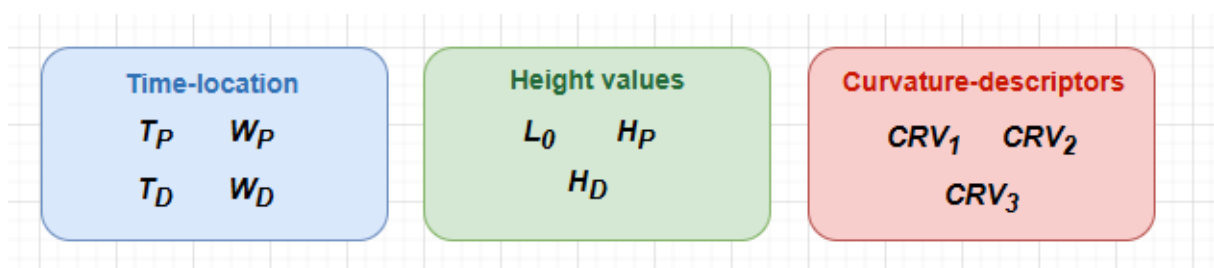


Figure 3-12 Visual representation of the block-based crossover scheme.

This approach helps keep related features, like the shape and timing of peaks or dips intact during crossover, while still allowing new variation to be introduced when needed. It works well for genotypes where some parameters depend on each other, since randomly mixing genes could break important combinations and lead to worse solutions.

3.6.6 Elitism Strategy

To ensure that the best solution found so far is not lost between generations, the algorithm incorporates an elitism mechanism. At the end of each generation, the current best individual is explicitly preserved and reintroduced into the next population without modification. This guarantees that the most fit solution is always retained, providing a safeguard against regression in solution quality.

3.6.7 Selection strategy

The genetic algorithm employs tournament selection, which was explained in the literature section. The tournament selection method is particularly effective for problems where the objective function includes multiple penalty/constraint terms since it relies on relative comparisons rather than global fitness scaling.

3.6.8 Parallelization

To significantly speed up the optimization loop, parallelization was implemented. The implementation is based on “distributed fitness evaluation” which is generally considered the most standard approach for parallelization in the context of GAs (Sivanandam & Deepa, 2008). More specifically a “Master-Slave” configuration is used, which uses a single population where the evaluation of solutions is performed in parallel by the “slave” processes. The “master” process is essentially responsible for the main GA loop, sending out each individual to available “slaves” and collecting the results and at the end of each iteration. The “master” process is also responsible for updating the population for each new generation, performing the mutations and crossover operation. In figure 3-13 the general structure of the “Master-Slave” configuration can be seen.

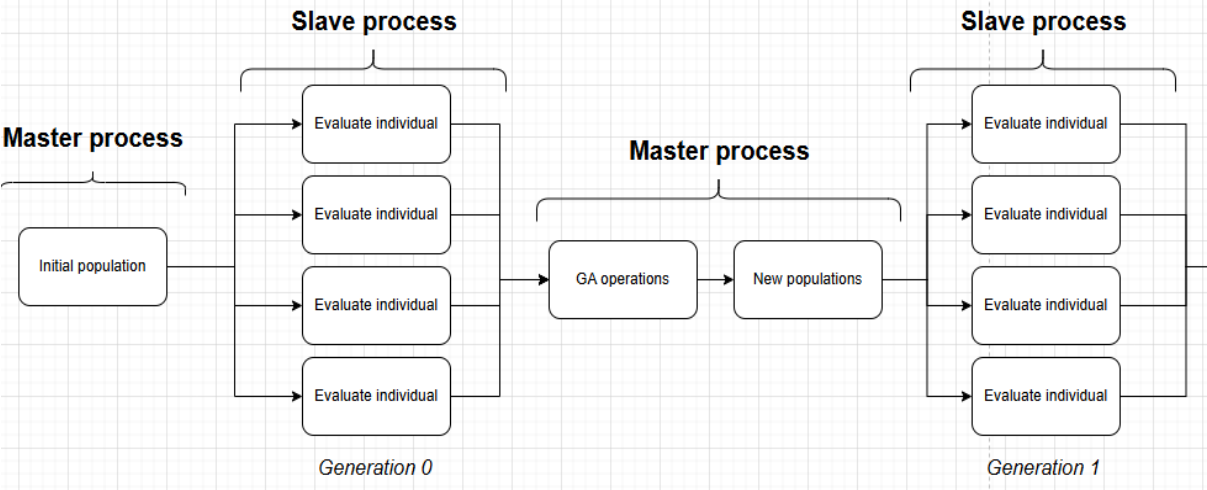


Figure 3-13 Schematic overview of the parallel evaluation process using a “master-slave” method.

3.6.9 Hyper-parameter strategy

A heuristic/"trial and error" hyperparameter tuning process was used to configure the genetic algorithm, remembering the "no free lunch theorem", there is no universally optimal set of parameters that works across different problems and system setups. The tuning was guided by structured experimentation and domain-specific insight. The goal was to identify a balance between exploration and exploitation in the algorithm, while maintaining reasonable run-times.

The most influential parameters were identified based on the shapes of the resulting setpoint curves. These were subsequently fine-tuned to ensure that various local optima and regions of the solution space were thoroughly explored.

3.7 Scenario evaluations

To validate the optimization algorithm, different scenarios were evaluated to confirm its function and validity. These scenarios are based on hydraulic models described in Section 3.2.

3.7.1 NET-1

Parallelization test

In this test the effect of parallelization was tested on the optimization algorithm using the NET-1 hydraulic model. Due to the available hardware, 6 processes were available to be run in parallel.

Net-1 Simple optimization scenario

In this scenario a simple artificial optimization scenario is introduced to provide meaningful insight in how the algorithm operates. The optimization algorithm can essentially be described as a "grey box" model. This means that it incorporates elements from white-box models, which are fully based on known physical laws and equations. But also contains elements of a black-box model, which are completely data driven. For a more complex hydraulic scenario, with different constraints, the reasoning of the model can be quite hard to follow. This scenario aims to use fewer objectives to provide a better understanding of how the algorithm operates.

In this scenario a special tariff period is introduced to the NET-1 hydraulic scenario. These tariff periods consist of 3 different periods: high, medium and low. Also, a general initial set-point curve was used as a starting point for the optimization algorithm. The tariff periods and the initial (unoptimized) setpoint-curve can be seen in figure 3-14.

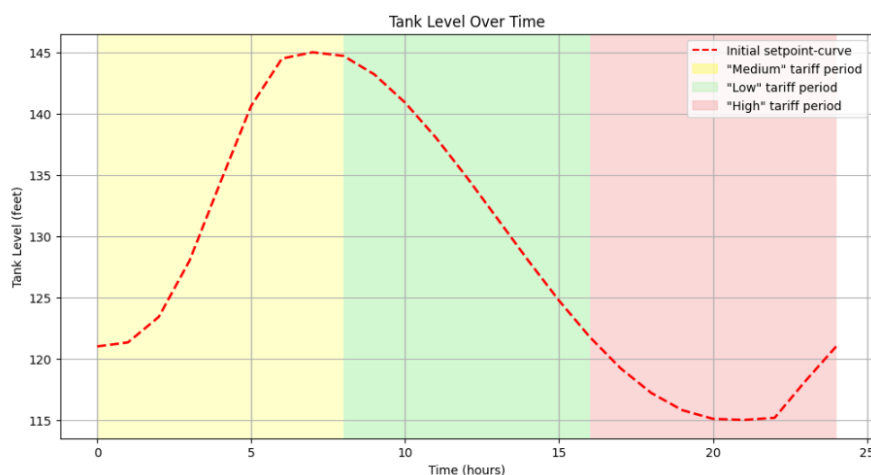


Figure 3-14 Tariff periods and initial (unoptimized) setpoint curve

It can be worth noting that this tariff scheme is not grounded in any actual data and thus holds no actual real-world validity. It is intentionally constructed so that the optimization scheme must make substantial changes to the initial setpoint-curve. providing clearer insight into its behaviour.

3.7.2 HPZ-G

Timestep Sensitivity Analysis

In order to evaluate how different timestep setting affect runtime and accuracy, a simple sensitivity analysis is done. The goal is to identify a suitable balance between efficiency and quality, by evaluating each timestep individually. As mentioned, the rule timestep cannot be explicitly set, and is instead calculated from the hydraulic timestep. It is therefore not included in the sensitivity analysis.

A high-resolution baseline scenario was established using low values for all the relevant timesteps. This baseline scenario was meant to represent a high accuracy simulation and served as a reference point for all the comparisons. The accuracy is determined by the energy cost calculation. The values used for the baseline scenario can be seen in table 3-3.

Table 3-3 Timestep variables for "baseline scenario"

Timestep:	Hydraulic	Reporting	Interpolation
Value:	5	10	10

Each timestep variable was then increased individually, enabling evaluation of how each variable affected simulation efficiency and deviations in simulation result. Certain timestep variables are dependent on others and cannot be changed completely independently. For example, the hydraulic timestep always need to be higher and divisible by than the reporting timestep. For these analysis's, the baseline value was increased, so that the isolated effect of the variable in focus could be seen.

Base cases

A 24-hour simulation was run for each day of focused period. To keep the results consistent the hyperparameters which were used were kept constant. Penalty factors had to be slightly adjusted during some days due to drastically changing electricity prices. The "Security of supply" penalty function was calibrated based on the current control scheme performed. If the penalty is violated any more than how the current control scheme would perform, a severe punishment is provided. This essentially means that each solution will have at least the same amount of demand security as the current setpoint curve.

The other penalty functions were tuned according to reasoning provided in the objective function chapter. Due to certain uncertainties regarding validity of the model, which can be seen in the results section, the clipping range of the setpoint curve was not changed from the base curve. There most likely is potential to utilize a greater portion of the reservoir, but since how these pressure levels would affect the rest of the system is unclear, they are not changed for the base case scenarios.

4. Data

The algorithm essentially is a simulation-based optimization, with data being used to as input and to calibrate and validate the model. The validation-data used was extracted from the SCADA-system and mainly consisted of pressure and flow measurements.

4.1 Electricity price

Historical data for electricity prices was taken from ENTSO-E which is a platform for collection and publication of pan-European wholesale market data (ENTSO-E, 2025). The data which was used was the “Day-ahead Prices” for price zone SE-3 in Sweden. As previously mentioned, the relevant period was from 2022-11-07 to 2022-11-14.

4.2 Suction pressure from the general WDS

Even though the pressure level in the general distribution is relatively stable, some daily variation can be seen. The actual suction pressure (Pressure entering the pump stations) was captured across the relevant timespan. An hourly resolution was deemed to be sufficient since most changes are relatively slow.

4.3 Hyperparameters

In table 4-1 to 4-5 a full list of all relevant hypermeters can be seen. These are related both to the optimization algorithm and the hydraulic simulation.

Table 4-1 General GA Settings

Name	Description
POPULATION_SIZE	Number of individuals per generation
MAX_GENERATIONS	Number of generations
HALL_OF_FAME_SIZE	Number of elite individuals to retain
P_CROSSOVER	Probability of applying crossover
P_MUTATION	Probability of applying mutation
TOURN_SIZE	Selection tournament size

Table 4-2 Penalty variables

Name	Description
PUMP_THRESHOLD	Max allowed pump switches before penalty
PENALTY_FACTOR_PS	Penalty multiplier for switch violations
PENALTY_FACTOR_MM	Penalty for minimum margin violation
PENALTY_FACTOR_SSV	Penalty for security of supply violation
PENALTY_FACTOR_FR	Penalty for final tank level deviation

Table 4-3 Crossover weighting variables

Name	Description
BLOCK_WEIGHTH_TIME_LOC	Odds of switching “time-loc” block between parents
BLOCK_WEIGHTH_HEIGHT	Odds of switching “height” block between parents
BLOCK_WEIGHTH_CURV	Odds of switching “curvature” block between parents

Table 4-4 Timestep variables

Hydraulic timestep	See chapter “timesteps”
Reporting timestep	-II-
Interpolation interval timestep	-II-

Table 4-5 Mutation probability variables

Name	Description
MUTATE_PROB_0	Chance of L_0 to be mutated when P_MUTATION is triggered.
MUTATE_PROB_1	Chance of H_P to be mutated when P_MUTATION is triggered.
MUTATE_PROB_2	Chance of T_P to be mutated when P_MUTATION is triggered.
MUTATE_PROB_3	Chance of W_P to be mutated when P_MUTATION is triggered.
MUTATE_PROB_4	Chance of T_D to be mutated when P_MUTATION is triggered.
MUTATE_PROB_5	Chance of H_D to be mutated when P_MUTATION is triggered.
MUTATE_PROB_6	Chance of W_D to be mutated when P_MUTATION is triggered.
MUTATE_PROB_7	Chance of CRV_1 to be mutated when P_MUTATION is triggered.
MUTATE_PROB_8	Chance of CRV_2 to be mutated when P_MUTATION is triggered.
MUTATE_PROB_9	Chance of CRV_3 to be mutated when P_MUTATION is triggered.

Table 4-6 Mutation Clip bound variables

Name	Description
MUTATE_RANGE_0	How much L_0 can be mutated in each direction
MUTATE_RANGE_1	How much H_P can be mutated in each direction
MUTATE_RANGE_2	How much of T_P can be mutated in each direction
MUTATE_RANGE_3	How much W_P can be mutated in each direction
MUTATE_RANGE_4	How much T_D can be mutated in each direction.
MUTATE_RANGE_5	How much H_D can be mutated in each direction.
MUTATE_RANGE_6	How much W_D can be mutated in each direction.
MUTATE_RANGE_7	How much CRV_1 can be mutated in each direction.
MUTATE_RANGE_8	How much CRV_2 can be mutated in each direction.
MUTATE_RANGE_9	How much CRV_3 can be mutated in each direction.

5. Results

5.1 Validating the HPZ-G hydraulic model

5.1.1 Pump curve calibrations

Pump station 1

In figure 5-1 the operating point for single and two pump operation can be seen.

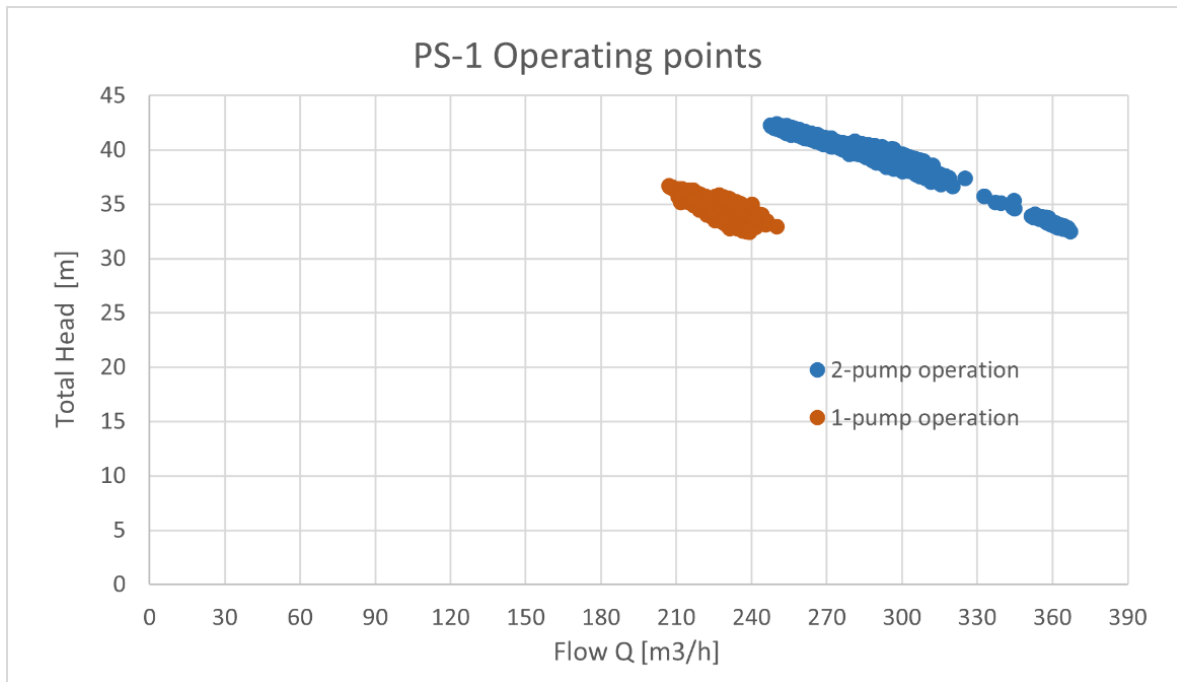


Figure 5-1 Operating points for pumps in Pump Station 1 for the HPZ-G network.

The flow correlation between the different operating points does not correlate 2:1, which would have been the case if the pumps were at the same capacity. To account for this, the flow values of the second pump were adjusted to match the difference in flows between 1 and 2 pump operation. The pump curves which were produced through polynomial regression can be seen in figure 5-2.

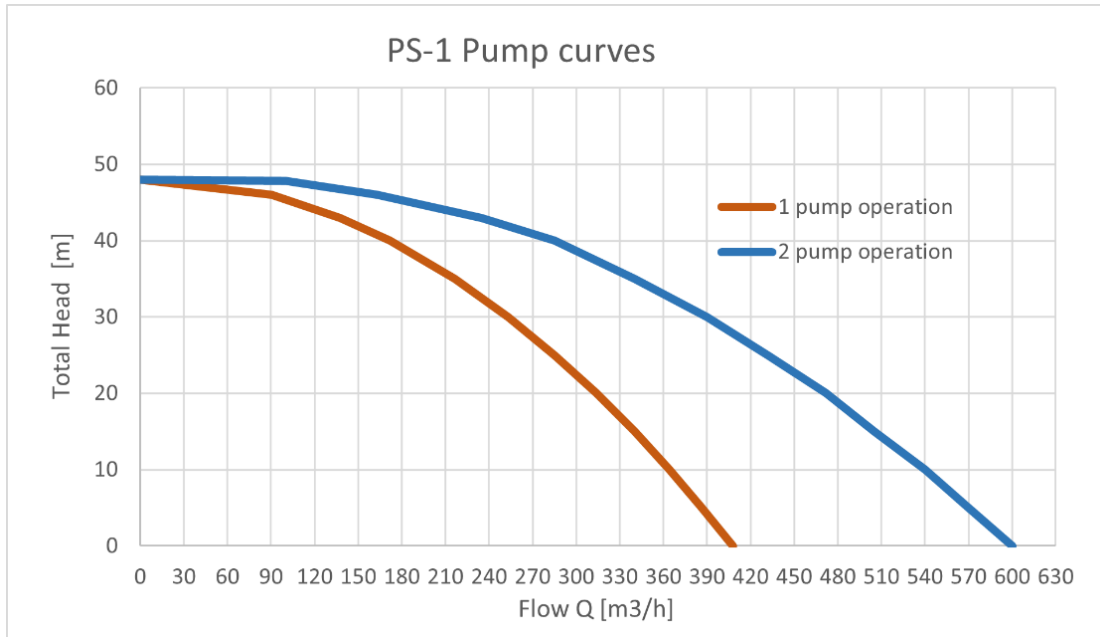


Figure 5-2 Recreated pump curves based on actual operating points in Pump Station 1

Pump station 2

In figure 5-3 the operating point for single and two pump operation during the period 2022-11-07 to 2022-11-14 can be seen. Here the flow correlation is 2:1, suggesting 2 pumps operating at the same capacity.

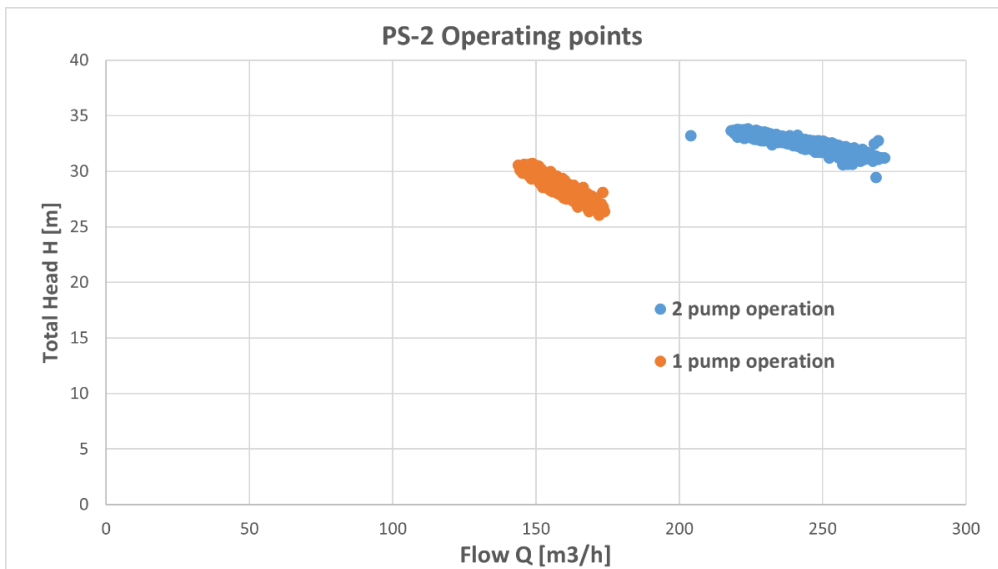


Figure 5-3 Operating points for pumps in Pump Station 1 for the HPZ-G network.

The corresponding pump curves for pumps station 2 can be seen in figure 5-4.

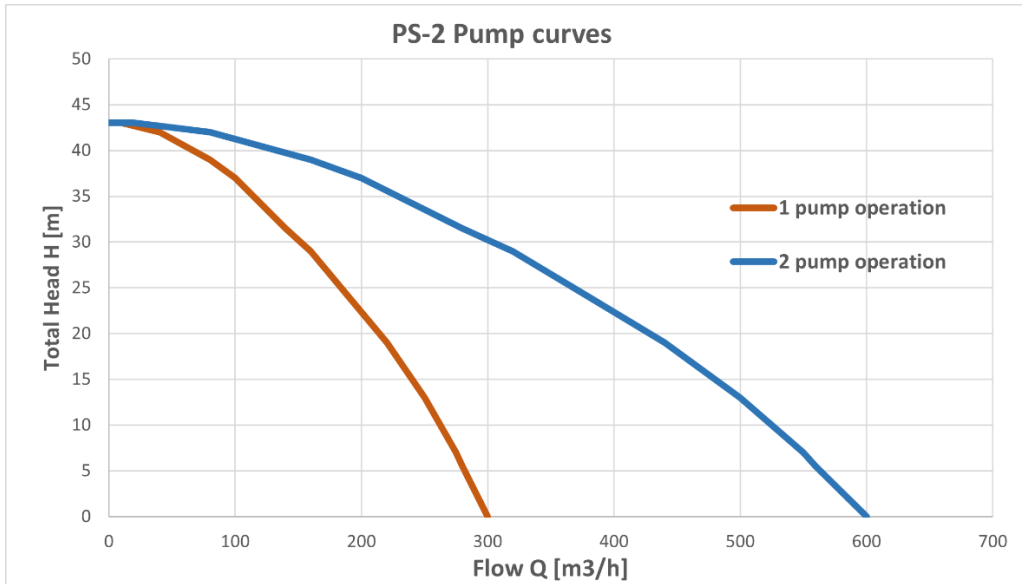


Figure 5-4 Recreated pump curves based on actual operating points in pump station 2

5.1.2 Simulating current control scheme

In this section the result from simulation is compared to the actual data from the real network during the same time. The entire 7-day period was simulated in 24-hour intervals. Only data from the first day, 2024-11-07 will be shown in the results section as it was deemed sufficiently representative of the accuracy of the entire period.

Water Tank Level

Figure 5-5 shows a comparison between the simulated WST level and the actual water level for the selected 24-hour period. The simulated curve closely follows the measured values throughout most of the day. Minor deviations can be seen, mainly around the peak period.

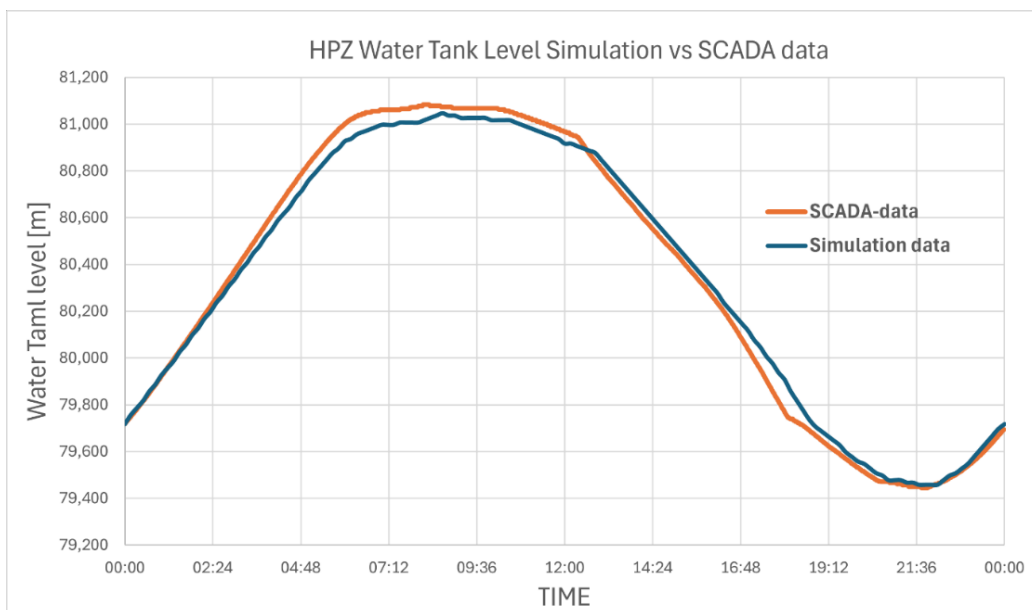


Figure 5-5 Actual WTL measured in the simulation and from real-world data.

Pump station 1

Flow

The outgoing flow from pump station 1 can be seen in figure 5-6. The simulation manages to capture the main operation patterns. Some deviations can be seen regarding the exact flow conditions and exact timing of pump switches.

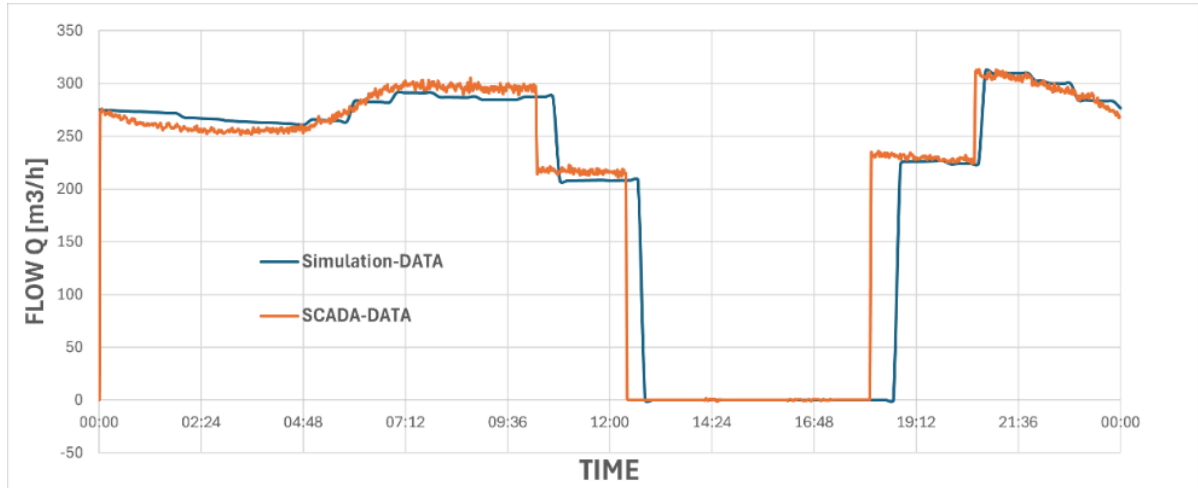


Figure 5-6 Outgoing flow from PS-1 measured in the simulation and from real-operation data.

Outgoing pressure

For the outgoing pressure shown in Figure 5-7, the largest deviations between the simulation and the measured data can be seen. The most significant difference occurs during periods when no pumps are operating, and the pressure is determined solely by the WTS. This pattern might suggest that the hydraulic model may underestimate the pressure losses between the WTS and the pumping station. However, it is difficult to draw definitive conclusions.

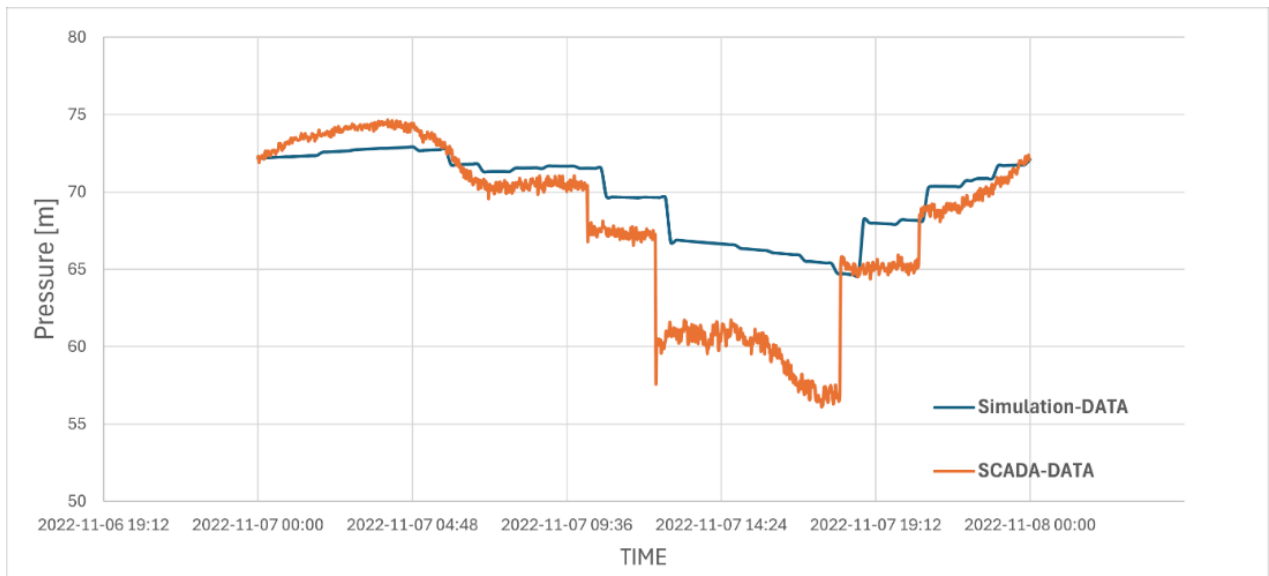


Figure 5-7 Outgoing pressure from PS-1 measured in the simulation and from real-world data.

Pumps in Action

Figure 5-8 shows the number of active pumps over the 24-hour period, comparing simulation results to real data. Overall, the simulation accurately replicates the timing and sequence of pump operations, with only minor discrepancies in the exact timing.

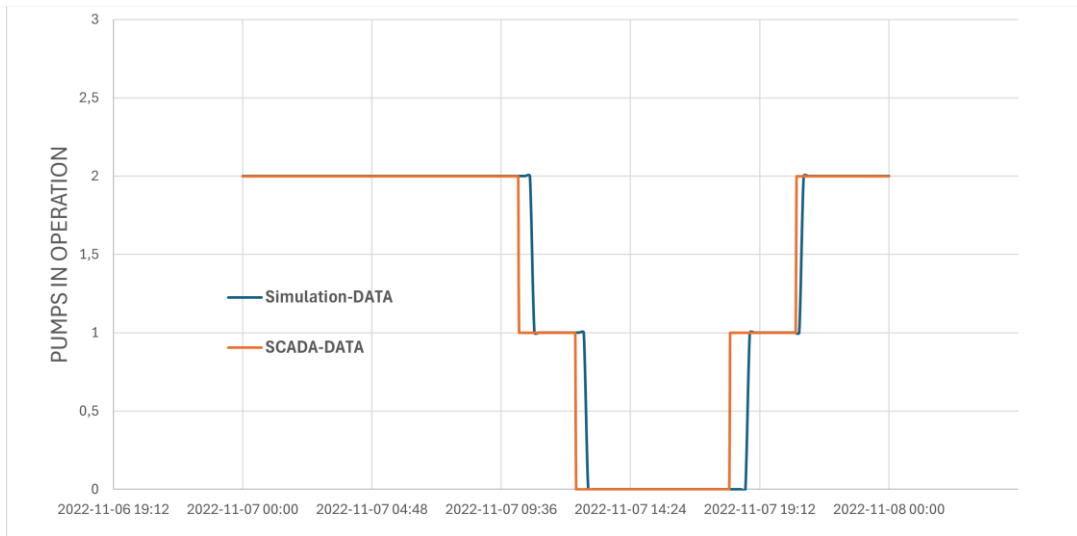


Figure 5-8 Active pumps in the simulation and during the real-world scenario.

5.1.3 Pump station 2

Plots for flow, outgoing pressure and pumps in operation can be found in the Appendix.

5.2 Net1

5.2.1 Parallelization

The simple Net1 model was used to evaluate the effect parallelization had on runtime. In figure 5-9 the effect on complete runtime can be seen

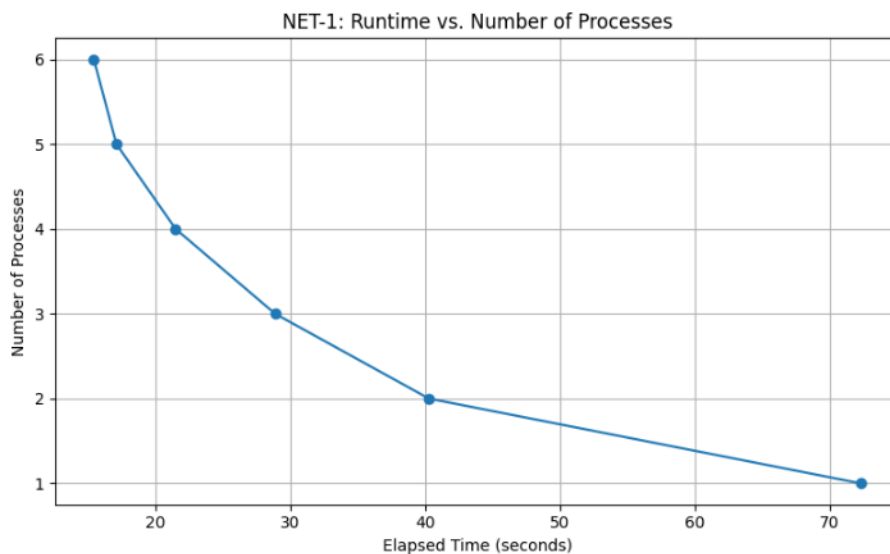


Figure 5-9 Runtime as a function of the number of processes used for parallel evaluation based on Net-1 optimization.

5.2.2 Simple optimization scenario

The graphs in this section provide the results for the “simple optimization scenario” described in method section. In figure 5-10 the population of each generation can be seen.

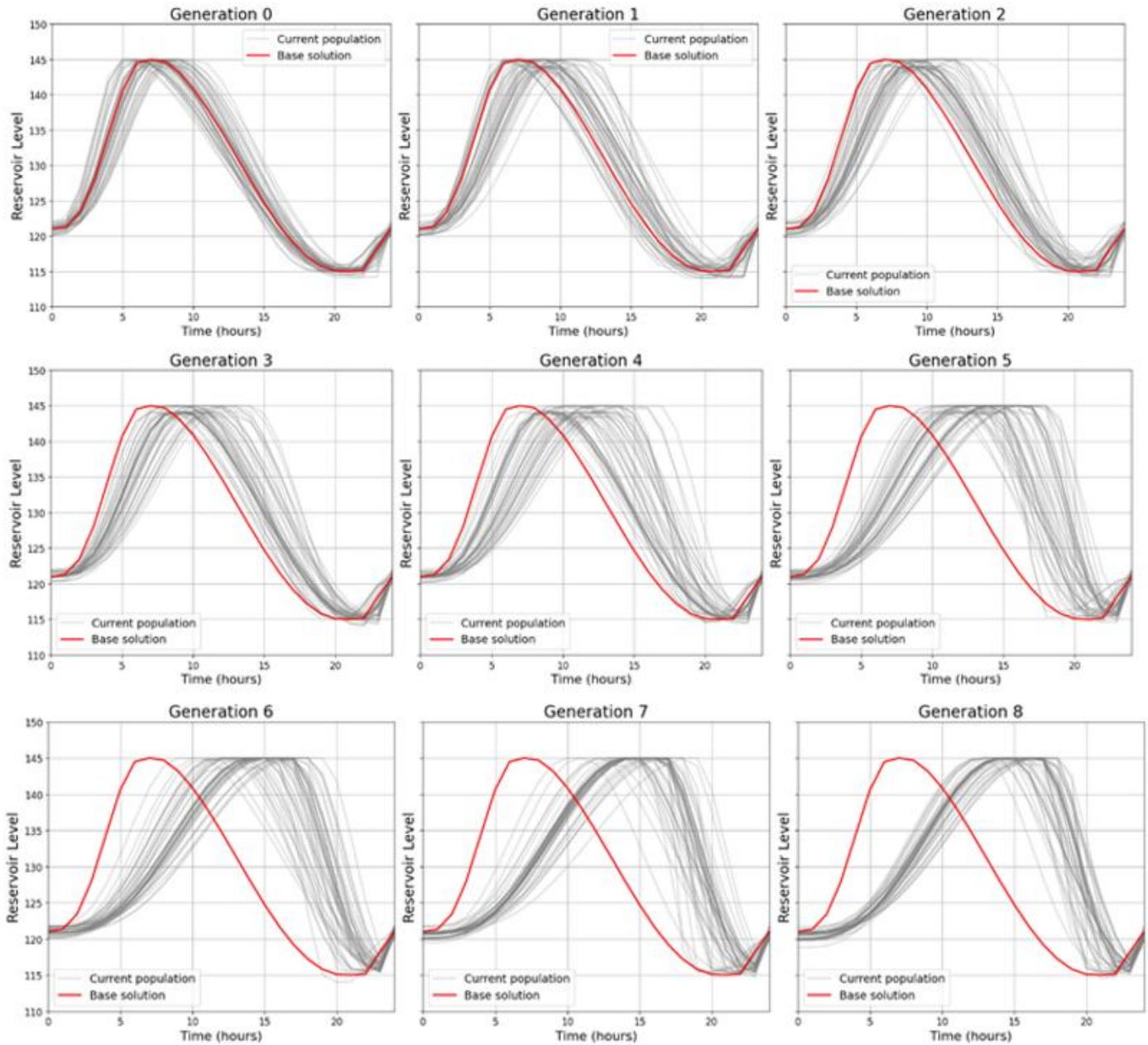


Figure 5-10 Evolution of reservoir level set-point curves over nine generations in the genetic algorithm.

The optimization algorithm here very clearly reproduces the expected pattern, shifting the peak towards the end of the low tariff period, whilst also implicitly delaying the main fill up period of the WST. In figure 5-11 the initial setpoint-curve and the new optimized curve are seen.

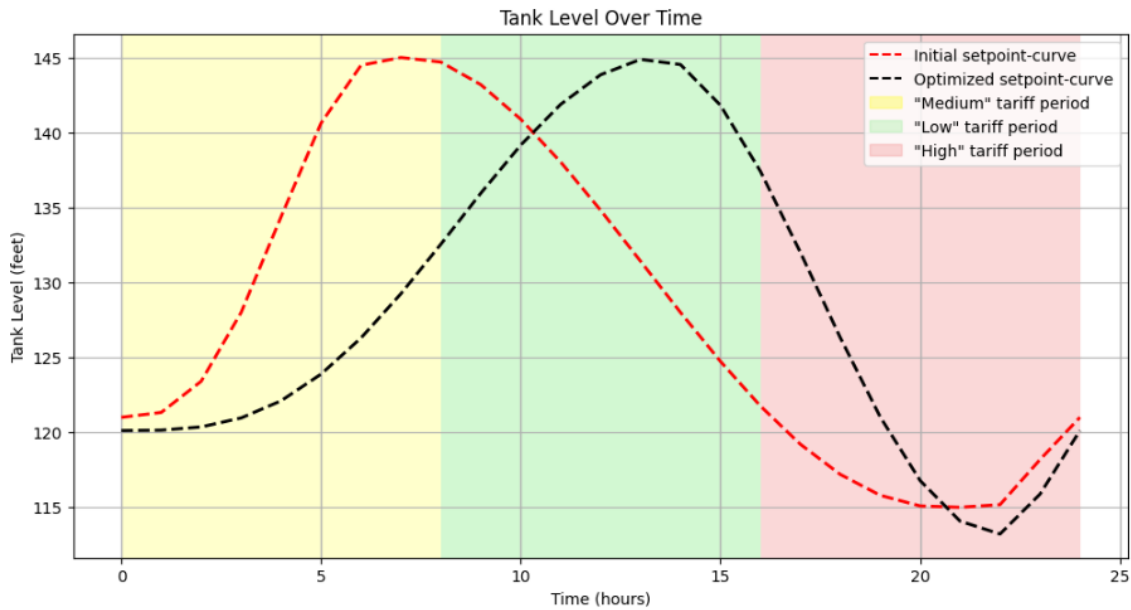


Figure 5-11 Comparison of initial and optimized set-point curves

The objective of the algorithm in this scenario was essentially to minimize the cost of electricity while maintaining hydraulic feasibility, initial/final volume balance, and the restriction which separates the dip and the peak.

5.3 HPZ-G

5.3.1 Timestep Sensitivity Analysis

Reporting timestep

The bar chart in figure 5-12 shows the percentage decrease in optimization runtime and the percentage change in energy cost accuracy as the reporting interval is increased from 20 min to 40 min and 60 min, relative to the 10 min baseline. A very slight decrease of runtime can be seen, with a bit more distinct decrease of the accuracy.

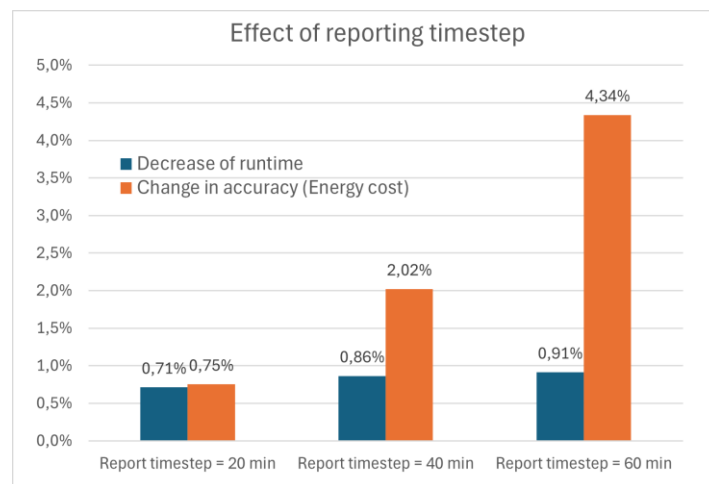


Figure 5-12 Impact of reporting timestep on average runtime and energy cost

Hydraulic timestep

The chart in figure 5-13 shows the percentage reduction in runtime and the change in energy cost accuracy when increasing the hydraulic timestep from 5 minutes (baseline scenario) to 15 and 30 minutes. A sharp decline in the runtime can be seen, without significantly affecting the calculated energy cost.

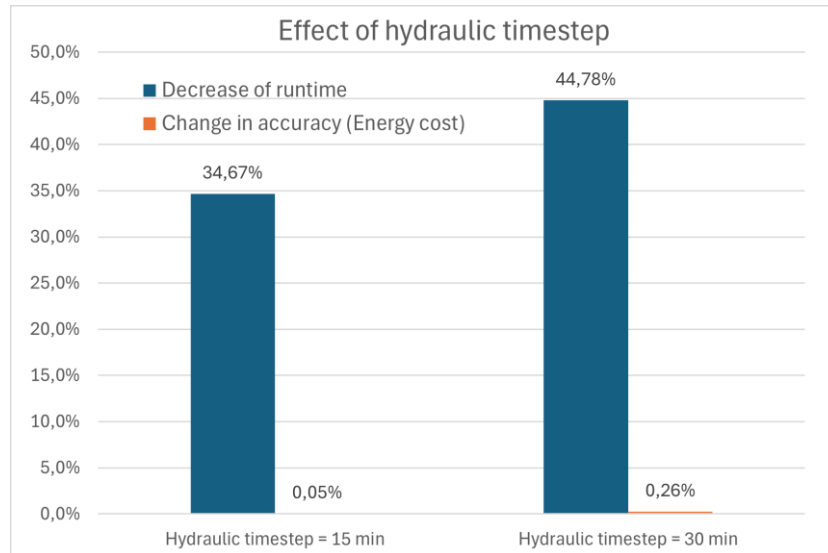


Figure 5-13 Impact of hydraulic timestep on average runtime and accuracy

Interpolation timestep

The chart in figure 5-14 shows the percentage decrease in runtime and the change in energy cost accuracy when the interpolation timestep is increased to 20 and 40 minutes, relative to the baseline configuration of 10 minutes. Similarly to the hydraulic timestep, a sharp decline in the runtime can be seen, without significantly affecting the calculated energy cost.

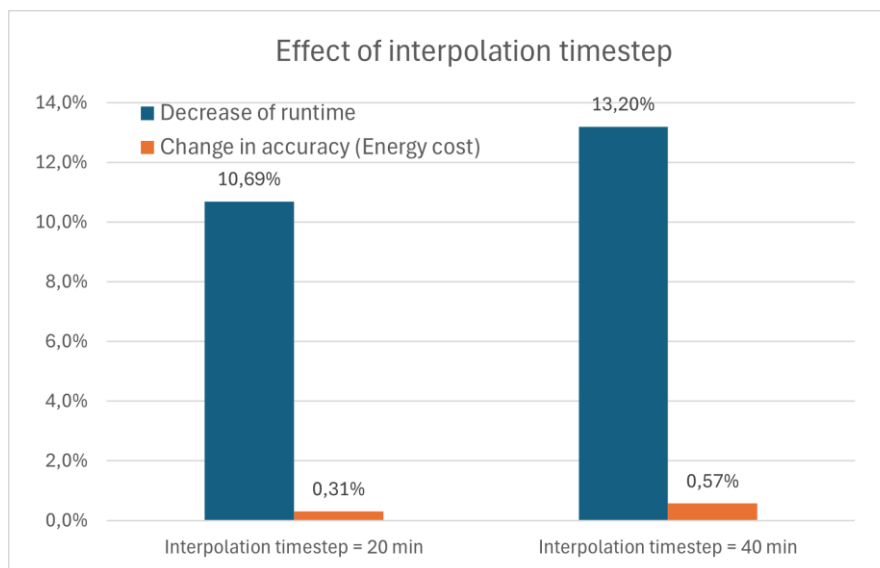


Figure 5-14 Impact of interpolation timestep on average runtime and energy cost

5.3.2 Main HPZ-G optimization results

In table 5-1 the potential energy cost savings can be seen for the entire evaluated timeframe. Whilst these results provide an indication of potential saving, it is important to consider the reflections about electricity as surrogate variable, as discussed in objective function section.

Table 5-1 Summary potential cost savings during the evaluated timeframe.

Date	Weekday	Cost before [SEK*]	Cost after [SEK*]	Decreased cost [%]
2022-11-07	Monday	510,7	471.45	7.68
2022-11-08	Tuesday	662.26	616.27	6.94
2022-11-09	Wednesday	445.41	421.98	5.26
2022-11-10	Thursday	538.30	454.88	15.48
2022-11-11	Friday	24.24	20.42	15.77
2022-11-12	Saturday	23.40	21.39	8.6
2022-11-13	Sunday	679.17	649.43	4.38

*Not actual cost as discussed in section 1.3.4.

More actual reservoir levels are presented for the dates 2022-11-07 (a Monday) and 2022-11-12 (a Saturday), as these were found to be representative of typical weekday and weekend conditions. The distinction was made since both electricity and water demand typically follow different patterns on weekdays compared to weekends. Electricity data and water demand data can be found in Appendix A and Appendix B, respectively.

5.3.3 Scenario 2022-11-07

In figure 5-15 the actual reservoir level for the current control scheme can be seen, along with the tank level for the optimized scenario (both simulated).

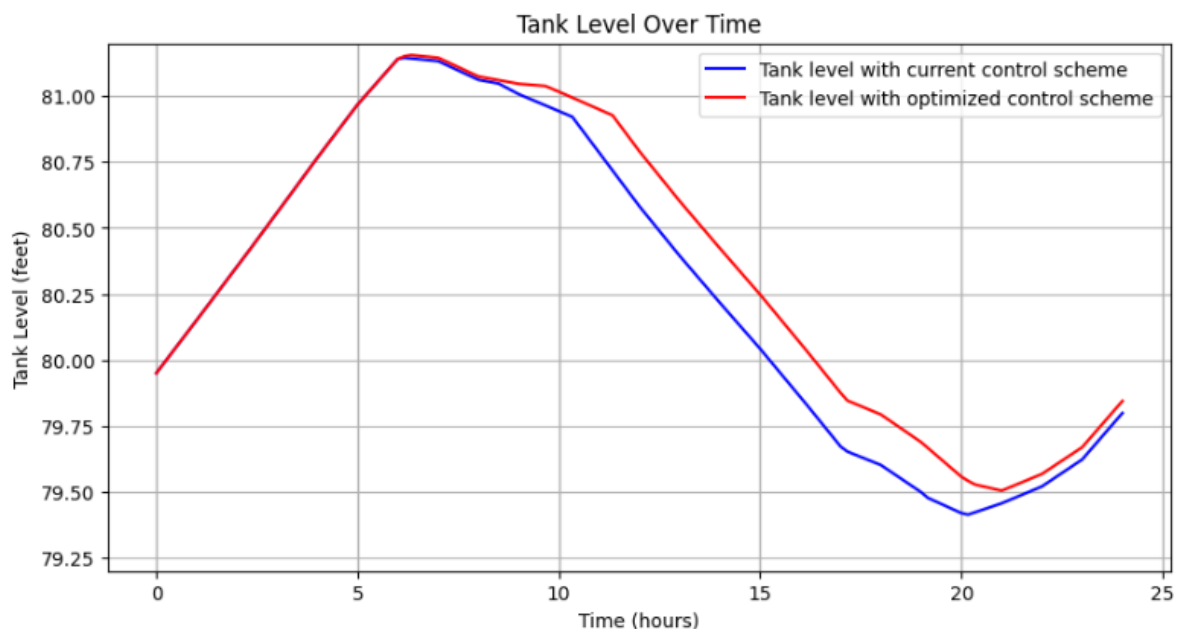


Figure 5-15 Optimized reservoir level and base solution (current control scheme) for the 24-hour time period: 2022-11-07.

In table 5-2 the final genotype for the optimized 2022-11-07 scenario can be seen.

Table 5-2 Key-parameter encoded genotype for optimized setpoint curve

Parameter	Variable name	Value
Baseline Level	L_0	79.95
Peak Height	H_P	81.1
Peak Timing	T_P	5.162
Peak Width	W_P	2.554
Dip Timing	T_D	19.384
Dip Height	H_D	79.912
Dip Width	W_D	0.704
Rise Curvature	CRV_1	-0.297
Descent Curvature	CRV_2	-0.327
Recovery Curvature	CRV_3	0.311
Pump trigger range	pt_var	0.424

5.3.4 Scenario 2022-11-13

In figure 5-16 the actual reservoir level for the current control scheme can be seen, along with the tank level for the optimized scenario.

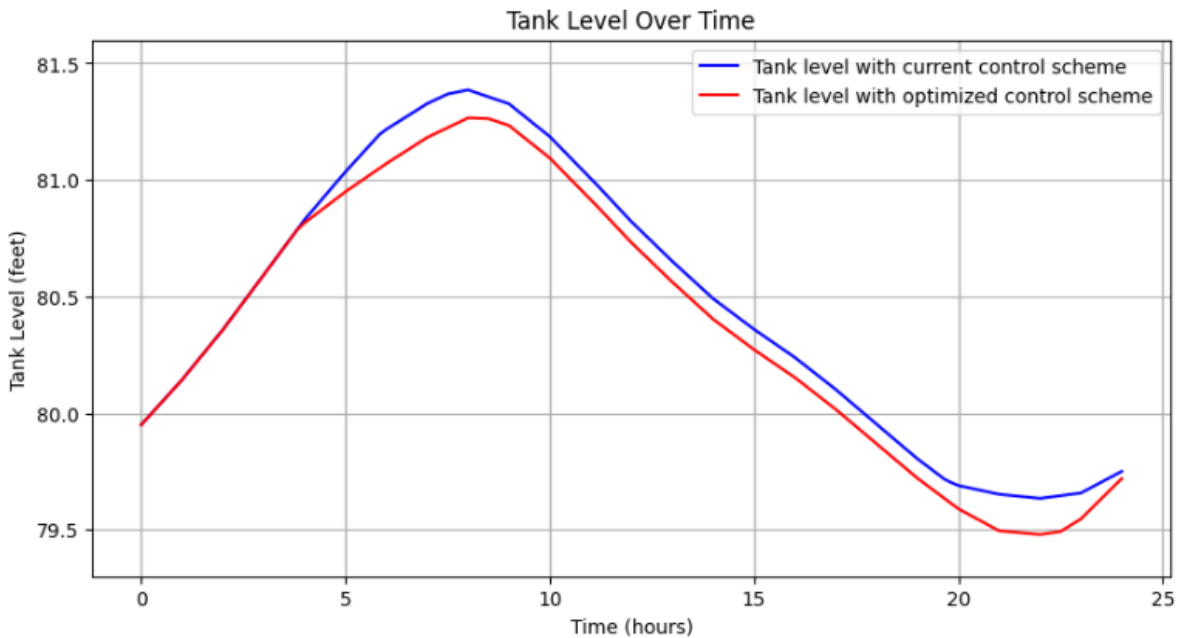


Figure 5-16 Optimized reservoir level and base solution (current control scheme) for the 24-hour time period: 2022-11-13.

In table 5-3 the final genotype for the optimized 2022-11-13 scenario can be seen.

Table 5-3 Key-parameter encoded genotype for optimized setpoint curve

Parameter	Variable name	Value
Baseline Level	L_0	79.95
Peak Height	H_P	80.991
Peak Timing	T_P	4.21
Peak Width	W_P	2.787
Dip Timing	T_D	18.109
Dip Height	H_D	79.65
Dip Width	W_D	1.222
Rise Curvature	CRV_1	-0.208
Descent Curvature	CRV_2	-0.006
Recovery Curvature	CRV_3	-0.332
Pump trigger range	pt_var	0.19

6. Discussion

6.1 Control scheme/Real network

A clear observation is that the actual WST level in the real system deviates significantly from the setpoint curve. This discrepancy is partly due to the substantial inertia present in larger water distribution systems (WDS). Additionally, it stems from how the pump trigger ranges are configured in relation to the setpoint curve. Implementing stricter on/off ranges for the pumps would reduce the deviation between the actual reservoir level and the desired set-point. Due to the significant variability in water demand and electricity prices, the current pump trigger ranges essentially limit the capabilities of the setpoint curve optimization.

The results from the HPZ-G optimization does however indicate that the optimal setpoint curve will look widely different depending on variation of demand and energy price. Thus, using a single setpoint curve, which is the case in the current scheme, is inefficient in adapting to these changes.

6.1.1 Validity of hydraulic model

Some very clear deviations can be seen between the simulation and the real-world data, particularly related to pressure values. These are likely attributable to discrepancies between the hydraulic model and the actual network. To obtain a model with greater accuracy, calibrations of roughness coefficients of pipes, valve settings and other aspects of the network would have to be made. The accuracy of the hydraulic model would be particularly important when analyzing changes which utilize a larger portion of the reservoir volume. This could lead to pressure conditions which are not present in the current system.

6.1.2 Validity of Energy data

To perform more accurate cost/energy use evaluations, more accurate energy data would need to be made available. Measurements regarding how the pumps perform during different operating condition would be essential for generating accurate optimization results.

6.1.3 Objective function development

Since the objective function aims to express everything in the same unit (SEK). A more in-depth discussion will need to be had in regards on the weight of the different components in the objective function. For example: How long should a zone be able to handle the demand if the incoming water supply is cut, and what is that worth in SEK? Also, since utilizing stricter pump triggers indicates great potential in decreasing energy usage, a more in-depth investigation would need to be had into the correlation of maintenance cost and pump switches.

6.2 Optimization algorithm

6.2.1 Shortcoming of the algorithm

Looking at the results from the NET-1 model it can clearly be seen that the model can perform in context of exploration and exploitation. Even though these results are not applicable to a real-world scenario it illustrates the model's capabilities in navigating the solution space.

The optimization scenarios based on HPZ-G indicated some of the shortcomings of the model in its current state. Whilst the key-parameter encoding offers an efficient and simple representation of the setpoint curve, it could use some additional gene parameters to introduce more degrees of freedom. More degrees of freedom would naturally increase the optimization potential. For instance, the

inclusion of a secondary peak in the setpoint curve can better reflect specific water demand and electricity price patterns. I will thus give provide better potential to capture.

It is, however, important to remember why setpoint curves are used from the beginning: to create a general, stable solution which can handle deviations from the predicted demand. As data is updated more frequently and predictions become more accurate, a more explicit pump scheduling scheme might prove to be effective.

6.2.2 Pump trigger ranges

The results indicate the increased potential when stricter pump triggers ranges are introduced. This suggests that including the pump trigger ranges as an optimization parameter is necessary. As previously noted, it is important to account for the resulting increase in pump switches and to carefully balance the associated pump switch penalty to maintain an effective trade-off.

6.2.3 Timestep variables

The sensitivity analysis of timestep variables was essentially performed to obtain a better understanding for which parameter could be relaxed to speed up convergence, without significantly impacting the accuracy. Ideally all time-step variables would be kept as strict (low) as possible to maximize simulation accuracy. However, this was not possible due to computational demands and time constraints on the project. Each run of the optimization algorithm can result in hundreds of hydraulic simulations, making optimizing performance of these hugely important.

The results show that certain timestep variables, particularly the hydraulic and interpolation intervals, had a substantial impact on runtime. It also indicated that changing the reporting timestep had the largest effect on the energy calculation. It is important to distinguish that the hydraulic simulation remains unchanged when the reporting timestep is adjusted. Instead, the difference occurs due to the optimization algorithm receiving the simulation results at a higher resolution, allowing for more precise energy calculations.

6.2.4 Parallelization

The results clearly indicate that the optimization algorithm massively benefits from parallelization. This is because most of the computationally intensive process is the evaluation function, which obviously derives from the hydraulic simulation. Some of the diminishing returns are because evaluation portion represents a lesser portion of the entire algorithm as it speeds up. However, for more complex networks, such as the HPZ-G simulation, this effect was barely noticeable because evaluation portion still represented most of the algorithm runtime, even at full parallelization. Thus, the increase HPZ-G simulation could be described by a near linear correlation. In this thesis the computational limitations were set at 6 processes. In theory, if more computational power were available, the process could speed up even more.

7. Conclusion and further development

In this final section the conclusion will be presented along with few recommendations in the context of further improving the general optimization algorithm and improving the current control scheme utilized at KoV.

7.1 Conclusion

- The key-parameter encoded Genetic Algorithm (GA) successfully generated optimized set-point curves which were compatible with the existing SCADA-based control scheme.
- Optimization runs on the HPZ-G network demonstrated energy-cost reductions of around 10 % compared to the current control logic.
- Introducing stricter pump-trigger thresholds would reduce level deviations and improve the alignment between optimized and actual operation.
- Using a single static set-point curve for all conditions was found inefficient. Optimal operation differs markedly between days due to differing water demand and electricity-price profiles.

Overall, the study demonstrates that a set-point curve optimization framework offers a practical and interpretable path toward energy-efficient operation, bridging the gap between rule-based and fully optimized pump scheduling.

7.2 Further improvement for optimization model

The main work needed for improving the optimization model is to further develop the encoding scheme. As previously noted, the objective is to transform the setpoint curve into a compact, interpretable genotype that the genetic algorithm can efficiently manipulate. Each added parameter will enable the capture of more subtle variations in reservoir behaviour, but it also expands the search space, making convergence slower and increasing processing demand. The natural development for the current “key-parameter” encoding scheme would be to add additional shape parameters which would allow the representation of more complex demand patterns, such as curves with multiple peaks. In its current form, the pump trigger thresholds are represented by a single variable derived from the initial control scheme. This could be replaced by individual trigger ranges for each pump, adding two variables to the genotype per pump. While this approach would increase the complexity of the model, it would also expand the optimization potential by allowing more tailored control for each pump.

An alternative to the key-parameter encoding is to represent the entire set-point curve as a periodic function as a sum of sine and cosine terms, more specifically as a truncated Fourier series. This truncated Fourier encoding was initially intended to be used in the optimization algorithm. But due to the difficulty of applying mutations/transformations, in combination with time-constraints, it was ultimately replaced by the simpler key-parameter encoding. A detailed description of the key-parameter encoding is provided in Appendix E.

In terms of wider the wider optimization model, added support for VSP would also be a necessary addition. Utilizing VSP enables pumps to run at a more efficient operating point, enables a more sophisticated algorithm, with a higher optimization ceiling. The model would benefit from additional

consideration of demand variation. This could be achieved by using multiple runs with different demand variations as a basis for the evaluation.

7.3 Recommendation of the current control scheme

Clear mismatches exist between simulation results and SCADA data, especially in pressure values → Model calibration is necessary to improve accuracy hydraulic model.

Current energy data is insufficient for detailed cost and energy usage analysis → Data on pump performance under different operating conditions is essential for accurate optimization.

Questions about how to weigh different aspects of the objective function:

- How long a zone should manage without inflow, what is the value of demand security?
- Trade-off between energy savings and increased maintenance from more frequent pump switches?
- What is the worth of grid stabilisations by shifting pump usage from peak periods?

Furthermore, the present control logic relies on a single, static set-point curve applied under all conditions. Optimization results from the HPZ-G application clearly indicate the benefit of using different set-point curves depending on the expected water demand and electricity price. A practical approach could be to generate a new set-point curve each day, using the most recent electricity prices and demand forecasts as inputs to the optimization process. Regularly updating the curve in this way can improve energy efficiency and increase operational flexibility

Utilizing narrower pump trigger ranges would further enhance the system's ability in utilizing these described variations. This is essentially confirmed by all optimized solutions having a $pt_var < 1$. The potential of incorporating variable speed pumps does also warrant further investigation but was outside the scope of this thesis.

References

- Abdelsalam, A. A., & Gabbar, H. A. (2021). Energy saving and management of water pumping networks. *Heliyon*, 7(8), e07820. <https://doi.org/10.1016/j.heliyon.2021.e07820>
- Alvisi, S., & Franchini, M. (2017). A robust approach based on time variable trigger levels for pump control. *Journal of Hydroinformatics*, 19(6), 811–822. <https://doi.org/10.2166/hydro.2017.141>
- Balekelayi, N., & Tesfamariam, S. (2017). Optimization techniques used in design and operations of water distribution networks: A review and comparative study. *Sustainable and Resilient Infrastructure*, 2(4), 153–168. <https://doi.org/10.1080/23789689.2017.1328921>
- Bunn, S. M., & Reynolds, L. (2009). The energy-efficiency benefits of pump-scheduling optimization for potable water supplies. *IBM Journal of Research and Development*, 53(3), 5:1-5:13. <https://doi.org/10.1147/JRD.2009.5429018>
- Creaco, E., Campisano, A., Fontana, N., Marini, G., Page, P. R., & Walski, T. (2019). Real time control of water distribution networks: A state-of-the-art review. *Water Research*, 161, 517–530. <https://doi.org/10.1016/j.watres.2019.06.025>
- Da Silva Brito, J. R., Lopes, R. V. V., Ferro, J. V. R., & De Barros Costa, E. (2023). A Crossover Operator Based on Building Blocks Preservation. *2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–6. <https://doi.org/10.23919/CISTI58278.2023.10211556>
- DHI. (2024). MIKE+ [Computer software]. DHI. <https://www.mikepoweredbydhi.com/>
- ENTSO-E. (2025). *ENTSO-E transparency platform*. European Network of Transmission System Operators for Electricity. Retrieved June 29, 2025, from <https://transparency.entsoe.eu/>
- Fortin, F.-A., Fortin, F.-A., Rainville, F.-M. D., De-Rainville, F.-M., Gardner, M.-A., Gardner, M.-A., Parizeau, M., Parizeau, M., Gagne, C., & Gagne, C. (2012). *DEAP: Evolutionary Algorithms Made Easy*.
- Giustolisi, O., Laucelli, D., & Berardi, L. (2013). Operational Optimization: Water Losses versus Energy Costs. *Journal of Hydraulic Engineering*, 139(4), 410–423. [https://doi.org/10.1061/\(ASCE\)HY.1943-7900.0000681](https://doi.org/10.1061/(ASCE)HY.1943-7900.0000681)
- Hagman, M., Pelin, V., Iggström, D., Lindblom, D., Possling, B., & Spännare, J. (2024). Dricksvattensystemets möjliga bidrag till stadens elförsörjning. *2024*.
- Housh, M., & Salomons, E. (2019). Optimal Dynamic Pump Triggers for Cost Saving and Robust Water Distribution System Operations. *Journal of Water Resources Planning and Management*, 145(2), 04018095. [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0001028](https://doi.org/10.1061/(ASCE)WR.1943-5452.0001028)
- Janga Reddy, M., & Nagesh Kumar, D. (2020). Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: A state-of-the-art review. *H2Open Journal*, 3(1), 135–188. <https://doi.org/10.2166/h2oj.2020.128>
- Janus, T., Ulanicki, B., & Diao, K. (2023). *Optimal Scheduling of Variable Speed Pumps Using Mixed Integer Linear Programming—Towards An Automated Approach* (No. arXiv:2309.04715). arXiv. <https://doi.org/10.48550/arXiv.2309.04715>
- Kramer, O. (2017). *Genetic Algorithm Essentials* (Vol. 679). Springer International Publishing. <https://doi.org/10.1007/978-3-319-52156-5>
- Kyriakou, M. S., Demetriades, M., Vrachimis, S. G., Eliades, D. G., & Polycarpou, M. M. (2023). EPyT: An EPANET-Python Toolkit for Smart Water Network Simulations. *Journal of Open Source Software*, 8(92), 5947. <https://doi.org/10.21105/joss.05947>

- Mala-Jetmarova, H., Sultanova, N., & Savic, D. (2017). Lost in optimisation of water distribution systems? A literature review of system operation. *Environmental Modelling & Software*, *93*, 209–254. <https://doi.org/10.1016/j.envsoft.2017.02.009>
- Marchi, A., Simpson, A. R., & Lambert, M. F. (2017). Pump Operation Optimization Using Rule-based Controls. *Procedia Engineering*, *186*, 210–217. <https://doi.org/10.1016/j.proeng.2017.03.229>
- Marini, G., Fontana, N., Maio, M., Di Menna, F., & Giugni, M. (2023). A Novel Approach to Avoiding Technically Unfeasible Solutions in the Pump Scheduling Problem. *Water*, *15*(2), 286. <https://doi.org/10.3390/w15020286>
- Nicklow, J., Reed, P., Savic, D., Dessalegne, T., Harrell, L., Chan-Hilton, A., Karamouz, M., Minsker, B., Ostfeld, A., Singh, A., Zechman, E., & ASCE Task Committee on Evolutionary Computation in Environmental and Water Resources Engineering. (2010). State of the Art for Genetic Algorithms and Beyond in Water Resources Planning and Management. *Journal of Water Resources Planning and Management*, *136*(4), 412–432. [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0000053](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000053)
- Quintiliani, C., & Creaco, E. (2019). Using Additional Time Slots for Improving Pump Control Optimization Based on Trigger Levels. *Water Resources Management*, *33*(9), 3175–3186. <https://doi.org/10.1007/s11269-019-02297-6>
- Rossman, L. A., Woo, H., Tryby, M., Shang, F., Janke, R., & Haxton, T. (2002). EPANET 2.2 User Manual. 2002.
- Salomons, E., & Housh, M. (2020). Practical real-time optimization for energy efficient water distribution systems operation. *Journal of Cleaner Production*, *275*, 124148. <https://doi.org/10.1016/j.jclepro.2020.124148>
- Sivanandam, S. N., & Deepa, S. N. (Eds.). (2008). *Introduction to genetic algorithms*. Springer.
- Svenskt Vatten. (2021). P114 Distribution av dricksvatten. Stockholm, Sweden: Svenskt Vatten AB.
- Tarek, D., Mahmoud, H. H., & Ismail, T. (2022). Energy Optimization and Cost Reduction in Water Distribution Networks. *2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 368–372. <https://doi.org/10.1109/CSNDSP54353.2022.9907950>
- TessaDEM. (2025.). *World elevation database: Near-global 30-meter digital elevation model (DEM)*. TessaDEM. Retrieved June 29, 2025, from <https://tessadem.com/>
- Thomas, M., & Sela, L. (2024). A Mixed-Integer Linear Programming Framework for Optimization of Water Network Operations Problems. *Water Resources Research*, *60*(2), e2023WR034526. <https://doi.org/10.1029/2023WR034526>
- Van Zyl, J. E., Savic, D. A., & Walters, G. A. (2004). Operational Optimization of Water Distribution Systems Using a Hybrid Genetic Algorithm. *Journal of Water Resources Planning and Management*, *130*(2), 160–170. [https://doi.org/10.1061/\(ASCE\)0733-9496\(2004\)130:2\(160\)](https://doi.org/10.1061/(ASCE)0733-9496(2004)130:2(160))

Appendix

A - Electricity prices

Table A-1 Electricity prices for 2022-11-07

Hour	Price [SEK]	Hour	Price [SEK]
0	1572,45	12	1103,52
1	1141,8	13	1074,48
2	1135,2	14	1087,57
3	1129,15	15	1091,2
4	1109,79	16	1177
5	1155,99	17	1619,75
6	1444,74	18	1697,41
7	1697,41	19	1756,92
8	1759,23	20	1644,94
9	1608,2	21	1380,83
10	1141,8	22	1139,71
11	1129,15	23	1122

Table A-2 Electricity price for 2022-11-13

Hour	Price [SEK]	Hour	Price [SEK]
0	128,37	12	462,88
1	143,88	13	426,58
2	160,82	14	389,07
3	179,63	15	421,19
4	190,19	16	1614,47
5	215,6	17	2063,71
6	245,85	18	1924,56
7	282,48	19	879,45
8	357,17	20	399,52
9	399,3	21	340,89
10	428,45	22	309,43
11	128,37	23	297,99

B - Water Demand data

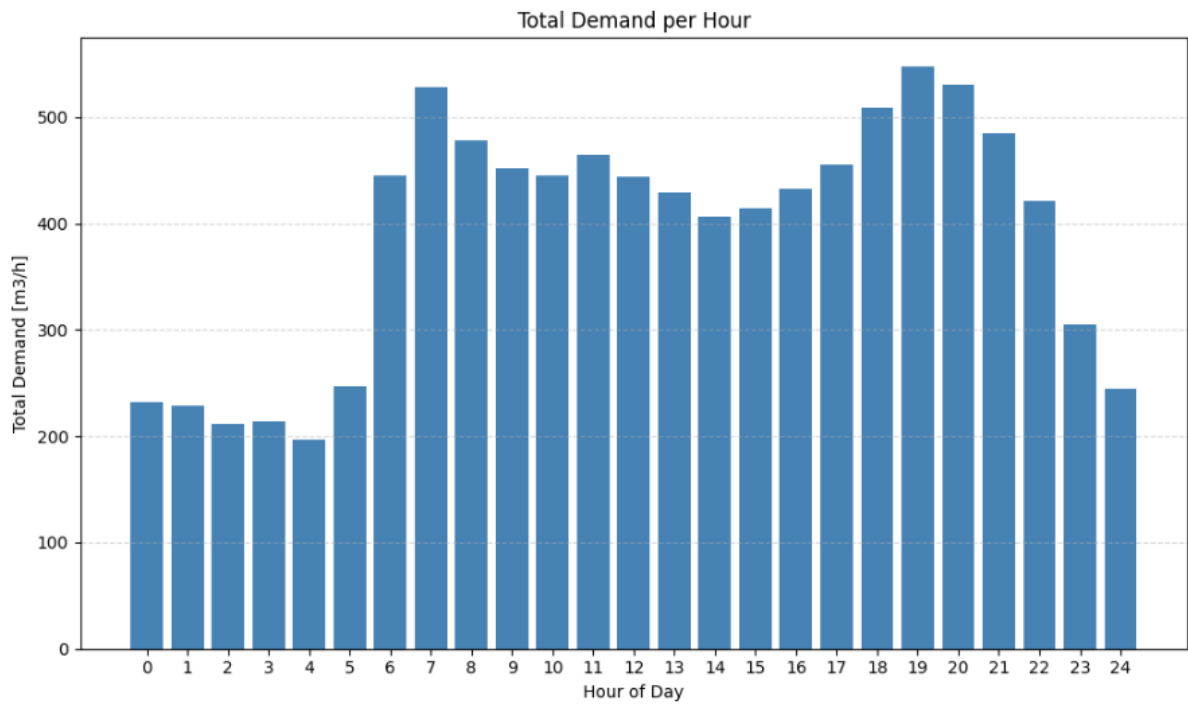


Figure B-1 Water demand per hour for 2022-11-07

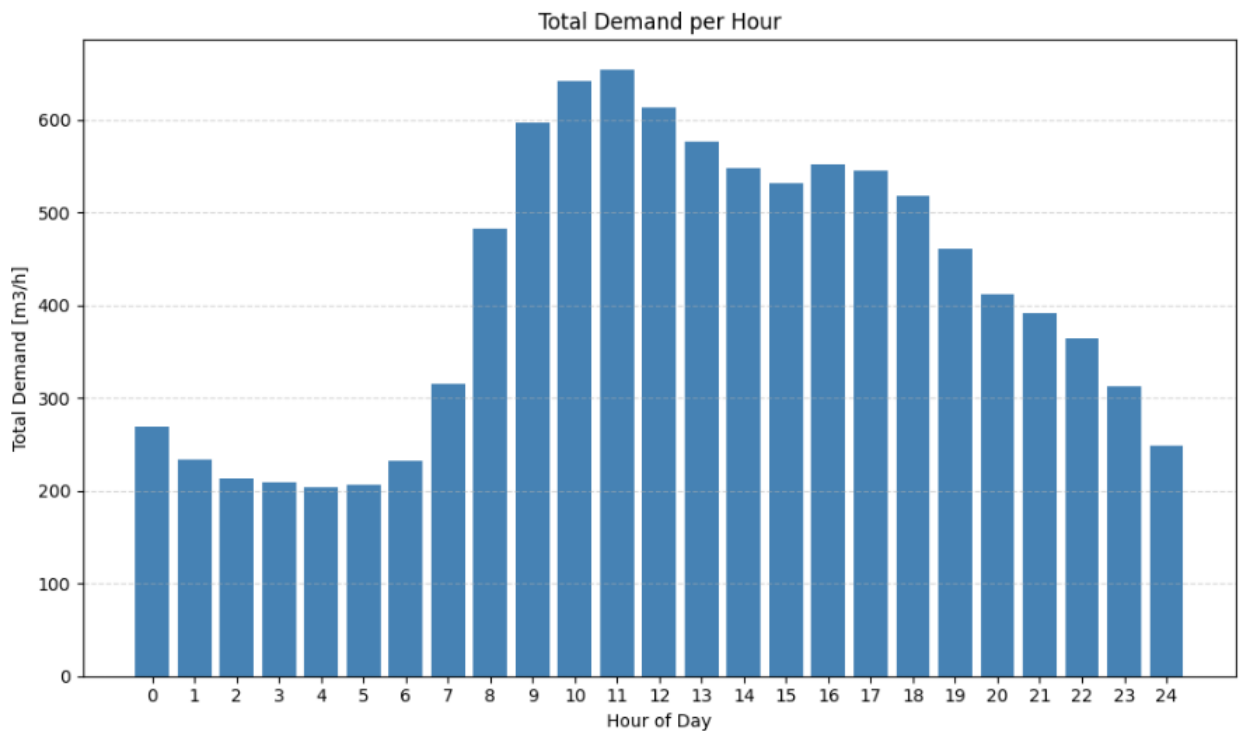


Figure B-2 Water demand per hour for 2022-11-13

C – General HPZ-G Run hyperparameters

Table C-1 Hyperparameters for general HPZ-G run.

General GA parameters:	
POPULATION_SIZE	24
MAX_GENERATIONS	17
HALL_OF_FAME_SIZE	6
P_CROSSOVER	0.4
P_MUTATION	0.8
TOURN_SIZE	3
Penalty Parameters:	
PUMP_THRESHOLD	10
PENALTY_FACTOR_PS	1
PENALTY_FACTOR_MM	5
PENALTY_FACTOR_SSV	0.05
PENALTY_FACTOR_FR	0.01
Crossover Parameters:	
BLOCK_WEIGHT_TIME_LOC	0.6
BLOCK_WEIGHT_HEIGHT	0.3
BLOCK_WEIGHT_CURV	0.2
Timestep Parameters:	
Hydraulic timestep	30 min
Reporting timestep	30 min
Interpolation interval timestep	30 min
Mutation Parameters:	
MUTATE_PROB_0	0
MUTATE_PROB_1	0.6
MUTATE_PROB_2	0.6
MUTATE_PROB_3	0.7
MUTATE_PROB_4	0.5
MUTATE_PROB_5	0.4
MUTATE_PROB_6	0.8
MUTATE_PROB_7	0.6
MUTATE_PROB_8	0.5
MUTATE_PROB_9	0.6
MUTATE_PROB_10	0.5
MUTATE_RANGE_0	-0.1 – 0.1
MUTATE_RANGE_1	-0.2 – 0.2
MUTATE_RANGE_2	-0.6 – 0.6
MUTATE_RANGE_3	-0.1 – 0.1
MUTATE_RANGE_4	-0.1 – 0.1
MUTATE_RANGE_5	-0.1 – 0.1
MUTATE_RANGE_6	-0.1 – 0.1
MUTATE_RANGE_7	-0.1 – 0.1
MUTATE_RANGE_8	-0.1 – 0.1
MUTATE_RANGE_9	-0.1 – 0.1
MUTATE_RANGE_10	-0.1 – 0.1

D – PS-2 Results from validation of hydraulic model 2022-11-07

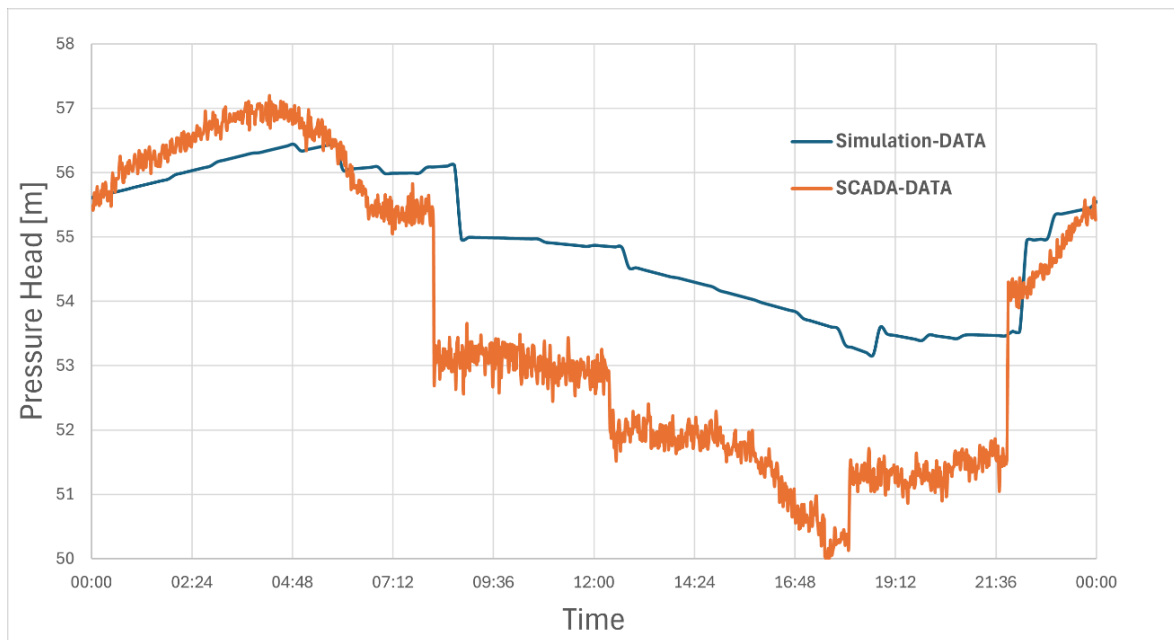


Figure D-1 Outgoing flow from PS-2 measured in the simulation and from real-operation data.

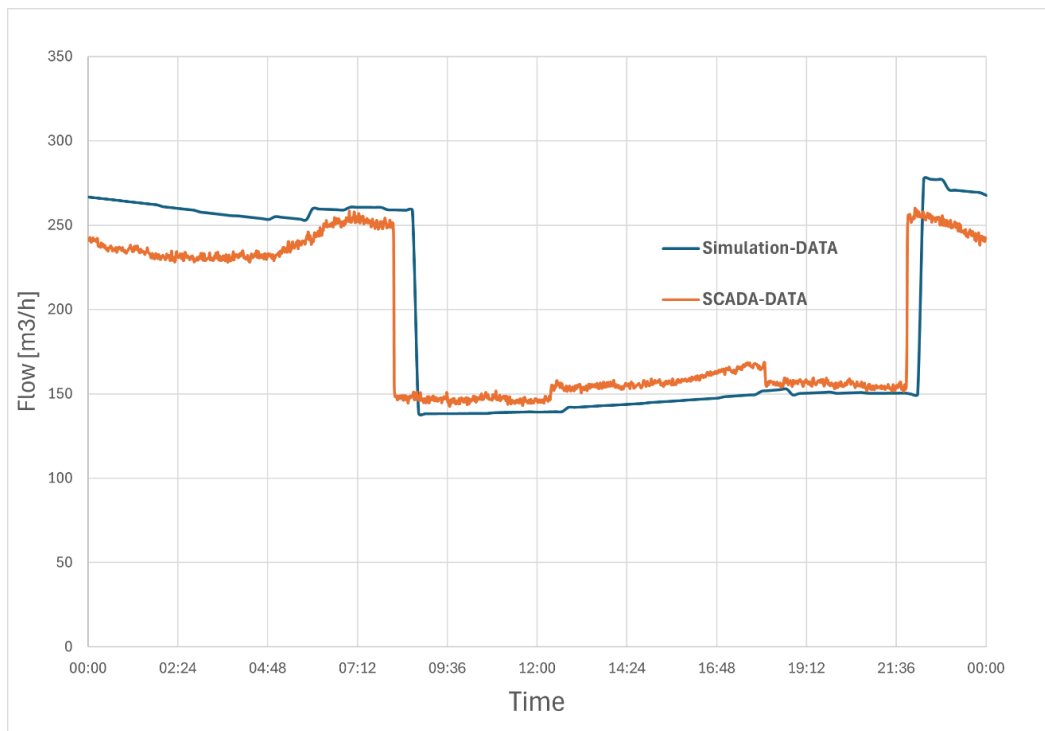


Figure D-2 Outgoing flow from PS-2 measured in the simulation and from real-operation data.

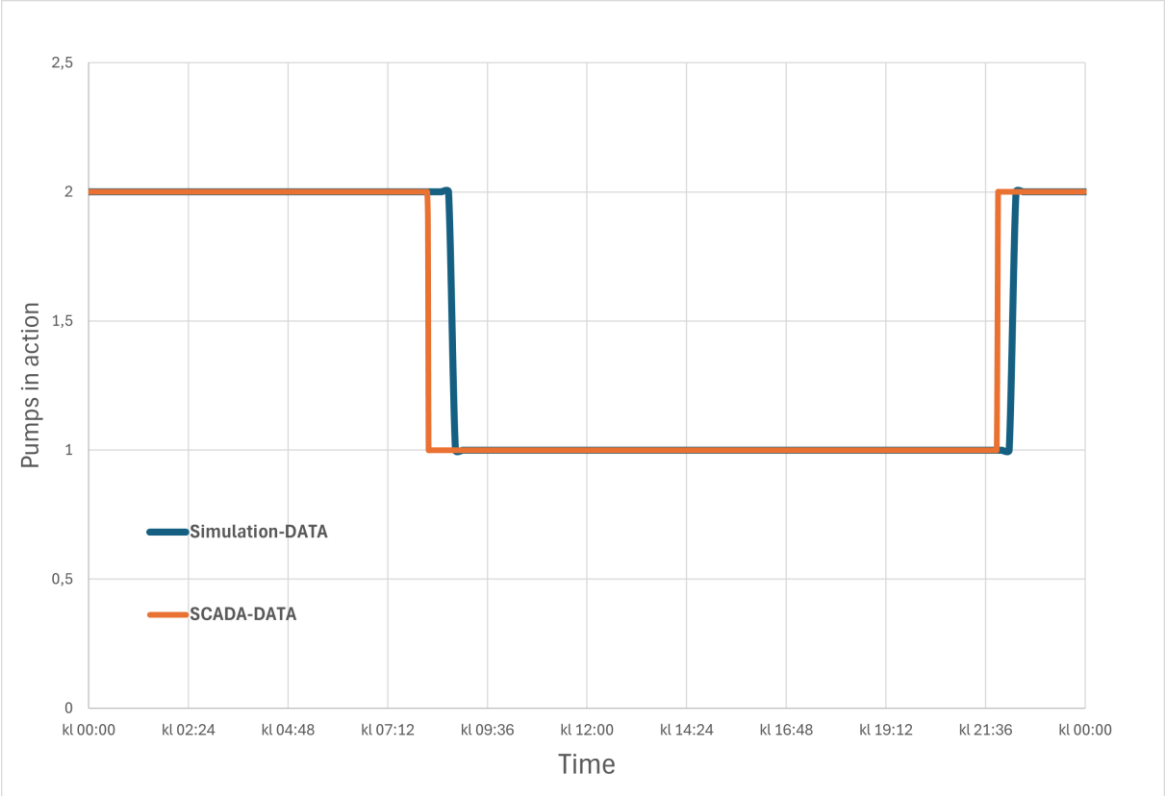


Figure D-0-3 Active pumps in the simulation and during the real-world scenario

E – Truncated Fourier series encoding

Given the specified problem approach, optimizing the 24 value setpoint curve directly will be challenging. The reservoir level at any given hour is dependent on its preceding value and thus also the entire history of the set-point curve. Traditional GA operators are thus inherently ill-suited due to the temporal disruption caused by the different crossover and mutations operations. Instead, a trigonometric function-based encoding can be utilized, expressing the reservoir level as a truncated Fourier Series.

A Fourier series expresses a periodic function as a sum of sines and cosines and is an example of a trigonometric series. Fourier series has been used in a wide array of scientific fields such as signal processing and physics to simplify complex, repeating signals, into more comprehensible harmonic components. In its general form a Fourier series has an infinite number of harmonics, which can be seen in equation E.1. For the specific application using an infinite number of harmonics would be infeasible and unnecessary.

$$f(t) = A_0 + \sum_{n=1}^{\infty} \left[A_n \cos\left(\frac{2\pi n t}{P}\right) + B_n \sin\left(\frac{2\pi n t}{P}\right) \right] \quad (E.1)$$

A truncated Fourier series limits the harmonics to a finite number of harmonics; the general formula can be seen in equation E.2. The choice of K is on a trade-off between complexity and stability. A higher K value results in more parameters along with a more complex search field. A smaller K-value generates a smoother curve but at risk of missing details of the signal. The harmonics need to be able to capture all expected trends in the reservoir level, without introducing unnecessary noise and complexity. It is important to note that this type of control scheme is not designed to optimize for a single, specific demand pattern. Instead, it should perform well across a range of demand scenarios and behaviors. As a result, the shape of the set-point curve is typically smooth and generalized, rather than highly reactive or tailored to one case. Based on current heuristic-developed set-point curves harmonics of the third degree (K = 3) were found to be a good tradeoff.

$$f_K(t) = A_0 + \sum_{n=1}^K \left[A_n \cos\left(\frac{2\pi n t}{P}\right) + B_n \sin\left(\frac{2\pi n t}{P}\right) \right] \quad (E.2)$$

In standard Fourier series each harmonic term two coefficient, which can be seen in equation E.3. Here A_n controls the cosine component, and B_n controls the sine component. Together they define both amplitude and phase of n^{th} harmonic.

$$A_n \cos\left(\frac{2\pi n t}{P}\right) + B_n \sin\left(\frac{2\pi n t}{P}\right) \quad (E.3)$$

These coefficients can be hard to interpret and in the context of GAs, independent mutations could be hard to predict result in undesirable solutions. A solution to this problem is to reparametrize these variables into an amplitude (ρ_n) and phase (φ_n) variables, as can be seen in equations E.4 and E.5.

$$\rho_n = \sqrt{A_n^2 + B_n^2} \quad (E.4)$$

$$\varphi_n = \arctan 2 (B_n, A_n) \quad (E.5)$$

Recoding the coefficients in this manner makes things easier to interpret since each parameter now represents a physical attribute of the curve. The final formula can be seen in equation E.6.

$$s(t) = A_0 + \sum_{n=1}^K \rho_n \cos\left(\frac{2\pi n}{P}t - \varphi_n\right) \quad (E.6)$$

Where:

A_0 = Baseline level

ρ_n = Amplitude of the n^{th} harmonic

φ_n = Phase shift of the n^{th} harmonic

K = The number of harmonics (e.g., 3)

P = The time period (e.g., 24)

DEPARTMENT OF DEPARTMENT OF ARCHITECTURE AND CIVIL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY