The use of Genetic Algorithms in Multi-Objective Optimization of Pump-as-Turbine Energy Recovery Systems

Tiago Monteiro Gomes Morais Baptista

Thesis to obtain the Master of Science Degree in

Civil Engineering

Supervisor: Prof. Helena Margarida Machado da Silva Ramos

Examination Committee

Chairperson: Prof. Vitor Faria e Sousa

Supervisor: Prof. Helena Margarida Machado da Silva Ramos

Member of the committee: Prof. Modesto Pérez Sánchez

June 2021

Declaration

I declare that the work presented in the document is of my own and therefore complies with the Code of conduct and best practices of the University of Lisbon.

Abstract

The water sector is in the process of suffering from the stress induced by the climate change that is occurring worldwide. An otherwise constant supply of water is now changing its patterns, in terms of frequency and intensity, with an increase in the drought season tendency, meaning that is imperative to deal more carefully with the existing water resources.

In the same line of thought, to reduce the unnecessary losses of water it is required to control them in the water supply networks worldwide. The current scenario has 35% predicted loss of water on average, from a global point of view. To control these losses, the common approach is to reduce the pressure present in the network since excess pressure induces more water losses in the system. Pressure Reducing Valves (PRVs) are commonly used to maintain the acceptable pressure levels in networks. Although these systems are highly effective in managing pressures, it is counter intuitive that the energy dissipated by the PRVs is not recovered.

The use of Pump-as-Turbines (PATs) as an energy recovery system to control the pressure in the supply network can be a feasible solution. Installing a system based on PATs can be an extremely complex problem, since there are multiple variables that influence the system behaviour. This study presents a new methodology and analysis that uses the concept of Genetic Algorithms to do a multi-objective optimization of a system with multiple PATs, selecting the correct PAT model and every definition operating rules that will influence the system's effectiveness.

Key- Words: pump-as-turbines (PATs), genetic algorithm (GA), multi-optimization, water losses, Micro hydro production.

Resumo

O sector da água encontra-se progressivamente sob a ameaça das alterações climáticas no panorama mundial. Os recursos de água, constantes e previsíveis, estão cada vez mais vulneráveis a efeitos de mudança de padrões e intensidade de precipitação. Simultaneamente, existe um aumento dos períodos de seca, tornando imperativo que exista uma melhor gestão dos recursos existentes.

Na mesma linha de pensamento, para reduzir os desperdícios de água é necessário controlar as perdas de água, que existem nas redes de abastecimento de água. No cenário mundial actual, existem perdas de água médias nos sistemas de abastecimento de água da ordem dos 35%. Para controlar estas perdas, a estratégia comum passa pela redução da pressão existente nos sistemas de abastecimento, dado que, existe uma correlação entre a pressão na rede e as perdas. As válvulas redutoras de pressão (PRV) são tipicamente usadas nas redes de abastecimento para manter os níveis de pressão recomendados. Apesar destes dispositivos serem muito eficientes no controlo das pressões, é contra intuitivo que a energia dissipada por estas válvulas não seja aproveitada para a geração de energia elétrica.

O uso de bombas a funcionar como turbinas (Pumps-as-turbines, PAT), como sistema de aproveitamento de energia para controlar as pressões presentes numa rede de abastecimento de água, podem ser uma solução possível. Dimensionar um sistema baseado em PAT pode ser extremamente complexo, dado que existem múltiplas variáveis e relações que influenciam a solução final. Este estudo apresenta uma metodologia nova e análise, que utiliza o conceito de Algoritmos Genéticos (GA) para executar uma otimização multiobjectivo de um sistema com múltiplas PATs. Entre as variáveis de análise está a localização da PAT, a seleção do modelo e todas as definições que influenciam o funcionamento do sistema.

Palavras chave: bombas-como-turbinas (PAT), Algoritmos Genéticos (GA), otimização multiobjectivo, perdas de água, micro hídricas.

Index

List of Figures

List of Tables

ABBREVIATIONS AND SYMBOLS

2D – Two dimensions

3D - Three dimensions

ALCC – Annual Life Cycle Cost

BEP – Best Efficiency Point

CFD – Computer Fluid Dynamics

DMA – District Metered Area

EPANET - Environmental Protection Agency Network

ER – Electrical Regulation

GA – Genetic Algorithm

H – Hydraulic Head

HER - Hydraulic Electrical Regulation

HR – Hydraulic Regulation

mw.c. – Meter Water Column

MOGA - Multiple Objective Genetic Algorithm

n – Rotational Speed

NSGA – Non-dominated Sorting Genetic Algorithm

NWR – Non-Revenue Water

P - Power

PAT – Pump-as-Turbine

PRV – Pressure Reducing Valve

Q – Flow Rate

RPM – Rotation Per Minute

SSP - Single-Serial Parallel regulation

SWRO - Seawater Reverse Osmosis Processes

VOS – Variable Operating Strategy

WDN – Water Distribution Network

1 Introduction

1.1 Scope

The present master thesis focuses on the study of pressure management of water supply networks and the consequent exploitation of electric energy generated with the use of Pumps-as-turbines (PATs) installed in the system. More specifically, the investigation of the model development and optimization analyses result from a multi-variable and multi-objective optimization approach, with the use of genetic algorithms (GA), to the inherent complex problem. The objective of this optimization is to ensure a better use, and effectiveness, of the interventions and corresponding resources used for the goal of regulating pressures in the water supply network. Simultaneously, and in a more elementary way, the goal is to contribute to the effectiveness of the water utility system, both directly related either the conversion of energy that is already available and contained in a pressure system, as the indirect cost that the reductions of pressure and the consequent decrease in water losses create (Clarke, 2010). Worldwide water losses, just in the supply system, as non-revenue water (NRW), is estimated to be on average 35%. In developing countries and regions more prone to excess pressure the values can reach 50-60% (Fields, 2015).

Climate change is a great challenge for the worldwide society to deal, it is already a centre subject in world politics, not just as a talking point, as a main topic that is actively being tackled by the society. A major consequence of climate change is the drastic changes in weather patterns, and global temperatures that could lead to an alarming crisis of water scarcity worldwide. In this fashion it is critical to preserve the already available water resources. The current model of managing water resources in Europe is already putting extreme pressure in the natural sources of water and have multiple ecological impacts since damming rivers and transferring water from different river basins affect the normal interactions and natural way of functioning of the ecosystem. Multiple regions, especially in Mediterranean countries are already suffering from water scarcity, some even already use desalination processes to generate freshwater, like Spain, Cyprus and Malta, constantly enhancing the production from these methods due to the increasing water needs and decreasing in natural water availability. The artificial production of water comes with a high energy price, in the range of 2-2.5 kWh/m^3 for the best Seawater Reverse Osmosis Processes (SWRO). It has a special impact as a cost since it's a natural procedure that could be avoided with correct management and this ends up harming indirectly, by inevitably using fossil fuels as energy source, causing gas emissions that contribute to the climate change (Craig R & Andes, 2015). The pressure on the different type of water resources changes from region to region, and seasonally, not only due to the natural cycle of water and the natural abundance or scarcity due to the variability within seasons, but also because the type of water necessities are also different in every region (Agency, 2019). Multiple climate models predict an increase in intensity and absolute yearly quantities of precipitation in Northern Europe due to climate change and the opposite in Southern Europe, with less precipitation and drier summers. (e.g. Southern Europe stresses the water supply both in rivers and ground water in spring and summer for agriculture purposes and Northern Europe stresses the surface water systems in the winter for cooling in energy generation and heating). The control of water losses in supply networks is a detrimental priority to fight the water scarcity problem in Europe.

The use of Pressure Reducing Valves (PRVs) to maintain the correct level of pressure in the supply network has been the standard technique used to mitigate the adverse problems caused by the excess of pressure. The installation of a PRV has the goal of ensuring the minimum pressures, customer satisfaction and the control of excess pressure that only contributes for more water losses in the form of small leaks, creates an increase in the probability of critical burst in the system that may impair the water supply. The control of pressures creates the indirect saving of reducing repair interventions and improving the life of the supply systems. The approach of using PATs working to achieve the same effects as a PRV has been in use for several years. The use of pumps to produce energy has the benefit of recovering the energy dissipated and transform it in electricity. Multiple steps were made, by autors such as Stepanoff (1957) , Alatorre-Frenk (1994) and Wiliams (1994) ,in the development of this technology have been taken, from the hydraulic analysis to predict the behaviour of a pump working in reverse mode, to the application of PATs in actual water supply networks and the study of economical and hydraulic feasibility of this type of solution, Derakhshan & Nourbakhsh (2007), Gonçalves & Ramos (2008) and Chacón et al. (2019). The optimization of the position and characteristics of the PAT performance is an extremely complex problem to solve. In this area, several works starting with simple PRVs application and reaching simple PAT installations have been produced. The use of GAs has been progressively used to study the location of the implementation, and pressure to be removed with a PRV. An extension of this concept has been applied progressively in terms of complexity to a PAT solution.

1.2 Objectives

A smart management and application of resources is critical in the modern economy. The same should be applied to the implementation of energy recovery systems in the water supply networks. The use of GAs could be the answer to the multi-variable problem of implementing the best solutions in the system to make it the more efficient as possible. Analysing each water network and recovery system individually is very straight forward and with multiple studies in the subject. A smart application of the system using robust algorithms, such as GAs that allow for a strong adaptability to different scenarios and necessities, could allow for viable and better solutions.

The purpose of this work is to study the effects of the application of a GA tool model in an integrated approach to the implementation of an energy recovery system, with the use of PATs. The goal is to apply all the variables, that influence the near future, in one compact “genetic” form. The variables used are power curves and H-Q characteristic curves for multiple turbine rotational speeds, implicating the use of electric regulation of the system conditions, at different demand patterns throughout the day. Although this work is focused on the application of a system in the short term, it opens the way for a long-term approach that could include as variable the progression of the demand pattern throughout the live cycle of the system. The evolutionary capacities of the optimization will be analysed, the effects of the general convergence of the pareto front and the adaptation of the final solutions in terms of defined characteristics of a PAT regarding to the available rotational speeds.

The work of developing the optimization based on the NSGA-II genetic algorithm (Deb et al., 2002) was done using MATLAB programming language. It is intended to evaluate the use of EPANET-MATLAB Toolkit. This toolkit creates an interface between both software's allowing for an easier data analysis from the hydraulic simulations done in the EPANET model. An EPANET model for the hydraulic simulation must be created first, before the optimization. This model consists in a simple water system developed for the purpose of this work.

To achieve the goals of the proposed research, fitness functions for each objective need to be established, in this case, the energy generation, the pressure regulation and the rentability of the system. A PAT library needs to be created with every characteristic that could influence the behaviour of each model at different working speeds. The demand pattern needs to be defined and imported to the water system. Likewise, characteristic curves have to be included in the network.

1.3 Thesis Structure

This document is comprised of 6 different main chapters: 1) Introduction (detailed previously); 2) Pressure regulation in water networks; 3) Genetic algorithms; 4) Methodology; 5) Case study; 6) Conclusion.

The pressure regulation methods and a presentation of the general behaviour of the solutions that are generally applied, such as pressure reducing valves are included in **Chapter 2**. An introduction to PAT technology is also included, with different control methods and the evolution of studies regarding the conversion from pump characteristics to turbine characteristics, from the analytical methods to the more recent improvements in Computer Fluid Dynamics (CFDs). The introduction to the turbomachine affinity laws, essential in the characterisation of the behaviour of turbines in different conditions is also detailed.

In **Chapter 3** the evolution of Genetic Algorithms is presented and the correlation to the solving of hydraulic related problems. A general introduction to different components that take part in the cycle of a GA and their most common variances are explained. The context and importance of the Pareto Front concept is related to the corresponding application in the hydraulic optimization problems.

Chapter 4 presents the methodology of the optimization used in this research. In the chapter, are also presented the relevant tools, software and element used. For each fitness function is detailed the process and the source of the results. The constraints and limits of the solution space are also defined.

The case study is detailed in **Chapter 5**. The characteristics of the hydraulic network on which the optimization process is run with are also presented. The same is done for the hydraulic analysis characteristics and times. The process and justification for the daily demand pattern used is also detailed in this chapter. The actual PAT library characteristic curves are defined, and the complete methodology used for the integral optimization is presented.

The last, **Chapter 6,** concludes the results of the approach developed and suggest future improvements to this research.

2 Pressure regulation in Water Supply Networks

2.1 PRVs for pressure and leakage control

There is a direct correlation between excessive pressure and water losses due to leakage in a water network. Therefore, a good pressure management is essential to regulate water losses (Clarke, 2010). Due to the later reduced corrective interventions expenses, such as fixing pipe bursts in the network, the better customer service by the water supply companies, and the savings in energy used in pumping and water treatment make this type of water losses management one of the most economical ones (Girard & Stewart, 2007). Pressure reducing valves (PRVs) are generally used to control the pressure by separating the system in district meter areas (DMA) that maintain a certain pressure range inside. The separation into DMAs allows for a faster detection of water leakages, better monitoring, and enhanced precision of the recovered data by reducing the complexity of the network to analyse. Llicic & Kovac (2009) concluded, after a 4-year project studying a DMA in Zagreb, Croatia, and the same DMA with pressure regulation that there was a reduction of 40% in major pipe bursts within the DMA.

PRVs control downstream pressure to a predetermined definition, this way the downstream pressure remains stable independently of the conditions, and its inherent fluctuations, upstream of the valve. To maintain a variable headloss, that is dependent on the system downstream pressure, a lock is actuated varying its level of actuation according to the necessary local head loss needed. There are multiple types of PRVs, spring, piston, and diaphragm (Ramos et al., 2005).

There are four main different types of operation condition for the valves:

- Constant downstream pressure.
- Constant head loss.
- Constant downstream pressure but with variability in time (mostly used with 2 times for high demand and low demand hours).
- Constant downstream pressure directly variable according to demand.

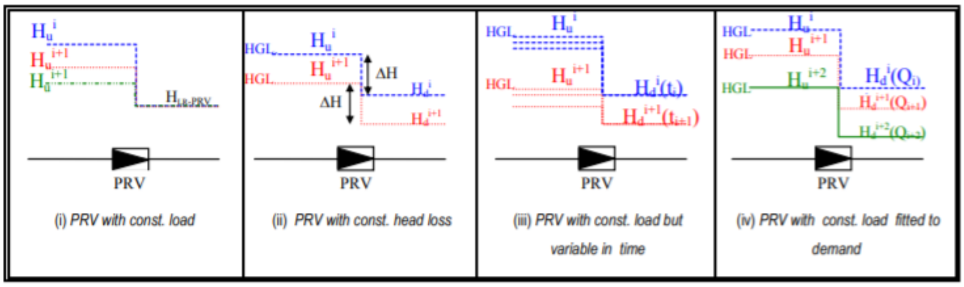

Figure 1 - Operation conditions of PRVs (Araujo et.al 2005).

In each different type of operating condition, the valve can be in different states: **Active estate -** Means the downstream pressure is higher than the reference and the valve is activated to induce a local head loss; **Passive state -**The pressure upstream is equal to the pressure downstream of the valve, meaning that in this case the valve is completely open; **Closed state -** The pressure downstream of the valve is higher than the pressure upstream, the valve is completely closed, working as a one-way valve.

Extensive research has been done in the incorporation of GAs within the optimization process of selecting the position and definitions of PRVs applied to a water system. Initial studies started with the use of simple GAs to predict the optimal locations of valves in the water systems, with the objective of minimising water leakage (Reis, Porto, & Chaudhry, 1997).

Later, Araujo .et al (2005) further improved the optimization of the valve locations. In this case, the work was comprised of two GAs based optimizations, the first improved the position of the valves by changing the roughness of each pipe as variable, this way a local headloss is simulated. In the locations chosen for PRV implementation, which were the pipes with higher induced headloss, a second optimization takes place adjusting the actual headloss to be applied by the valves.

2.2 Pump as turbine (PAT) for energy recovery and pressure management

PRVs induce a local dissipation of energy in the water supply network. This dissipated energy could be harvested with the aid of hydropower devices that allow for energy production while at the same time the pressure requirements are achieved. Contrary to traditional hydropower generation in rivers or other natural water sources, the flow characteristics in a water supply system are hard to predict and are very unstable, affected by multiple seasonality's in a daily, monthly, and yearly basis. To add more barriers to an already complex problem is essential to maintain the original purpose of the system of offering a reliable water delivery, of maintaining minimum and maximum pressures, of maintaining water quality and keeping with the required demand all the time (Fontana et al., 2012).

Typically, small hydropower generation relied in small scale variants of the traditional turbomachines, such as Pelton, Francis, and Kaplan. These alternatives require big initial investments and despite offering high efficiencies they require long payback periods to have economic viability. For this reason, a route towards simpler turbines was taken and, in the same way, the consideration of a pump working as turbine was seen as a simple, but robust, alternative. PATs are extremely simple machines that are readily available worldwide, that can offer low maintenance costs and a fast payback period of around two years or even less (Derakhshan & Nourbakhsh, 2007). Gonçalves & Ramos (2008), on an implementation proposal for a PAT pressure management and energy recovery system in Aveiro, Portugal, have achieved pay-back periods of 7 and 8 years. Perdigão (2018) made a similar assessment of the most adequate PAT system to replace PRVs in the water distribution network (WDN) of Funchal, Portugal and concluded that there was a high variability of pay-back periods depending on the different methods of energy output connections. In this case, grid, local and battery connection were analysed and in the same order payback periods of roughly 5, 3 and more than 16 years were achieved. By comparing the reduced efficiencies of PATs with their traditional counterparts, and taking also into consideration the reduced investment, it is possible to assume that this system can be an important alternative for energy generation (Chacón et al., 2019).

Motwani et al. (2013) proceeded with a cost analysis of a PAT implementation in a rural area. The authors compared a Francis turbine implementation with the PAT. They were able to make the conclusion, that in the reality of the study the PAT solution presented a smaller annual life cycle cost (ALCC), achieving, respectively for Francis and PAT, a cost per unit of energy produced of 6.08 and 5.07.

Puleo et al. (2014) studied the application of PAT in WDNs, the conclusion from the investigation is that PAT application inside the DMAs can be inefficient and unable to comply with the objective of recovering energy. The authors suggest that is more advisable to apply PATs directly downstream of the DMA entrance and in the main supply connections. The instability of local demands could compromise the system. In the other hand, Fontana et al. (2012) studied the application of a PAT pressure control system in the DMA and concluded that the solution showed "that a relatively large energy recovery could be coupled to a significant reduction in water loss".

An obstacle to correctly implement pumps working in reverse mode, is the lack of data of each pump working in those conditions. The curves are generally not provided by the manufactures. This created a demand in the investigation of different methods to determine the curves of the pumps working in reverse from the original pump curves (Derakhshan & Nourbakhsh, 2007). In the late century, many authors worked to achieve good predictions in turbine behaviour, Williams (1994) illustrated how a lack of precision in this procedure can have major consequences in the viability of the PAT implementation. The same author compared multiple analytical prediction methods developed, defined ranges of error acceptance, and applied the methods on 35 pumps. The methods tested included, in example, Stepanoff (1957) and Alatorre-Frenk (1994). The variability of results was clear, being the Sharma's (1985) method the one with the smallest error. The method got, nevertheless, 20% of the pumps analysed with results out of the desired range. The multiple methods referred in the study used the pump efficiency to achieve the turbine H-Q characteristic curves and, in the case of Sharma's method, the relation between specific speeds developed by Engel to define the relations between characteristics in pump and turbine modes (Williams, 1994).

Derakhshan & Nourbakhsh (2007) developed a prediction method to determine the Best Efficiency Point (BEP) for low specific speed centrifugal pumps. Simultaneously, concluded that centrifugal pumps can appropriately run in multiple rotational speeds, heads, and flow rates and that the efficiencies are very similar in pump and turbine modes. For the same specific speed, a pump with higher efficiency works at higher H and Q than the less efficient counterpart.

Singh & Nestmann (2009) further investigated prediction methods for PAT performance. Created a model to optimize the prediction of performance characteristics based on multiple experiments of other pumps working in reversed mode and the classical principles of turbomachines. The model presents an exceptional precision in most characteristic curves. A further improvement in the data base of the model would improve the precision of the predictions.

2.3 Computational fluid dynamics (CFD)

With the same goal of improving the prediction methods of the characteristics that define the behaviour of pumps working in reverse mode, many studies, and progress, has been made by using computers to simulate the behaviour of turbomachines in a virtual system. The system is defined and solved by mathematical systems that replicate the boundaries and fluid behaviour. Models of the machines are constructed by a 3-D grid software as the one represented in Figure 2. The model and the fluid are represented by the 3-D mesh, the higher the quality of the mesh and higher the number of cells that make the grid, the higher are the quality of results. A constant balance between grid density and efficiency in the simulation processing must be achieved (Frosina, Buono, & Senatore, 2017). A performance improvement in the approaches to predict turbomachine behaviour with CFD could allow for a fast and realistic simulation of different turbomachines operating in different environments with a very high precision.

Taking into consideration the variability in the quality of simulations, Carravetta et al. (2012) concluded that CFD predictions are an “valid alternative to experiments” that aim to determine the pump reverse curves in the absence of a characterization in reverse working condition by the manufacture. Nautiyal et al. (2010) also used a CFD software to achieve the characteristics of a PAT, the conclusion of the study was that this type of technique can be a very effective tool. The authors report small errors in the prediction and suspect that an improvement in the grid model definition can solve the errors. Nevertheless, the authors consider that “more experience in computational analysis will also help to obtain accurate convergence of CFD”. Frosina et al. (2017) validated a CFD model of three pumps with the normal operating mode and then simulated the same computer models in reverse conditions of flow. The results were compared with the previous analytical methods used to predict reverse behaviour. It was found that the CFD results are in accord to the more precise analytical methods, like Sharma’s and Stepanoff’s. CFD is a vital part in the conception and development of new solutions, offering better solutions adapted to more diverse scenarios (Simão, 2009).

On the other hand, Derakhshan & Nourbakhsh (2008) concluded after a comparison between experimental results of a PAT characteristics and the ones obtained in a computational model that the results are already viable in pumping mode, but in turbine mode there is still a considerable difference in results, the authors point out that in turbine mode a higher sensibility to the model grid complexity must exist. Fontanella et al. (2020) worked in an analytical method that compiled many existing PAT curves to calibrate the analytical model by considering that the errors associated to the early methods were caused by the small database from which they were extrapolated. The authors consider that CFD methods create additional difficulties, due to the validity of mathematical models and the approach used.

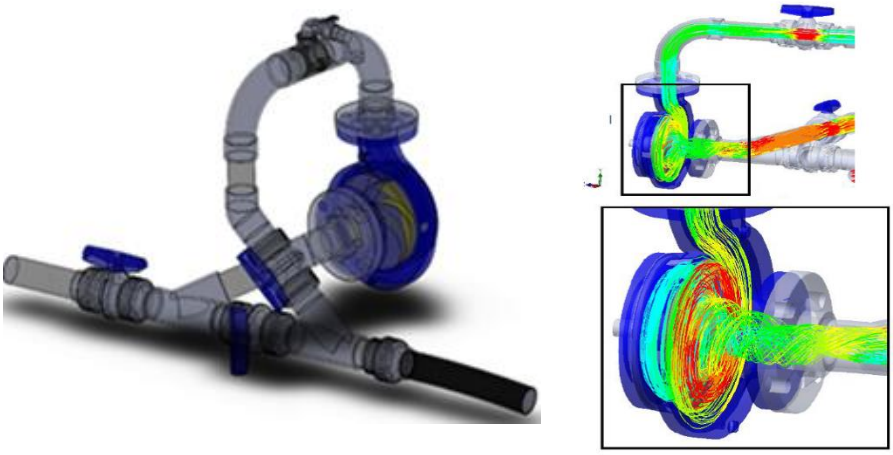

Figure 2 - Example of 3-D mesh used in CFD simulations and the pressure diagram.

2.4 Variable Operating Strategy (VOS)

Traditional hydropower installations have very strict operating conditions, the fluctuations are small, there is no daily variance of great impact in the conditions, being possible to exist seasonal changes that in some extent are also easily predictable. When applying energy recovery systems in WDNs, there are a multitude of variable operating conditions, in this case, with hourly, daily, and yearly variance that influence the system project and viability. If a PAT was to be installed in a water supply system for a steady state and constant conditions, like in major hydropower facilities, it would likely have during its lifecycle various moments where it would be off the minimum desirable operating conditions for which it was design.

Carravetta et al. (2012) proposed a procedure, named variable operating strategy (VOS), for the election of the most appropriate system. The authors based this method in a preliminary use of the electric generation plant overall efficiency. This efficiency takes into consideration the energy recovered by a PAT for each hydraulic analysis time interval and respective operation point of the turbomachine and correspondent hydraulic conditions offered by the network.

The proposed strategy (VOS) starts with an investigation of the available conditions, respective available head and flow rates and the type of turbomachine selection. For a wide set of characteristic curves, the plant overall efficiency is calculated. Based on the results of efficiency, the best set of curves is selected, and a near-optimal real curve is selected from actual turbines in the market. The same procedure is repeated with the real curve to have a true, and final, evaluation of the operating performance. As previously said, the authors consider that this procedure with the use of CFD curves provides a great alternative to experimental curves since the method focus initially on a selection phase and the results provide accurate solutions.

Carravetta et al. (2012) proposed a method of regulation to achieve better efficiency from the overall system, the Hydraulic Regulation (HR). HR is proposed to have the PAT running at optimal conditions without being dependent on the natural PAT behaviour and to allow for an adequate water supply. (Figure 3). HR works by regulating the flow that enters the PAT with the use of a bypass regulating valve. The flow and head conditions are adapted to the PAT. If a discharge reduction is needed to achieve the ideal PAT working characteristics the valve opens, creating a reduction in the flow that goes throw the main pipe and actual PAT and instead runs through the above-mentioned Bypass. A similar procedure is done to achieve the ideal head for the optimal PAT performance. A PRV installed in series upstream of the PAT induces the necessary head drop to achieve the desired conditions.

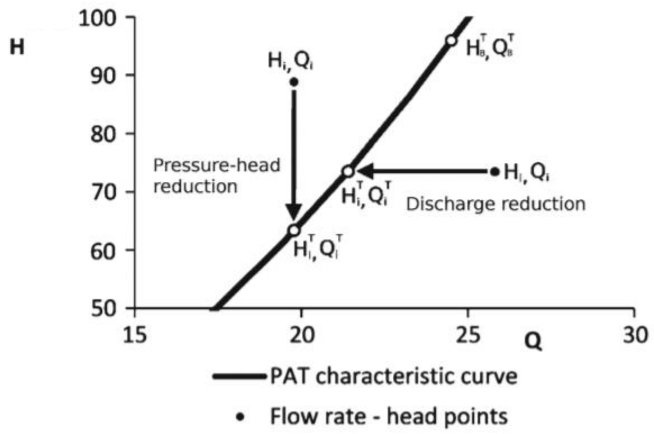

Figure 3 - Hydraulic Regulation, effects of the regulation system (Carravetta et al., 2012)

Carravetta et al. (2013) extended their studies on regulating PAT operations in variable conditions by exploring Electrical Regulated (ER) systems and comparing them with HR systems. In this method the PAT working conditions are adapted to the flow and head that exists in the system to optimize the energy production without any other mechanical controller. The behaviour of the turbomachine depends on the rotational speed it is working on, by changing the frequency on which the generator in producing electricity, due to the way it produces electricity by the rotating magnetic field induced inside by the magnetic poles, it is possible to change the turbomachine speed. When the frequency changes, with the use of an inverter between the generator and the corresponding output connection, the rotation of the PAT changes. The different rotational speed produces modifications in the characteristic curve of the equipment. Higher frequency produces higher rotational speed. The opposite conclusion can be made in opposite conditions. When the rotational speed is higher, the head drop created by the PAT is also higher, with smaller flows. When the rotational speed is lower, a higher flow goes through the system with corresponding smaller head drop induced in the system. A generic example of such variation is rotational speeds and their respective effects in the characteristic are shown in Figure 4.

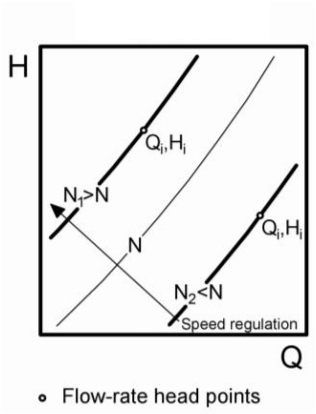

Figure 4 - ER, and the corresponding characteristic curves changes (Carravetta et al., 2013)

In this regulation method, information regarding PAT performance at different rotational speed and its efficiency curves are essential. It is possible to obtain the different characteristic curve by the three different methods explained above, by CFD, by experimental results and from one-dimension prediction approaches. Different rotational speed curves and their respective behaviour regarding efficiency also need to be defined, for this it is possible to use the turbomachine affinity laws to provide different curves for different rotational speeds. The process used to define the curves is described in the correspondent chapter.

An intermediate approach to both previous regulation schemes is the Hydraulic Electrical Regulation (HER). This method uses both regulation systems interconnected to gain even further adaptability. It has the option of using the mechanical valves for regulation and the electrical regulation using the inverter to achieve optimum performance in multiple scenarios.

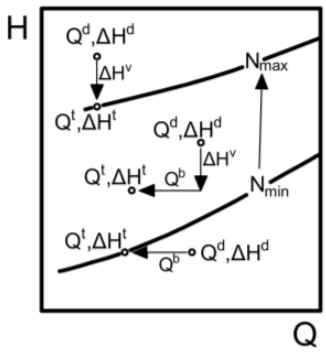

Figure 5 – HER and its mutual adaptation of flow and PAT characteristics (Carravetta et al., 2018).

It is the most effective system, but only suitable for high power installations that produce enough energy to support the double investment, in the inverter and valves, and yet have the ability to be profitable. Like previously demonstrated for the other methods, Figure 5 shows simultaneous adaptation of flow characteristics and PAT characteristics.

The same authors explored a different regulating method, the single-serial-parallel regulation (SSP). To eliminate the need of expensive additional equipment's, such as PRVs, inverters and control boxes, a simplified system that offers three different operation modes was proposed (Carravetta et al., 2018) The installation diagram and corresponding working conditions of the system are represented in Figure 6.

This system relies on two PATs and three control valves installed as described in Figure 6. As previously referred, this method offers three different operating modes (Carravetta et al., 2018):

- Valve I on, PAT A **ON**, valve II and valve III **OFF**, PAT B **OFF** – single PAT.
- Valve II and valve III **ON**, PAT A and PAT B **ON**, valve I **OFF** – series PATs.
- Valve I and valve III **ON**, PAT A and PAT B **ON**, valve II **OFF** – parallel PATs.

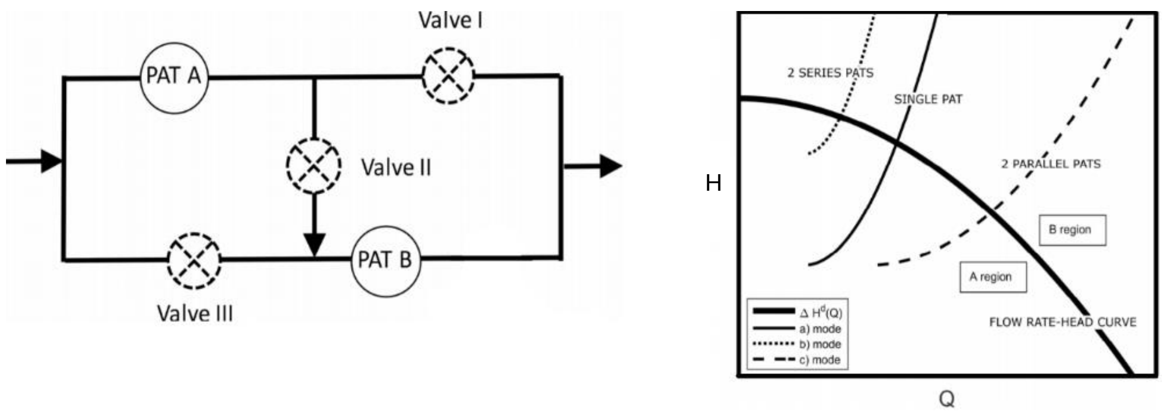

Figure 6 - (a) Diagram of installation scenario and (b) correspondent curves with the different working modes (Carravetta et al., 2018)

2.5 Turbomachine affinity laws

Turbomachine affinity laws are a critical element in the prediction of real scale working conditions from data obtained in a small-scale model. The turbine efficiency can be extrapolated from the specific speed of the flow going in and out (or any other equivalent points) of the turbine wheel. In the same fashion, it is possible to conclude that machines, identic in their geometry and specific velocities, have the same efficiency.

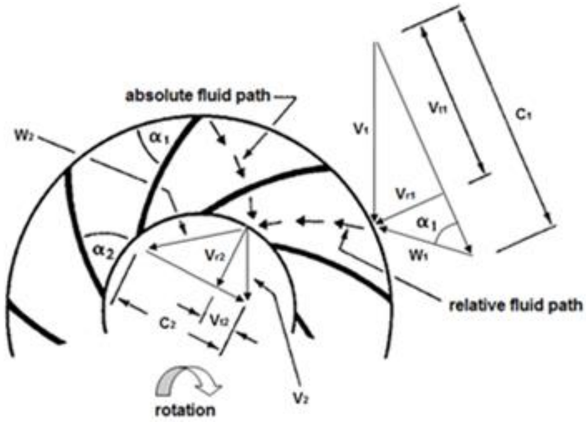

Figure 7 - Specific speeds in a turbine wheel.

The relation between peripheral velocity (C), size of the turbine and the respective rotation on which it is working, and the above relation between specific velocities allow for an also true relation between rotational speed, induced head drop and size of the turbomachine. If we consider this relation, it is possible to apply it to the same turbomachine running at different rotational speeds and have the capability to predict reliably the behaviour of, in this case, PATs at different speeds.

$$\frac{D}{D'}\,\frac{n}{n'} = \frac{C}{C'} = \left(\frac{H}{H'}\right)^{1/2} \qquad \textbf{(1)}$$

The decomposed relations obtained from the specific speed previously presented and directly used in this research to determine the correct performance characteristics for the multiple PATs are presented below:

$$\frac{n}{n'} = \left(\frac{H}{H'}\right)^{1/2} \qquad \textbf{(2)}$$

$$\frac{Q}{Q'} = \left(\frac{H}{H'}\right)^{1/2} \qquad \textbf{(3)}$$

$$\frac{p}{P'} = \left(\frac{H}{H'}\right)^{3/2} \qquad \textbf{(4)}$$

3 Genetic Algorithms

3.1 Concept and evolution

Genetic Algorithms (GA) are a heuristic search method developed by John Holland, colleagues and students at University of Michigan during the 60's (Whitley, 1994). GAs are based on the dynamic system that make the theory of evolution in the natural world. They consist on the survival of the fittest solution and its development to become even better adapted, in this case, with the possibility of surpassing the original fittest solution, becoming itself the fittest. (Goldberg, 1989). This procedure works on every new generation of solutions. Evolution is, indeed, a method of search for the best solution in an infinite time frame, and the fact that the current solution is the best now, does not imply that it is already the optimum. The optimum and it path is an uncertainty. Evolutionary algorithms are very well suited to problems that require a strong capability of adaptation to continue performing adequately in changing environments, well suited to problems that require innovative solutions that do not fit the "traditional progression" of ideas. In the same way, GAs can tackle problems where the size of the solution space is extremely big with a high complexity of variables that do not allow for a clear and reasonable approach "by-hand" with enumerative methods, neither by a deterministic approach. (Mitchell, 1995).

Goldberg's describes the optimization with GAs as a two-part optimization, the first is the improvement of the results, the second is the achievement of the true optimal point. Goldberg's objective with this separation is to empathise that GA is not, especially in scenarios with multiple solutions, a tool that aims directly to achieve the overall optimal point. The focus is the fast improvement of solutions.

In the adaptation of the evolution present in the natural world to a computational system, there are three main groups that stand out that define the solutions and its evolutions (Holland, 1992):

- Environment on which the system/solution is undergoing the processes of evolution.
- Adaptive plan that allows the system/solution to actively make changes for a better adaptation to a changing environment.
- A measure of the system/solution performance in the environment.

Although all the previous points are important, the main component, is the adaptive plan. Supposing a constant environment, it is possible to eliminate a level of complexity. Since the context of this research is the application of an engineering solution, and there should not exist further adaptation of the system at least in the physical properties that are fixed. The adaptive plan is faced with specific obstacles in its design, as suggested by Holland in *Adaptation in Natural and Artificial Systems:*

- The solutions to evaluate are in great number.
- In complex problems is difficult to determine that a certain variable is the cause for good or bad performance of the solution.
- The performance functions can be extremely complex.

- In complex problems there is a very big flow of information. From environment characteristics to performance evaluation.

To find a solution, Holland suggests a robust and simple adaptive plan. The use of the chromosome concept is suggested to be applied as the base structure of solutions with the use of operators of selection, mutation, and crossover as part of the adaptive system, and thereby becoming themselves a fundamental part of every GA. These operators use probabilistic rules to define their action in the chromosomes. Although they use probability, it is important to clarify that this probabilistic transition is not equivalent to a flip of coins. The use of probability consists in small random choices that guide the search into best regions of the solution. (Goldberg, 1989)

Each chromosome has the complete characteristics of its variables, and consequently, its complete solution. The elements that make the chromosomes in a population are often called "locus", in analogy to the real biology counterpart, and are comprised most often by bit strings (i.e., strings of 1s and 0s) (Mitchell, 1995). This offers a simple and easy to process "language" that allows for a strong robustness of the system and its adaptive operators. Multiple solutions/characteristics of a problem can be encoded by this simple bit-string method. (Mitchell, 1995)

A Genetic Algorithms is different from normal optimization methods in four ways (Goldberg, 1989):

- Only deals with the results and the coding of the parameters of the solution, not with the parameters themselves.
- Searches in simultaneous a population of points, not a single point approach.
- Is based only on the feedback of a payoff information, defined by objective functions. Does not rely on other auxiliary knowledge, information, or mathematical characteristics of the solutions.
- Relies on probabilistic transition rules, not deterministic.

By working with multiple points simultaneously, not a point-to-point approach, there is a multiple optimization of various points. This means that the probability of achieving a false peak in the solution is reduced adding to the robustness of the solution.

In the same manner, the lack of dependency on auxiliary information to determine the route to optimization makes the GA independent on how strong or relevant the additional information is. The lack of previous information and the blindness on which the algorithm runs, are a major benefit of this type of system. The blindness allows for a more robust system by not ignoring innovative solutions that, by definition, where not in the scope of predicted objectives and allows for a bypass on more errors associated with external information (Goldberg, 1989). GAs are completely driven by the feedback of the objective functions.

Relying only on a probabilistic transition rule to choose the next solution point to evaluate, means that the likelyhood of getting a false result, for example, a local maximum, is reduced. The probability used in this changes work in the same way as the lack of auxiliary information to enhance the robustness of the algorithm, decreasing the impact of unpredicted behaviour of the function/solution (Goldberg, 1989).

In line with this research, there were multiple studies and progress in the use of genetic algorithms in a multi-objective problem. In a real world situation, the optimization of multiple characteristics is necessary. The solution to this problem is usually approached by creating a fitness function that evaluates multiple criteria simultaneously to improve the solution. The decision maker, especially in multi-variable problems, does not know the correct relative importance of every objective (Fonseca & Fleming, 1993). The use of a fitness function that combines every objective to optimize without knowing the exact importance of each objective would be a step back on the robustness implied by the original concept of Genetic Algorithms. Setting aside the concept of weighed objective function to determine the optimum result, the idea of *Pareto-optimum* and *Non-dominated solution* is introduced by Chankong & Haimes (1983) and Schaffer (1985). Multiple *Pareto-optimum's* make the solutions in the pareto front. This front consists of every solution that cannot be considered better than any other solution in the front. Meaning that these solutions are not dominated by any other. The concept of *domination* is applied when a solution performs better in every objective in comparison with other solution. A *Nondominated* solution, as said before, takes part in the pareto front. This means it dominates all but one objective in comparison with other solution.

This method of approach to multiple objective problems was applied by Schaffer (1985) with GAs and originated the *Vector Evaluated Genetic Algorithms* (VEGA) that was later on noted, that due to the selection method, that consisted in the shuffling of the available vectors, the optimization tended to concentrate only in one area of the search space, promoting similar solutions. In the case of a concave pareto front, the solutions tended to concentrate in two locations (Richardson et al., 1989). The problem was predicted and named *speciation* previously by Schaffer (1985).

A new approach to selection needed to be created to avoid combining the objectives. A domination rank was created to define the pareto front in this new solution presented by Fonseca's and Fleming's called *Multiple Objective Genetic Algorithms* (MOGA). It explores the deficiencies of VEGA, applying a rank selection based on level of domination. (Fonseca & Fleming, 1993).

A similar approach, but with an "inverted" method of selection was developed. It defines the ranking of solutions looking from the other perspective, as suggested by Goldberg's, deciding the rank by the *nondomination* of the solutions. This system, called Nondominated Sorting Genetic Algorithm (NSGA), is the predecessor of the algorithm used in this research, NSGA-II, and was created by Srinivas & Deb (1994). The same authors developed the second version to solve problems that were present:

- High computational complexity of nondominated sorting.
- Lack of elitism.
- Need for specifying the sharing parameter.

On the initial version of the algorithm, the nondominated sorting was accomplished by comparing the multiple solutions and the multiple objectives. This means a total of (MN) comparisons to analyse one solution and (MN^2) to analyse every solution for the first pareto front, being M the number of objectives and N the size of population. In the worst case possible, where only one solution exists per pareto front, the total comparisons to analyse would be (MN^3). A new solution on NSGA-II called *Fast Nondominated Sorting* has a complexity of (MN^2) (Deb et al., 2002). Every solution is analysed only one time, and a count of domination is created for every solution. When the solutions on the first front, with domination count (0), are taken from the population, the remaining solutions can go down in one level of domination. The new pareto front is now nondominated.

In the same way, the first generation had a lack of elitism and studies showed that it was an important factor in the performance of evolutionary algorithms (Zitzler et al., 2000). In this manner elitism was inserted in the selection methodology. NSGA had the dependency on a "sharing parameter" to preserve diversity in the pareto-front. The main problem was the necessity to specify the parameter that goes against the desirable parameter-less mechanism, as said before, one of the strong points of GAs.

3.2 Methodology and components

As part of the adaptive system that characterises GAs, there are three main operators that, based on probabilistic results, create new population individuals. The general methodology and interaction between components of a GA is presented in Figure 8. There are multiple studies that explore in more detail the characteristics of these operators and the different approaches they should take. (Hassanat A. , et al., 2019) For the purpose of this research it will be introduced the general concept of each operator and their variances.

Crossover

Crossover operator has the purpose of combining the characteristics of two chromosomes (Parents) to create new ones (child/offspring) with mutual characteristics. It is a crucial component in the adaptation and the idealist objective is that the offspring chromosome is comprised of the best genes from both parents to create a better individual. Being GAs nondependent on previous knowledge and with this operator decision making governed by a probabilistic system, the perfect combination of the best genes is unlikely but as stated by Goldberg's the route of GAs is the gradual improvement (Beasley et al.,1993) The Crossover probability is user defined, and multiple studies have been done to determine the best probability and its effects on optimization. (Hassanat A. , et al., 2019)

Multiple methods of crossover exist, including more than two parent crossovers that are not relevant to this research since they are uncommon and not applied to NSGA-II (Kaya et al., 2010). A few types of crossovers are: *Single Point Crossover, Two Point Crossover, Multi-Point Crossover, Ring Crossover.* It is also possible to use heuristic, arithmetic, or intermediate methods. These are more suited to GAs that do not work in a binary format of genes.

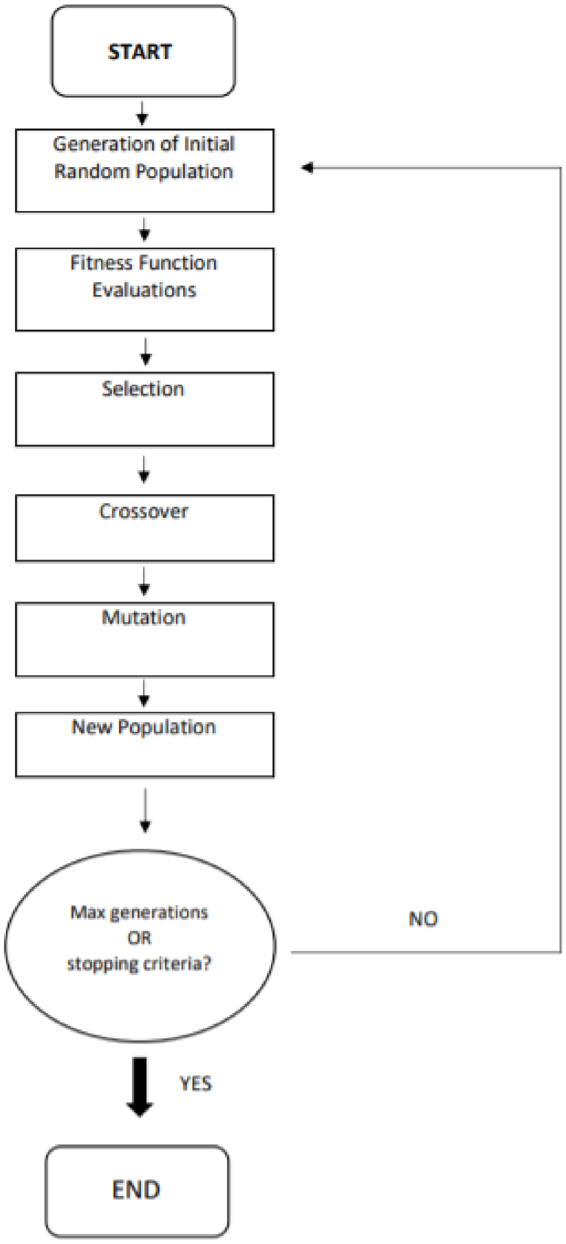

Figure 8 - Generic diagram of GAs methodology.

Single Point Crossover

A single point on both parents is randomly defined. The data that is present in the genes after this point are swapped between chromosomes creating two new separate solutions/offspring's (Kaya et al., 2010).

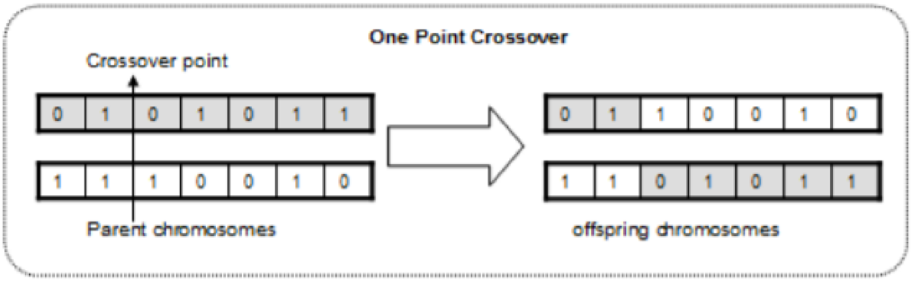

Figure 9 - Single Point Crossover Diagram (Kaya et al., 2010).

Two Point Crossover

In two point crossover, two points are randomly selected in the parents chromosomes. A swap of the genetic material is made with the genes that make the stretch of genes inside the points selected. This type of crossover is more efficient than the single point, reason why it was used in this research (Kaya et al., 2010).

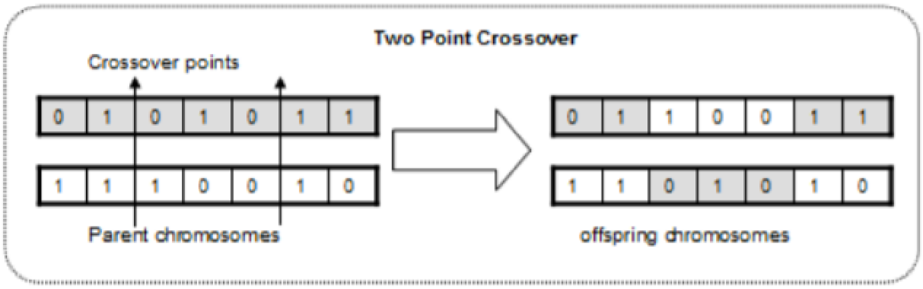

Figure 10 - Two Point Crossover Diagram (Kaya et al., 2010).

Ring Crossover

The ring crossover is a method similar to the one present above, has the advantage of offering a slightly better variability in offspring. The procedure begins with the union of the parent's chromosomes in a ring, the parents are joined at both ends. A random number decides the point in which to cut the ring and create the new offspring (Kaya et al., 2010).

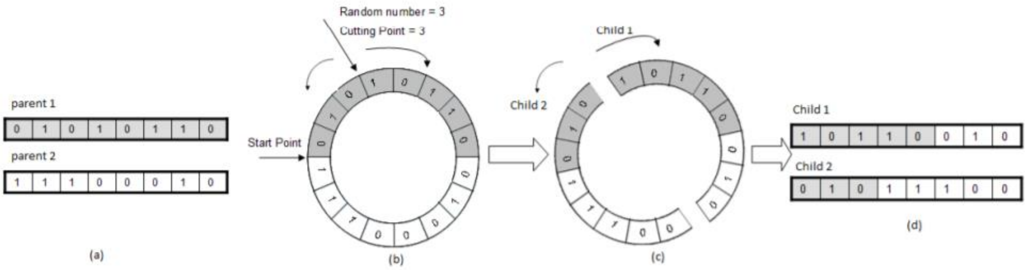

Figure 11 - Ring Crossover Diagram (Kaya et al., 2010).

Mutation

Generally, mutation occurs after the crossover operation, as presented in Figure 8. It applies random changes in certain genes, also randomly chosen, to create new solutions in solutions zones that may not have been explored (Hassanat A. , et al., 2019). This exploration of new solutions, avoids the same local optimum, allowing for new optimums to be discovered. Crossover by itself would converge to the same local optimums and get stuck in them. Mutation is crucial in maintaining diversity in the population (Korejo et al., 2009).

Figure 12 - Mutation example in a binary chromosome (Beasley et al., 1993)

Selection

The selection operator, based on the rank of nondomination, defines the best set of solutions that will make the next population. In the case of NGSA-II, that contemplates the use of elitism, this selection will have the size of N/2. Half of the population, the best population, will maintain the same characteristics, and the other repeated half will go to the adaptive operators, crossover, and mutation.

Before the selection phase, according with the NSGA-II method, a crowding distance is defined for every chromosome in each rank. The crowding distance was incorporated in the selection to enhance its reliability and to reduce the computation intensity. Originally, in NSGA, the preservation of diversity in the solution pool relied in a sharing function and the comparison between every two solutions. The sharing function relied in the sharing parameter that defines the largest distance within two solutions for sharing their fitness characteristics. The crowding distance offers every solution a value of density in the current front of a given solution. The total crowding distance consists in the sum of the distance obtained for every objective and for each objective is calculated the normalised distance, in the given front of solutions. Comparing them only to the corresponding neighbouring solutions in that objective. By avoiding a comparison between every solution and only a comparison between adjacent solutions in a given objective, the computational complexity is reduced and the independence from user inputs in the definition of the sharing parameters is obtained (Deb et al., 2002).

3.3 Pareto Front in Multi-Objective optimization

In multiple objective problems, there is no unique solution. Instead, there are multiple “acceptable trade-off optimal solutions”. This set is the Pareto front and a concept generalised by Vilfredo Pareto. The use of this concept as several advantages, allows for a more informed decision by the decision maker that has the opportunity to decide between multiple optimums according to its necessities. Single objective optimization leads only to one solution and ignores different trade-offs between objectives. Simultaneously, allows for a better understanding of the system from the developer perspective. It is easier to detect, in an already optimised Pareto Front, the effects and consequences of a certain input in the overall performance of the system (Ngarchou et al., 2005).

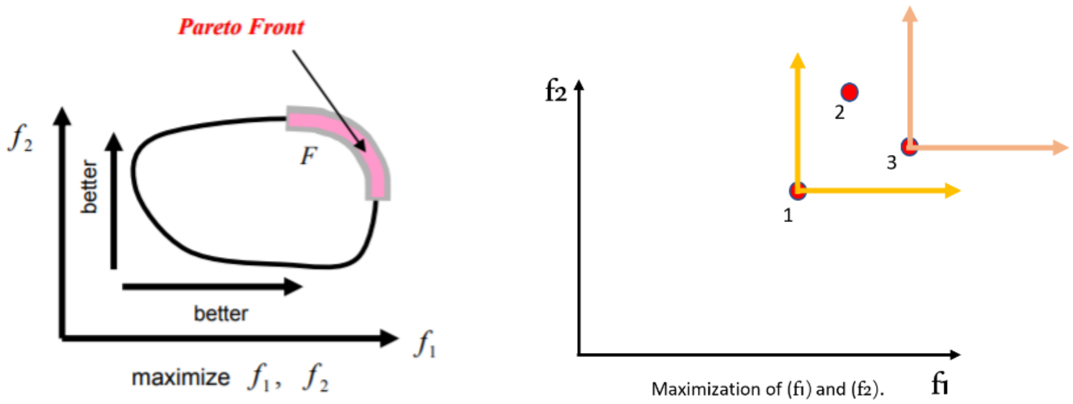

Figure 13 – a) Pareto front bi-objective (Ngarchou et al., 2005) and b) graphical example of the dominance concept.

Pareto dominance, or pareto optimality, is the characteristic that allows a solution to be part of the pareto set. It means that “there is no other solution that can improve at least one of the objectives without degradation any other objective” (Ngarchou et al., 2005). Is presented in Figure 13 a) the graphical representation of the pareto front in a bi-objective maximization. Another way to graphically visualise the domination or nondomination of a certain solution in comparison to other, is to evaluate the presence of any solution in the regions detailed. Solutions 2 and 3, being inside the region of dominance of solution 1, dominate solution 1. In the scenario of the second region detailed in Figure 13 b) the region of dominance of solution 3, does not have any solution inside. This means solution 3 is nondominated by any other solution. The same conclusion is possible to be deducted from solution 2 region of dominance.

3.4 Why Genetic Algorithms?

As stated, GAs are very efficient in dealing with high complexity problems that deal with an infinitude of solutions to be evaluated, and multiple variables with a complex behaviour (Mitchell, 1995). The optimization of a water distribution system is a multi-variable high complexity problem. The possibility of combinations regarding the position of PATs alone is extremely high. When taken into account the available PAT models in the market, the multiple rotational speeds it can run, the multiple demand patterns that the system goes through every day and the projected variations in this demand during the life cycle predicted for the system the total of possible solutions reach a very high search domain of solution space. For this reason, is clear that the decision maker for a PAT implementation needs at least some guidance to achieve desirable results.

Part of the problem is not only the size of the solution space but also the big interconnection between modifications in a network, giving the decision maker even less possibility to proceed by a rational trial and error method of finding the correct solution. What a decision maker knows in this kind of problems are the objectives it wants to achieve from the implementations. By working with the payoffs of the solutions, and not with the actions, GAs give the decision maker a stronger control of the implementation.

In the same way, NGSA-II (and Genetic Algorithm in general) being nondependent on previous information, are also nondependent on information related to the relative importance of a given parameter of the network in comparison to another. By using multiple point analyses at the same time and the construction of a Pareto Front, like the ones represented in Figure 14, this kind of method allows a final decision, and final in-depth analysis by the decision maker.

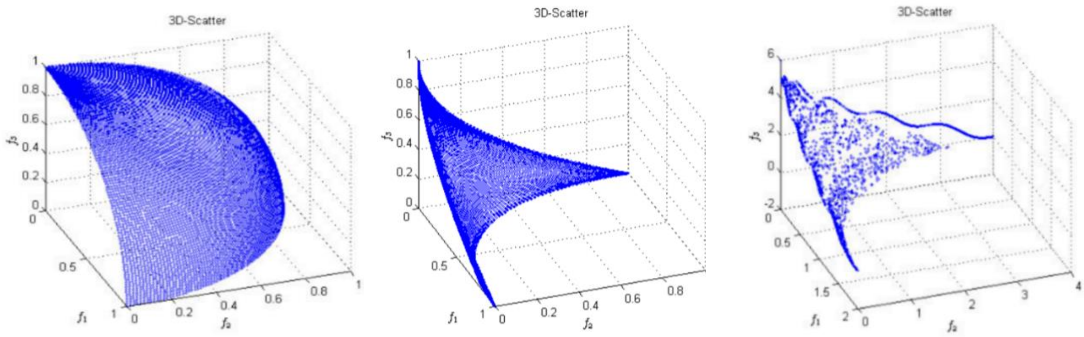

Figure 14 - 3D visualization of different pareto fronts (Ibrahim et al., 2005).

4 Methodology

4.1 Hydraulic Simulation

The hydraulic simulation of the network to be improved is performed on EPANET 2.2. It was originally created in 1993 as an open source software by the *United Estates Environmental Protection Agency.* It has the capability to solve the system of flow continuity and headloss equations to achieve a desired level of accuracy for each time step, defined by its own demands and characteristics. Originally the main objective of EPANET was to evaluate and better understand water quality and its behaviour throughout the system. Being a free software released in the public domain, it offers the possibility to interact with every function in the program through many programable languages with a command window. This enables multiple possibilities of optimization and a more automated data analyse.

The approach used to solve hydraulic flow equations is called the “Gradient Method” developed by Todini & Pilati (1987) and later extended by Salgado et al. (1988). This method has the advantage of solving the system of equations only with the simple inputs of the network system characteristics, length of pipes, elevation of nodes, a simple matrix of connectivity between nodes and the roughness coefficient of the pipes (Simpson, 2011). EPANET 2.2 is also the chosen hydraulic solver that has already been widely tested during multiple years, having a very big support and documentation freely available as an open source (Araujo et al., 2005)

4.2 EPANET-MATLAB Toolkit

The capability to interact with every functionality in EPANET through multiple programming languages is a very strong characteristic of the solver but has the disadvantage of not being a “user friendly” interface requiring a high level of computation skills by the end user, developer, or student to approach new problems with this method (Houcque, 2005). Besides skills, takes a long time to develop and test the programs and needs to be very specific to the final use, although writing a program in a high-level computation language like C++ has inherently a high versatility, and running velocity. MATLAB, in the other hand, is a software package very adapted to engineering applications from the prototyping and researching standpoint. It has a moderate cost, is very easy to program and debug, allowing for an easy interface with the user and with the data recovered from the simulation (Ibrahim D. , 2010).

To solve the stated problems and achieve a compromise between methods, EPANET-MATLAB Toolkit was developed as a MATLAB class function. It works as an interface between EPANET, and it has a skilful programming approach to interact with different optimization objectives and data analysis friendly MATLAB software. This solution allows to program in MATLAB language every interaction with EPANET networks, from modifying the network, modifying simulation characteristics and to export the desired results to be further analysed (Eliades et al. 2016).

Due to the former mentioned advantages, the program used in this research to incorporate the hydraulic simulation and its interaction with the genetic algorithm runs in MATLAB code and interacts with EPANET and the network by the interface EPANET-MATLAB toolkit.

4.3 Objective Functions

To select the best individuals in the solution space created by the genetic algorithm and the hydraulic simulation, a competition amid objectives must take place. It was chosen for this study the route of single-phase optimization, meaning variables defined and the corresponding impact of them in the GA adaptive phase is the same and executed in the same iteration.

Many more objectives can be applied to the routines in this kind of algorithms, even conflicting ones (Deb, 2014). The main objectives that should be achieved from the installation of a PAT in a water distribution network and utilized in this study are:

- Regulation of pressure in the network.
- Production of electricity.
- A fast payback period of the investment.

An evaluation of the fitness of every solution needs to be done taking into consideration every objective specified.

4.4 Fitness function – Pressure regulation

An initial approach to the pressure regulation function was made with an extrapolation of the methods used in multi-objective optimization of water networks with the implementation of Pressure Reducing Valves (PRV) (Araujo et al., 2005). These methods need the use of a weighted fitness function to take into consideration the total number of valves installed. Since in a PAT application the reduction of the number of PAT installed is not a direct objective. Because the true objective is the payback period and the economic feasibility of the installation, this PAT reduction objective was discarded. In the same way, weighted parameters are against the principles of robustness and independency from previous information that the GA defends.

For the mentioned reasons, the pressure function used was based on the *Root mean square error (RMSE)* (H. M. Awad, 2005). The RMSE is calculated between the pressure in the nodes that is given by the hydraulic solver in each iteration and the recommended minimum allowable pressure in the network or desired target pressure. In this research the target pressure remains constant during the simulation. The use of different requirement of pressure for different areas of the water system could also be applied in the GA. The pressure fitness function is presented in equation **(5)**

$$F.Pressure = [\frac{1}{n}\sum_{j=1}^{n}(h_j - h_{ref})^2]^{1/2} \qquad \textbf{(5)}$$

Where, $\boldsymbol{h_j}$ is the pressure at node (j) in a given time, **n** in the number of nodes in the networks and $\boldsymbol{h_{ref}}$ is the reference pressure assign for the network.

Constraints and Limits

RMSE has the advantages of always offering positive solutions that create a more robust system, less prone to errors in the sense that output values are more predictable. By multiplying the difference between desired pressure and the actual pressure to the square, the result provides an automatic valorisation of smaller errors and a natural penalization of nodes that have a very high pressure. This sensitivity to outliers is essential in the valorisation of the best results (Pontius et al., 2007). Constraints can be considered as implicit bound constraints, explicit variable constraints, and implicit system constraints (Zidan et al., 2017).

The **implicit system constraints** are the foundation of the system solvability. The conservation of mass **(6)** and the conservation of energy **(7)**.

$$\sum_{j} Q_{ij} - D_i = 0 \qquad \textbf{(6)}$$

$$H_i - H_j = h_{ij} \qquad \textbf{(7)}$$

- Q_{ij} is the flow in every connection to each node;
- D_i is the demand in each node;
- $H_{i,j}$ is the head in the respective nodes connected by a pipe;
- h_{ij} is the loss coefficient in the connection pipe;

The **explicit variable constraints** are related to the PAT data base and respective speeds. A library of 7 different PATs was used in this research as variable to the system optimization and 8 different rotational speeds for each PAT were used. In total, 56 different operation H-Q characteristic curves were provided for the optimization of water system.

The **implicit bound constraints** composed of restrictions applied by the user for the behaviour of the water network. They include possible velocity and pressure limits. The only ones applied in this research were pressure constraints. To simplify, the use of the positive values of RMSE was utilized. When a pressure limit, in the down and upper limits was achieved a penalization was applied to the overall fitness value of the system. The values were multiplied by a constant of penalization to be removed from the region of pareto. If the low pressure limit is achieved by a solution, the penalization excludes this solution with a high multiplier, meaning that for low pressures there is no tolerance by crossing the limit in the search space. This was applied because low pressure is even less desirable than the high pressure because it eliminates completely the system propose of supplying water.

4.5 Fitness Function – Cost/Payback

On its own, energy production from a PAT should be viewed, at least from one perspective, as business like solar farming, micro-hydro generation or other decentralised form of energy production. Having rentability is a crucial step to be implemented and explored. It should not be viewed only as an alternative to PRVs that as the possibility to generate some extra income over the years. PAT recovery systems in water networks can have the possibility of being a productive investment that, besides regulating the water network, can stimulate investors outside the water supply companies to invest in these solutions.

A function of cost alone, and its minimization, would not be representative of the rentability of the project since higher cost could be offering a higher production of energy. The same situation occurs with the minimization of PAT installations, as already been done when applying GAs to PRVs systems.

For this research, to have comparison between solutions, a cost per kW of energy produced was calculated for the fitness function. The decision to use cost per energy produced was made because it is conservative and above all accomplishes the comparison aspect of the pareto solutions and the absolute values in this situation were not taken as a priority. The cost function was imported from Novara et al. (2019), where the authors compiled the cost of 301 Radial and 42 Vertical Multistage pumps/PAT with a high variation of Best Efficiency Points (BEP) and working conditions. A similar compilation was made with more than 286 asynchronous induction generators, with one, two and three pairs of magnetic poles that correspond to 3000, 1500 and 1000 revolution per minute respectively to run at an electrical frequency of 50-Hz. With the decrease of nominal speed there is an increase in cost of the generator, with special emphasis when is required to step up to 3 pairs of magnetic poles. The data was recovered from the chart in Figure 15.

A function of cost was broken into two regions: (i) from 0kW to 1kW; (ii) for > 1kW. Multiple points were introduced in an Excel chatter plot, and reasonable function, similar to the behaviour of the scatter plot in Figure 15 was defined. For the first region, a third-degree polynomial function was very similar to the behaviour and in the second region it was more suited an exponential function. Both were introduced in MATLAB as Fitness function.

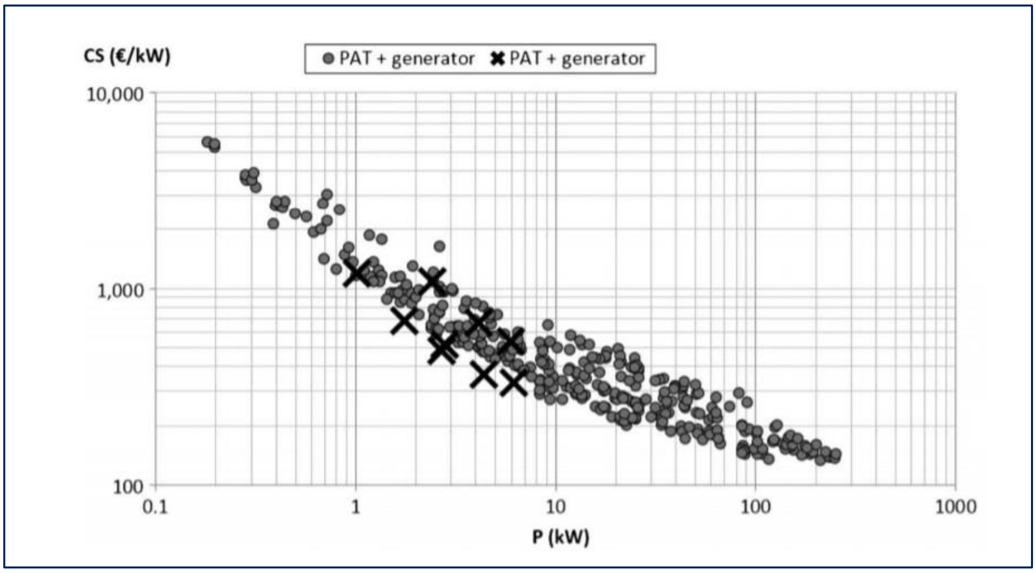

Figure 15 - Comparison of Cost/kW from multiple manufactures (Novara et al., 2019)

For region (i) from 0 kW to 1 kW:

$$Cost\left(\frac{€}{kW}\right) = -17512P^3 + 38193P^2 - 28846P + 9448{,}3 \tag{8}$$

For region (ii) from 1kW to 100kW:

$$Cost\left(\frac{€}{kW}\right) = 1498{,}4P^{-0.686} \tag{9}$$

4.6 Fitness Function – Energy Production

The last fitness function measures the accumulated electric power produced in the network. During the procedure, later described in PAT characteristic curves, the power curve from each PAT, Figure 16, at the correspondent velocity was incorporated in the input data received by the GA in the optimisation process. The curves were imported using a matrix format. The number of points that define the curves is similar to the ones defining the characteristic curves of each PAT, with 7 points in the relevant curve with extra points in the penalization/neutral region as demonstrated later in Figure 20.

To recover the hydraulic power removed from the system in each PAT, the fitness function uses the power curve data that is already incorporated in the GA library. After locating the correct curves of the

PAT model and rotational speed for a given time step, the fitness function defines the generated power by interpolating the PAT flow that came from the hydraulic simulation with the values on the power curve.

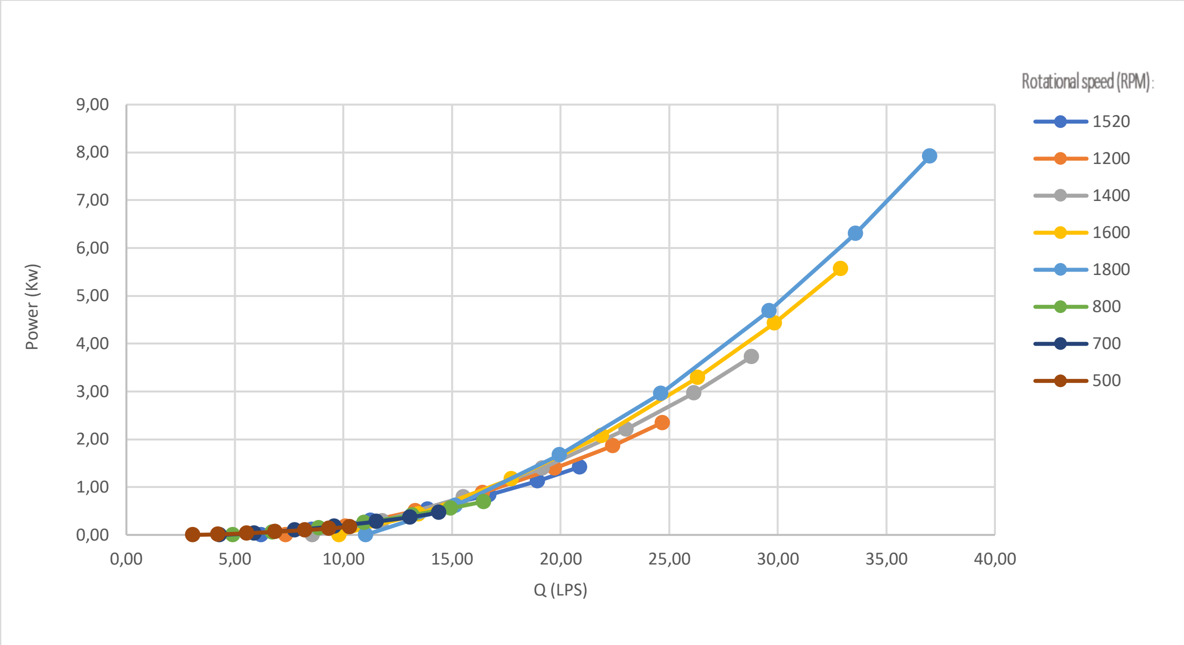

Figure 16 - Power curves for multiple specific speeds of Etanorm 65-160.

5 Case study

5.1 Preliminary approach

In this research, it is developed an approach for the use of a GA in a multi-objective problem to the implementation of a PAT solution to control the excess of pressure and the use of the recovered flow energy to create electric energy. It was used the developed EPANET-MATLAB, to create a software in MATLAB language to interact and process the data provided from the hydraulic simulations in EPANET. The language was chosen due to the already familiarity and the presence in engineering applications, being a great prototyping language very user friendly to decode errors, process data and visualise results with the possibility to use the EPANET-MATLAB toolkit. It was created an example of a water supply system to simulate the concept and plausibility in a reduced and controlled environment. Multiple PATs were analysed and compiled into a data library to be used by the GA.

Figure 17 - Section of Funchal water distribution network in an EPANET model.

The first attempts to apply the algorithm used in this research were done in a stretch of a water distribution network from a real life scenario in Funchal, Portugal, as displayed in Figure 17. Immediately, it was clear that the method used, that consists in the hydraulic solver used and its interface with MATLAB programming language, does not allow for quick enough speed in the resolution of big networks, even in this hypothetical scenario it would be impractical. Depending on the computational processing speed, that inherently fluctuates, a complete genetic algorithm routine with 200 generations would take years to run in a big and complex network. It is important to reiterate that the lack of speed in dealing with complex networks is because the MATLAB-EPANET toolkit applies the changes in a network file and does not work directly with the solving of the hydraulic matrix. The smaller the network, the faster the functions in MATLAB can change the network characteristics to simulate the new proposed solution/chromosome and obtain the fitness results. The inverse applies to the big water networks on which the MATLAB functions are slow to change the water network characteristics. In the same way, originally the simulation had a duration of 24 hours to simulate the variability of demands and changes in daily operating conditions. Generic Purpose Valves (GPV) are the elements that allow to simulate a turbine behaviour in EPANET. These valves have the exception, within the rest of EPANET accepted elements, of not allowing simple controls to automatically change its definitions during the simulation, only from OPEN to COSED, or vice-versa. This incapability adds a level of processing time since that, for every hour, is needed a “new network”.

For these reasons, a different route had to be taken to accomplish the desired, at least conceptual, objectives. The simpler example network was used, and the duration of the hydraulic simulation is now only 4 hours. The variability of demand conditions is also applied in the 4 hours period.

5.2 Hydraulic Network

The supply network created aimed for a system with variable flow conditions and excess pressure. It was design in the EPANET workspace consisting initially in 20 links connected to a reservoir on top of a hill. Three different district metered areas (DMA) existed in the network with every node and link inside each DMA at the same elevation. The DMAs selected in Figure 18 were supressed from the optimization because the flow conditions would not create suitable conditions for a PAT application. The correspondent demand was transferred to the node in the main supply system. The optimal region for a PAT application has therefore 10 links where a PAT may be installed by the optimization algorithm. The head created by the reservoir (node 1) is 200 m and the following ones downstream go from node 1 to node 10. The general representation of the network is below in Figure 18.

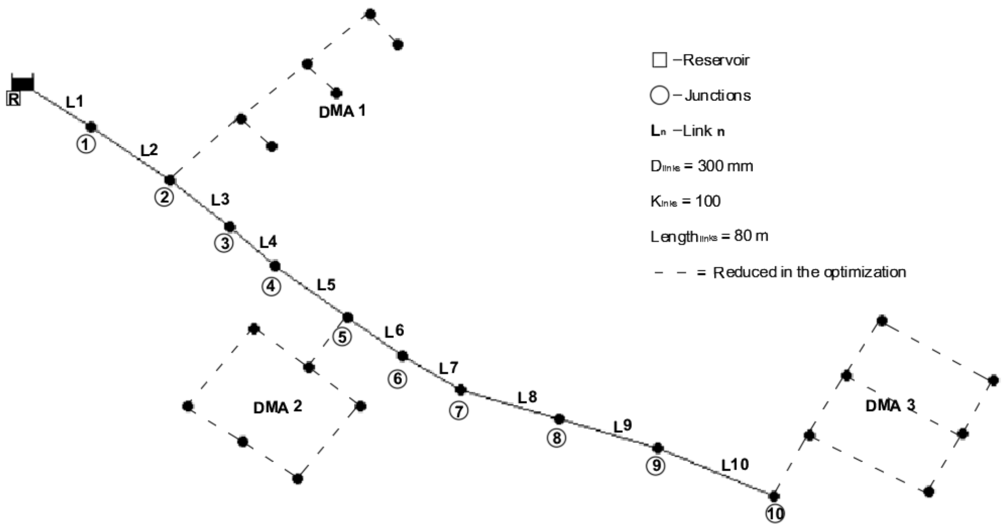

Figure 18 - Representation of the components and general structure of the network.

To achieve excess pressure in the network by elevation changes a vertical step between nodes was created with increments of 20m. The profile of the supply network is shown in Figure 19. This way excess pressure is ensured. To maintain pressure, and create a more controllable scenario, the diameter of the pipes throughout the water network was maintained in the predefined EPANET 300mm.

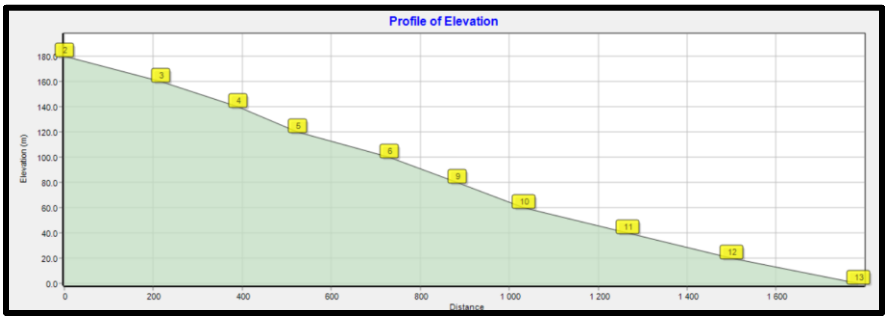

Figure 19 . Profile of the supply network.

The Hazen-Williams formula was used to define the headloss as function of the flow in each pipe. For the correspondent roughness coefficient, an average value of 100, similar to steel pipe characteristics, was used. The connections have the length as detailed in Figure 19. For every node, a base demand was defined. For this research, the demands were chosen to be generally compatible with the available PAT library. In Table 1, the demand and the elevation for each node are presented.

Table 1 - Demand and elevation by Node.

Node ID	Base Demand (LPS)	Elevation (m)
Reservoir	-	200
1	3	180
2 – DMA1	3	160
3	3	140
4	3	120
5 – DMA2	3	100
6	3	80
7	2	60
8	2	40
9	2	20
10 – DMA3	5	0

5.3 Generic Purpose Valve

A GPV is a fundamental element to simulate a turbine operation in EPANET. This type of valve works dynamically, in the sense that it changes its headloss according to the flow that runs through it and the respective headloss curve (Figure 20) that is associated to the valve behaviour. The headloss curve is the respective characteristic curve of a PAT in the EPANET model.

GPVs do not have the capability, as a PRV, to change the headloss curve with the use of Simple Controls or Rule Based Controls (Rossman, 2000). For this reason, a pre-selection of the curve for a chosen PAT for every moment in the simulation was not possible, leaving only the possibility to change the network valves in every iteration of time step to have a different performance for the different flow conditions. The bottleneck in this procedure is setting up the network conditions, consisting of mainly applying new valves in the water network.

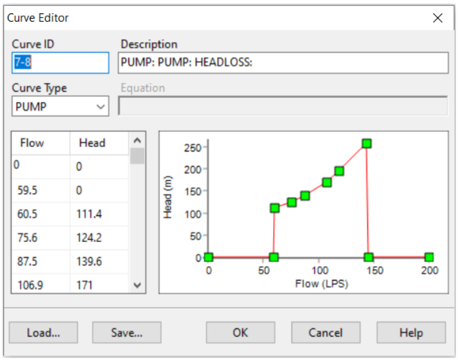

Figure 20 - PAT characteristic curve in EPANET workspace.

5.4 Hydraulic analysis

The hydraulic options chosen for the analyses are compiled in the table below. Especial regards should be taken to the definition of accuracy of 0.05. The definition relates to the proportion of changes in a iteration to the flow values while solving the hydraulic equation in comparison to all link flows. This was done because the application of multiple headloss curves, that simulate different PAT operating conditions, could create unstable resolution of the hydraulic equations. The instability can be caused by headloss curves with high derivatives or when the flow conditions are in the same region where a point that defines the curve in EPANET exists. For this reason, the smaller error compared with the usual predefined 0.01 was used. The selection of this value was done by an iterative process in cases where this instability occurred and later defined for the optimization process. The specific gravity, which is the relative density of the simulation fluid to water at 4ºC was maintained at 1. The same applies to the relative viscosity of the fluid in relation to the viscosity of water at 20ºC.

Table 2 - Hydraulic analysis options.

Flow units	LPS
Headloss Formula	Hazen-Williams
Maximum Trials	40
Specific Gravity	1
Relative Viscosity	1
Accuracy	0.05
If Unbalance	Continue

5.5 Hydraulic Times

Originally, in the initial process of the work, the hydraulic simulation duration for each hydraulic simulation was 24 hours. Like previously noted, is not possible in this setup to change the valve parameters by automatic controls. This condition dictates an increase of 24 times the duration of the optimization process. To avoid such an increase in the optimization duration, a decrease to only 4 hours was made in the hydraulic simulation duration. Since the simulation reporting time step and the hydraulic time step is 1 hour with the first simulation at 0:00, there is a total of 5 different time steps. The relevant hydraulic time definitions are shown below:

Table 3 - Hydraulic Times definitions.

Total Duration	4:00
Hydraulic Time Step	1:00
Pattern Time step	1:00
Pattern Start Time	0:00
Reporting Time Step	1:00
Reporting Start Time	0:00

5.6 Demand pattern

The demand pattern was adapted to the constrains imposed by the hydraulic times, and at the same time applying the variability that characterises WDN and the difficulty in applying a correct PAT that is required to have the most acceptable behaviour during these operational variable conditions.

Data from a demand pattern presented in Figure 21 was used as baseline to apply a similar pattern to this work. The data comes from mixed zones with commercial and residential consumers supplying more than 1 million people. The pattern was originated in an average year of monitoring in a literature reviewed network (Candelieri & Archetti, 2014).

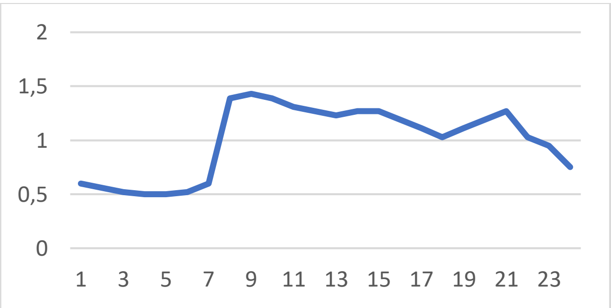

Figure 21 - Dimensionless demand for 24h in the WDN.

The data from the measured demand pattern, was imported to a spreadsheet (Figure 21). A dimensionless pattern was created and an average of the daily demand for 5 different time steps was used to replicate and adapt the natural variability of the demand in just a comprised time step. The variability, which is essential to validate the model is, therefore, preserved. The fundamental aspect of evaluating the adaptation of the PATs operating characteristics to the hourly demands is maintained with a decreased in optimization time.

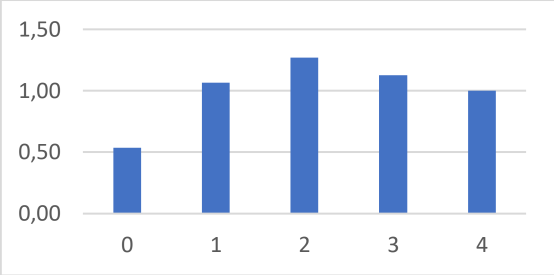

Figure 22 - Demand pattern for 4h (5 time steps).

5.7 PAT characteristic curves

The characteristic curves and the corresponding PATs used in this research come from the valves and pumps manufactured by **KSB.** The curves are already provided for pump-as-turbine mode. A library of 7 different PATs was used as variable to the system optimization. The different characteristic curves at the nominal rotation speed (N = 1520 rpm) provided by the manufacture are compiled in Figure 23. The combination of the chosen PATs was made to ensure an evenly spread operation zone. To achieve it, were selected pumps both with high head and low demand, and vice versa were chosen.

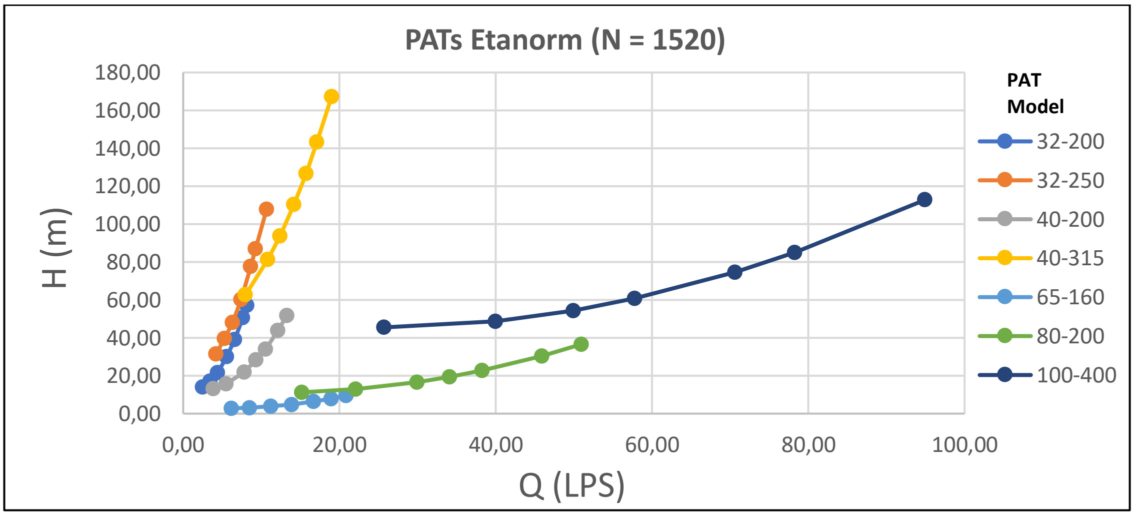

Figure 23 -Compilation of characteristic curves of different PATs.

The characteristic curves at nominal rotation speed were exported to AutoCAD and extrapolated from the provided curves. The points recovered were introduced in a spreadsheet and the characteristic curves for 8 different rotational speeds, with an interval of 200 rpm between each curve, were calculated. The rotations have a range between 900 rpm and 2300 rpm. The only exception was Etanorm 65-160 that as a nominal speed of 1015 rpm and in this case the range is from 500 rpm to 1800 rpm. For each point in each PAT the turbomachine affinity laws were applied, defining the H-Q characteristic curves and providing the behaviour on the best operating point. Also, the power curve was calculated, in the same way, using the affinity laws of turbomachines.

In the same spreadsheet, was done a preparation of the curve points to be extrapolated to the EPANET network file. Headloss curve files were created for each rotational speed in each PAT and, accordingly, were easily named (e.g. 1-2, is PAT 1 and velocity 2) to be reached when needed by the optimization algorithm. The corresponding curve, obtained from the use of turbomachines affinity laws for the Etanorm 65-160, are shown in Figure 24.

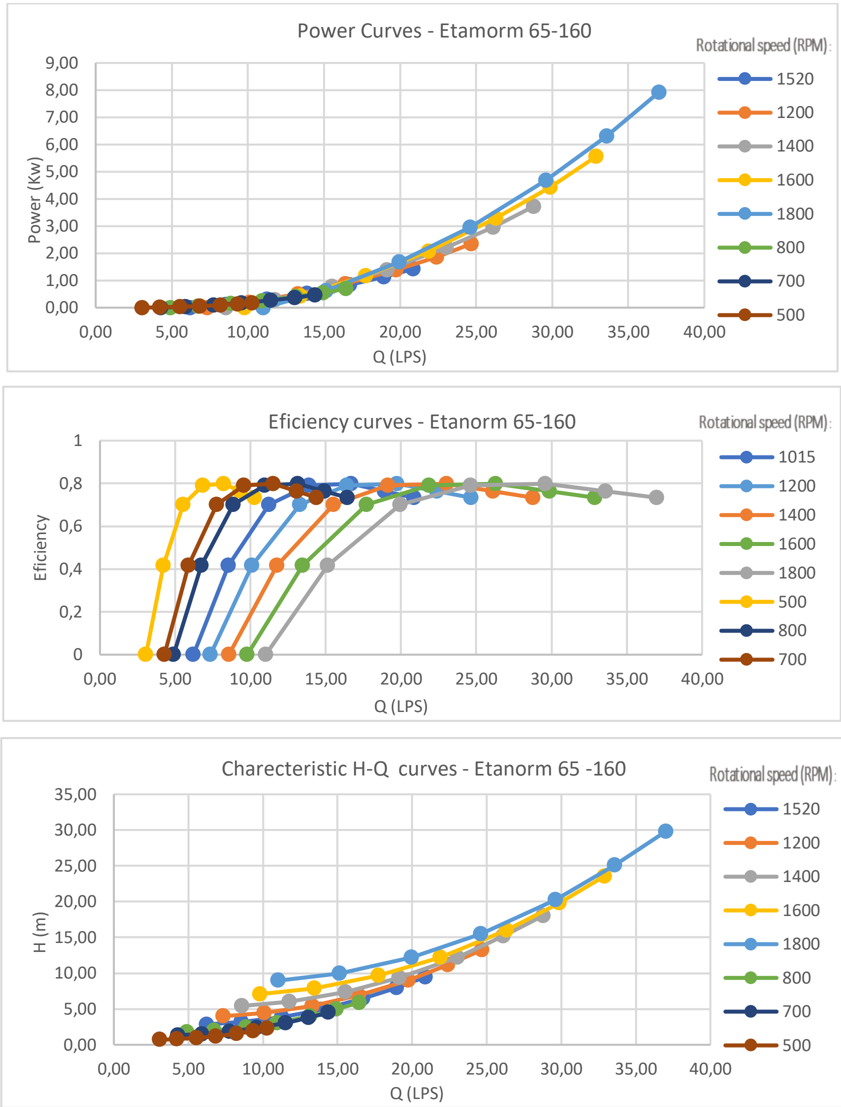

Figure 24 – Characteristic curves of H-Q, power, and efficiency of Etanorm 65-100 at different rotational speeds.

The curves applied on EPANET model have a constant line of zero headloss applied between flow zero and the start of the actual H-Q curve of the PAT. The same constant line with zero headloss is applied in the end of the H-Q curve as represented in Figure 20. This decision works as an automatic penalty in the optimization algorithm. In this way, it is more likely for a PAT to be correctly applied and to be working in the desired conditions without additional constrains to the system.

5.8 Optimization process

5.8.1 Methodology

The components of the GA process had to be adapted to the optimization and interaction with the hydraulic solver. Two main components in the process should be noted: (i) the genetic optimization algorithm and (ii) the hydraulic network edition and simulation.

For an efficient interaction between the two simulation tools, the network morphology of each solution was comprised in one common matrix as displayed in Figure 25. For each level, in the (z) axis of the matrix, corresponds a chromosome of each solution, meaning that the size of the matrix in this dimension depends on the number of elements in the initial population decided by the user. Inside of each (z) plane there is a line in the (y) axis for each link of the network and every column, in the (x) axis, is responsible for a characteristic related to the possible PAT installed in the link, whether there is in reality a PAT to be installed (On/Off) with the correspondent model and speeds in ER mode.

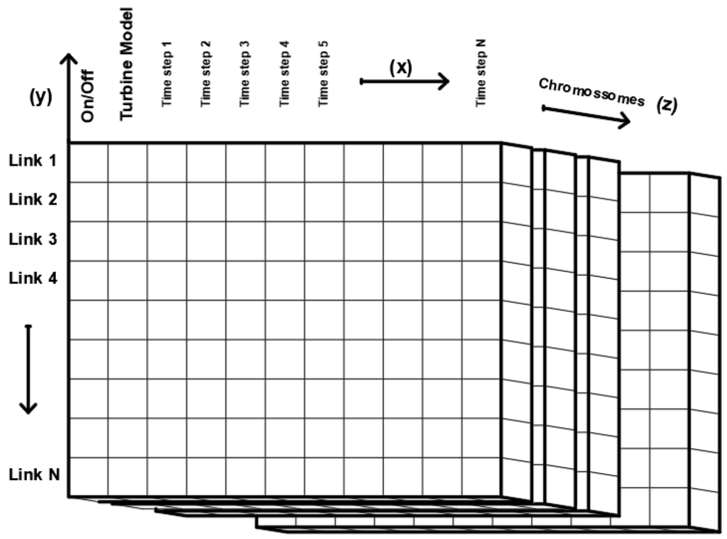

Figure 25 - Population Matrix.

By having every element of the population comprised in one matrix with a simple nomination of the characteristics, such as the binary or index connotation of the features to be stated in the network, not only it becomes easy to process the hydraulic network but allows for compatibility with simple evolution methods of mutation and crossover. A solution that was processed by the optimization algorithm is represented in Table 4. This solution was removed in the early stages of the optimization. Knowing that the mutation operator changes the values of the first column for 0 or 1, it is noticeable the already present effect of the mutation operator in the first column with the presence of already mutated values.

Table 4 - Example of chromosome/solution used in the optimization.

		Rotational speed at time step:				
PAT On/Off	PAT model	1	2	3	4	5
1,000	1	4	7	6	1	8
0,935	6	7	4	1	1	6
0,241	1	2	6	2	2	8
1,000	5	2	8	2	7	1
0,227	6	7	1	8	8	7
0,523	1	4	1	5	5	4
0,238	2	1	3	1	8	6
1,000	6	6	1	1	7	2
0,962	1	6	6	8	7	5
1,000	4	4	7	8	7	7

Hydraulic simultation and network edition

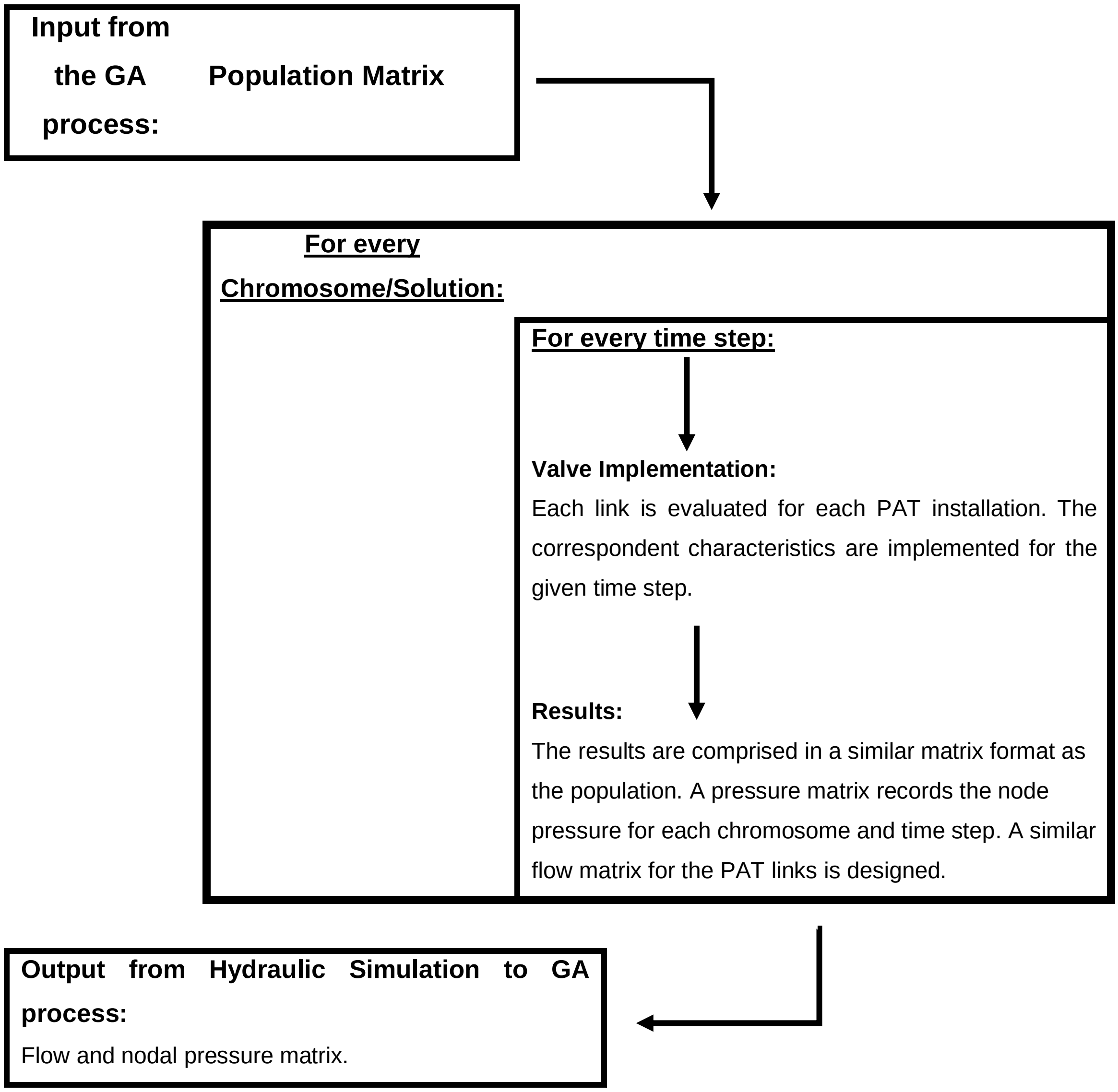

Figure 26 – Flow chart for the hydraulic Simulation and Network edition diagram.

As previously stated, the procedure bottleneck is the interface between the optimization procedure in MATLAB and the hydraulic simulation of the water network in EPANET. The GPV implementation and corresponding characteristics are the most critical step in the process, not only because is repeated multiple times (time steps*number of valves per generation*number of generations), for every time step and PAT (in form of valve), but also because it is dependent on the size of the network and the velocity of the interface toolkit. This step in presented in Figure 26 as a diagram for a clearer understanding of the work cycle.

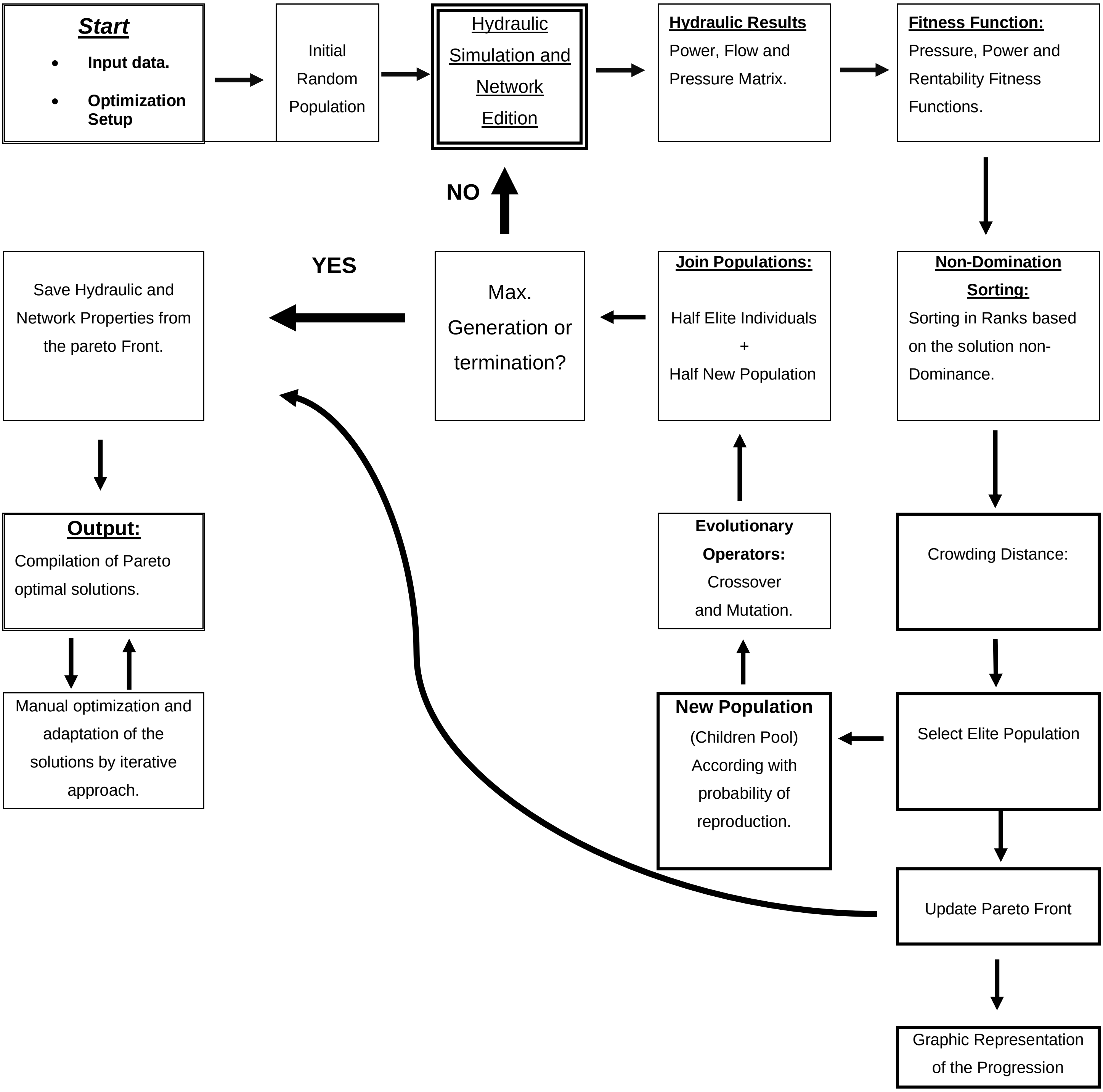

Figure 27 - General workflow of the GA in the case study.

Figure 27 describes the general methodology of the routines used in the present study. It is important to note that every major variable that impacts the system performance in the short term was incorporated in this optimization process. Meaning that the GA must deal with a complete simulation that takes into consideration not only a demand pattern but also a multitude of options in the PAT library. With this procedure a higher range of possible solutions exists and the difficulty to achieve good solutions is also inherently higher. As previously described the goal of this research is to analyse the entire network and its characteristics in only one robust system. The simulations were made with an AMD Ryzen 7 3750H (2.3Ghz) CPU where only one core was dedicated to the processing.

It is important to note that the apparent simplicity of this network is not representative of the level of possible solutions and combination of parameters that may exist in this problem. Taking in account the number of solutions permutations possible with the chromosome/solution matrix, the multiple PATs and operating conditions available and the time steps, the total number of possible solutions is **$4{,}12.10^{56}$**.

The space of solutions to be analysed, comes from a non-continuous function. A GA approach to a continuous function, where the changes in inputs can be smooth, offering a constant and gradual progression of results. In this kind of approach to a non-continuous solution space the resulting convergence is predicted to behave in a breakthrough-to-breakthrough evolution. The true pareto front is not made of continuous points, and each pareto solution may be very distinct from each other not only in terms of the fitness function output, but also in the true characteristics of the chromosomes. A geometrically imperfect pareto front is thereby expected in this multivariable non-continuous solution space. This means that when observing the pareto front in a graphical representation one could suppose that there would exist missing solutions in a certain region due to the distance between results, the pareto front general appearance and due to the lack of solutions in a certain space/region of the graphical representation. There is the possibility that the pareto front with those apparent defects could be a good approximation to the true pareto front due to the discontinuity of the true values.

5.8.2 Inputs and setup of optimization options

The inputs required at the beginning of the optimization cycle dictate the evolution of the system and at some level part of the system constrains. The elements that comprise the input data are as follow:

- Crossover Ratio
- Mutation Ratio
- Population Size
- Total number of generations
- Percentage of high-pressure tolerable region
- Reference of ideal pressure
- Reference of high pressure

- Y – Probability of not applying a PAT

Like previously stated in the methodology and components, the Crossover and Mutation are both critical elements of a GA optimization. Both depend on a user input that defines them respectively by the Crossover and the Mutation ratios The ratios are the equivalent probability of a certain characteristic in the chromosome of the solution to be modified when under the evolutionary processes to find a better suited individual. During the Crossover operations, the respective ratio was used to define the actual solutions that should take part in the exchange of genetic material to create two new chromosomes. In the mutation operator, the ratio was used freely. Meaning that a random number is associated, coordinate wise, to every gene in every chromosome of the solutions to adapt. This number was created randomly. If it were inside the range of probability defined by the ratio a mutation would occur. The mutation operator intervenes only after the crossover operator.

The effects of different ratios in the evolutionary operators is the topic of multiple studies (Hong et al., 2000; Eiben et al., 2007; Hassanat A et al.; 2019) The different methods used to apply both evolutionary operators and the corresponding ratios can change the results and the convergence of the optimization. Usually, with the use of static ratios, meaning that remain the same during the whole duration of the optimization, the values of the mutation probability are very low when compared with the crossover probability. Mutation exists mainly, not entirely, to guarantee the discovery of new regions of the solution space and crossover to optimise the individual solution in each local maximum.

In the start of any GA, there must exist an initial population that is randomly generated, in many cases (e.g. as in the case of function optimization) it does not require special attention to the randomly generated variables. In this case, there is a physical implementation of a turbine, and it could be relevant to change the initial concentration of PAT, from the analysis's perspective. A variable defined as γ was included in the input data and defines the probability of not having an installed PAT at a given link. In the creation of the new population if the randomly created variable exceeds γ, then a PAT is considered active in that link. The population size and the total number of generations are also defined. The correlation between these two parameters is also difficult to correctly determine. The traditional approach is to maintain a constant population, but studies have concluded that for small searching spaces a small population is more effective, being the opposite true to find solutions in large search areas (Rajakumar & George, 2013) (Abdelaziz, 2016). The approach used in this research was to maintain the traditional constant population.

Directly related with the hydraulic conditions of the network, objectives and constrains must be added as well. The primary input is the reference for the desired objective pressure to be present in all nodes to ensure a correct supply to the end consumer. The maximum desired pressure value was also incorporated in the input data. This value aims to create a region within it is allowed to have the flexibility to change the network characteristics in such a way that the power generation has more versatility if possible (e.g. if it makes more sense for the overall objectives to build pressure for in a line to achieve better rentability with a bigger PAT capacity). To allow for a greater focus on energy production, and at the same time stimulate a natural equilibrium in the overall pressure in the nodes, a maximum region with pressure above the reference of maximum desired pressure was created. Similar to every characteristic in the input data, the absolute value used may produce different results and the comparison between each other by running multiple simulations is always recommended. The value of 35% for the region of high pressure was used in the optimization process. When a certain solution exceeds this region of pressure a penalization is imposed in the pressure fitness values.

5.8.3 PAT curves compatibility with operating conditions

The PAT library created for this research, is purposely not selected for an in depth compatibility with the conditions present in the network. As previously said, the combination of the chosen PATs was made to ensure an evenly spread operation zone, both PATs with high head and low demand, and vice versa were chosen. Even if certain models are completely incompatible with the conditions, it would be against the ideology of robustness that characterizes the GA to pre-select only the solutions that can offer an interesting performance. In this line of thought, Figure 28 overlays the multiple characteristic curves of the PAT Library with the flow present in each stretch. The flows that are widely available are represented in green. The red regions represent flows that are not available in the network.

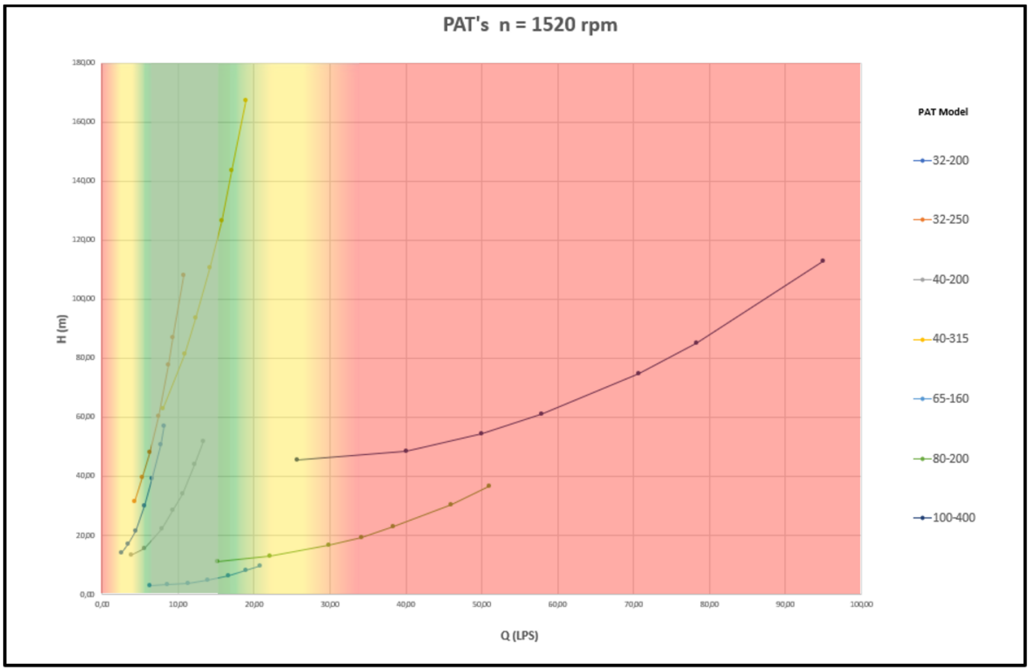

Figure 28 - Pat Library at 1520 rpm overlaid with initial hydraulic flow conditions in the network.

By observing Figure 28, it is possible to previously conclude, even though the figure only shows the standard rotational speeds of the PATs in a direct interpretation, that the majority of low flow – high head PATs are the ones that mostly adapt to the majority of hydraulic conditions .Although, flow wise, this type of turbines have a great compatibility, they can offer higher headloss than recommended in the network. Contrary to the previous group, in the high flow–low head PATs, may be less compatible flow wise but with a higher compatibility in the headloss that is induced in the network. Etanorm 100 –

400 has a very low area of influence in the characteristics offered by the network and therefore in less likely to be used by the optimization in a solution. The other two options of low head PAT, in comparison to the later one, are more suited but also not ideal since they cannot fulfil every region of the imposed hydraulic conditions. However, the Etanorm 65-160 could offer a great solution only in a high rotational speed achieving thereby the desired headloss for the links of the case study system, but at the same time has a less desirable ability of controlling the pressures in the network in low demand hours. This difficulty in finding the correct system solution, in such a simple network, shows the complexity of tackling such a problem in a larger scale. Beforehand, it is possible to predict that the evolution of the pareto front will converge more likely in solutions more dependent on high head PATs, based on the above approach of overlaying the characteristic curves and flow availability the evolutionary algorithm should have a tendency to use on one hand, since there are more types of high head PATs to choose from that can be suited to the system. In these cases, it is expected that, to maintain the pressure inside the inferior parameters of the network, it should have high pressure build up zones to have then a PAT installed that would remove the high pressure accumulated. In the other hand, a solution based on a gradual use of low head PATs is also expected to be applied since it could be a gradual use of this turbomachines in many links that would maintain adequate levels of pressure and a gradual accumulation of power generation in each PAT.

5.9 Optimization Results

During the optimization, the pareto front results for each generation and their conversion were registered and presented in the MATLAB interface as shown in Figure 29.

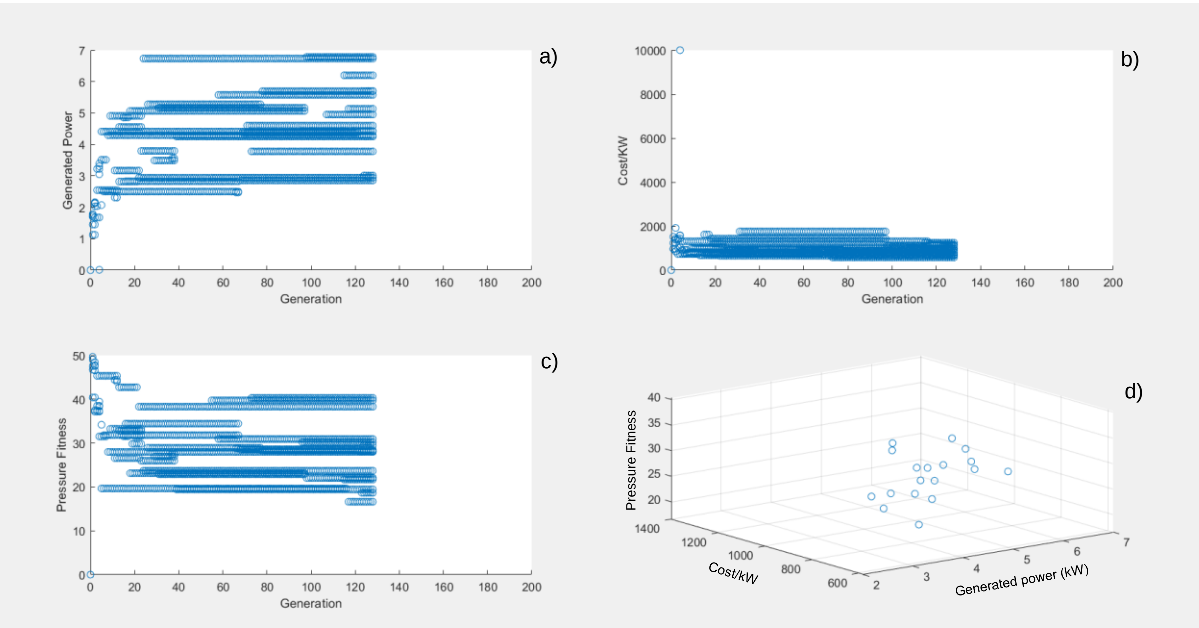

Figure 29 - Pareto Front representation and the corresponding results for each generation. a) Generated power (kW); b) Rentability (Cost/kW) c) Pressure Fitness d) Final representation of every solution in the pareto front.

In each generation of the GA convergence, the pareto front for that generation was saved and each solution represented as one circle with the corresponding fitness values associated. For Figure 29 a), b) and c) only the better results at the time of the respective generation were selected. Solutions that remained dominant for multiple generations create a line made from constant points of the same pressure fitness. When a solution is no longer present in the next generation in the graph, it means it was surpassed by another solution created with the evolutionary operators.

In Figure 29 a) the generated power (kW) for each pareto front solution in a certain generation is presented. The fitness function results that represent the cost per energy unit of each solution and the pressure fitness results are represented in the same way in Figure 29 b) and c) respectively. In Figure 29 c), the current pareto front for each generation was presented in a 3D space and updated for each generation of the GA, allowing for an easy interpretation of possible relations between solutions and fitness functions. With no previous interaction, by the reader, with the values in a dynamic graph it is difficult to visualise in this 2D format the shape of the pareto front scatter of point, therefore, reference lines were drawn to help visualise the tendencies of the pareto front final shape in Figure 30.

With the referred processing power, the analysed optimization took approximately 390 hours, which correspond to 16.25 full days, to achieve the results in Figure 29.

5.9.1 Evolution and convergence

In Figure 29, some observations on the convergence of the pareto front can be made regarding the predictability of the results:

Previously in the optimization process, the possibility of having a breakthrough-to-breakthrough optimization was proposed due to the non-continuity of the solution space. This phenomenon was indeed observed in the convergence of the solutions. It is clear in all the 2D figures (Figure 29 a); b); c)) that it took place according to that scenario, taking, at times, multiple generations of the GA without achieving a breakthrough. This was clearer after the initial convergence, where the effect of the crossover operator was probably less effective due to the convergence of the pareto front. Thereby, the breakthrough relied more in the use of the mutation operator to introduce the required variability to the new chromosomes in the population pool.

A rapid convergence took place in the initial generations of the optimization, in accordance with the convergence that it is also reported in the studies regarding the different GAs method and its application to different mathematical problems that was presented previously in 3.1 (concept and evolution of GAs) This fast convergence is justified by the high initial variability of the solution which forces better results provided by the intersection of this genetic material in the solution with the use of the Crossover Operator. Simultaneously, a hard approach to the limits of the search space could also have an influence in this original convergence. In 4.4 (constraints and limits), it was specified that the tolerance to low pressure in the nodes, would be null, meaning that in the initial population the majority of the solutions didn’t had a competitive ranking since it was considered to be out of bounds. Therefore, the reproduction operator ended with few solutions, having those solutions more probability to produce offspring. With more chances of mutation and crossover in the children's pool adding to the already high probability of an alteration to the solution to create a better one, since very few good solutions had already been discovered that could provide a competitive dominance.

The ER, in the first interactions had also a very small adaptability, only later in the phase were the solutions in the pareto front became more stable the regulation of the appropriate rotational speed for each PAT and each hour of the day started to have a permanent effect. Before this phase, a regulation in rotational speed could be very quickly surpassed by a substitution in PAT model or simply the domination of other solution.

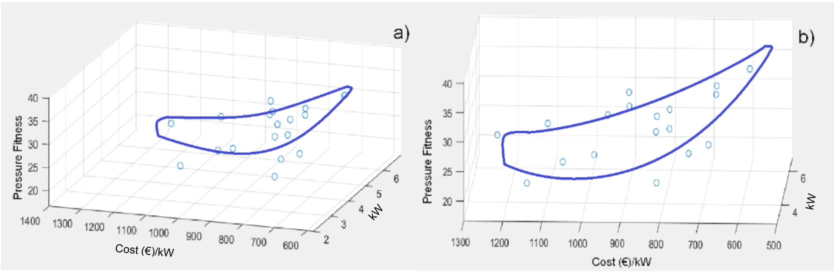

Figure 30 - Pareto front general shape emphasised in different perspectives a) and b).

5.9.2 Analyses of the results

Regarding the previous observations, there are conclusions to take from the relations between fitness parameters and their convergence:

- A clear relation that could be previously expected is that with higher power generation the lower the fitness pressure is. It is a simple and straight forward condition, that although simple, is a testimony of the correct behaviour of the optimization algorithm. Logically the reduction of pressure is equivalent to the reduction of potential energy in the water network. By removing that excess potential energy, even in a scenario that the PATs would be working in undesirable efficiency conditions the energy recovered would have the tendency to follow that inverse route.

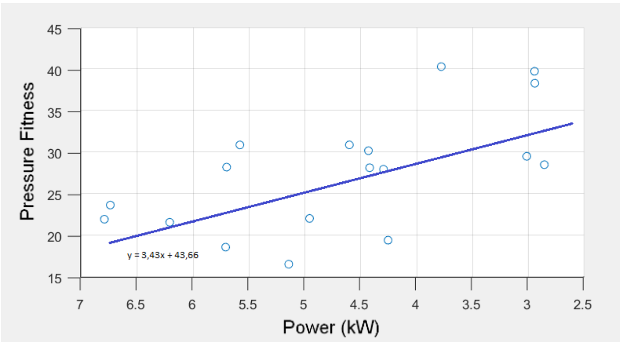

Figure 31 - Relation between pressure fitness and power generated.

- Another relation between solutions that also is presented by the pareto front is the higher rentability, or higher cost, that comes from a better pressure regulation in the water network. By analysing the recovered data, the solutions that have better pressure regulation have followed the logical route of applying more small power PATs in more links of the network to create less accumulation of back pressure. Since the rentability fitness function comes from a survey of multiple PATs, as detailed in (4.5 - Fitness Function – Cost/Payback), and the function used provides higher cost for low power installations the result is an inevitable reduction in the rentability potential.

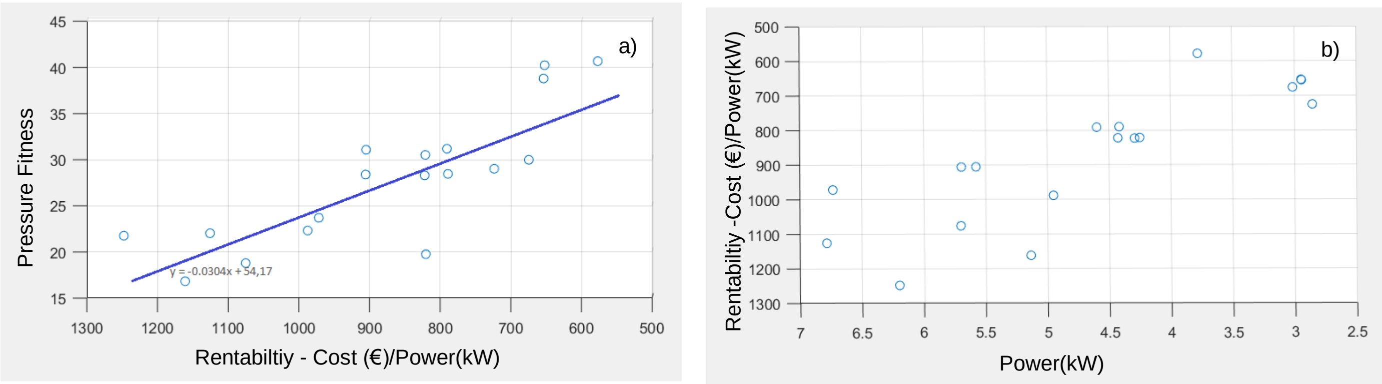

Figure 32 – a) Relation between pressure fitness and rentability. b) Relation between rentability and power generated.

- The rentability of the solutions in the pareto front is also lower, meaning a higher installation cost per energy unit the higher the power generation is represented by the linear regression in Figure 32 a). This comes in agreement with the previous point.

5.9.3 Potential influence of modifications in the results

The generations, in Figure 29, that remain in general the same, waiting for the inherent probability associated to this optimization method to act and generate a better solution for a faster convergence create the question whether an adaptive mutation and crossover ratios could have an impact in the convergence of resulting pareto front. By using adaptive mutations ratios, either a predefined transformation according to the number of generations or the continuous adaptation to the modifications in the pareto front, one could create an incentive by improving the mutation ratio when the pareto front starts to stabilize. Hence, the variability is forced to be induced in the pareto front and could accelerate either the discovery of new regions in the solutions space or could improve the tuning of the ER definitions for each time step.

The no tolerance approach described previously in 5.8.1 - evolution and convergence and in 4.4- Constraints and Limits may have an impact, not only in the probability of the first generations, but more importantly in the approach to the solutions make towards the optimization. In the approach used in this research the convergence only can occur from the high-pressure region to the low-pressure, not allowing for a convergence from both sides of the spectrum, and the low-pressure solutions are considered immediately out of bounds and do not have a reproductive chance. In the case of allowing the existence of solutions that are outside the possible solution space of the water network because of the low pressures in the nodes, there is the possibility that it would create more variability in the pool of solutions that have reproductive possibilities and that could in future generations evolve, in the correct direction, and became a completely viable solution in the pareto front. An approach to these results with a standard penalization, like when it is applied for the case of a too big high-pressure region may not be enough, since it may offer to much equality between solutions. In one case the solutions are actual viable solutions, and in the other where there are negative pressures in the water network are not physically possible or adequate to the supply of water. It is possible, that an approach using more criteria to the evaluation of solutions, giving them the option of reproduction but not allowing for an official first rank in the pareto front could be a way of opening a window to more variability in future works.

A critical aspect that was already justified in item 5.0, is the small population size and equally small number of generations used in this research due to the performance of the algorithm running in MATLAB and using the EPANET-MATLAB Toolkit. A critical aspect of optimizing a pareto front in a big search space, as this study and especially in the case of a full-scale water network, is the size of the population. A central component in the concept of GA is the mass testing, competition, and viability to reproduce at large scale. Bigger populations create more probability in each generation of finding a new offspring that could be the successor of some individual in the current pareto front. The children pool only has half the size of the original population, meaning that if a certain pareto front achieves a similar size, the reproductive probability although high in comparison with non pareto solutions, loses its influence leaving pareto solutions, individually, with very little reproductive power since the slots available for reproduction in the children’s poll are restricted. By having a very large population in comparison to the pareto front, the solution has multiple opportunities in just one generation to suffer the effects of the evolutionary operators and thereby have more chances of creating solutions (e.g. in the worst case scenario of having the same number of elements in the pareto front as the size of children pool, each solution would only have one chance in each generation of creating viable offspring). In the adopted population in this study, where it was necessary to run the optimization with the available computation performance, the pareto front gained dimensions that may have impaired the reproductive effort. This may also be a cause in cases where the convergence did not develop for multiple generations at a time. In future works, a more efficient system must be adopted to have the possibility of reducing the lack of reproductive ability with the use of large populations and therefore to converge faster in time and number of generations.

5.9.4 Optimal convergence of the solutions

A reference to evaluate the convergence of the optimization in this research is based on the pressure fitness function, since it is the only quantifiable fitness function due to being unknown the true pareto front for studied system.

The arbitrary average difference of 10 m w.c. in every node was used as a very good reference of pressure management and with these values of pressure it was obtained a final value of pressure fitness to compare the results. Other reference values were also obtained for 20, and 35 m w.c.. If the pareto front achieves the region of no penalization, an "artificial" drop in the pressure fitness value would happen in the scale of 100X inferior. The penalization for excess nodes with high-pressure above the desired value in the water system consists on the multiplication of the pressure fitness function result by a penalization constant, for this study is 100. In these reference values the penalization is added to maintain the values in the same scale for a comparison.

Table 5 - Simulation of different pressure results and correspondent pressure fitness function results.

Excess Pressure/node (m w.c.)	10	20	35
Pressure Fitness Function	0,6	1.9	5.7

In Table 5, the side-by-side comparison of the values created by the fitness function show the natural penalization of the effect of error amplification that the square factor of the fitness function creates. In comparison with best result optimized (pressure wise) with a pressure fitness of 16.62 it is clear that more work has to be done in calibrating the procedure used but simultaneously demonstrates that it is a viable system.

Although the results appear to be a long way from the references of Table 5, one must not discard the possibility that due to the network profile, the demand pattern and the PAT options available the values presented as reference could be very difficult to achieve or even can be out of the solution space being therefore impossible to achieve.

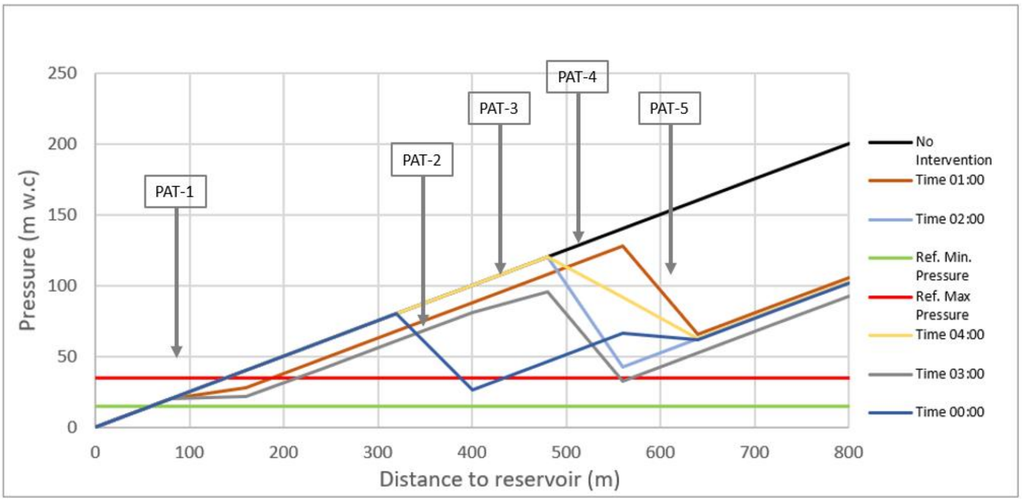

Figure 33 - Pressure profile in the network for each time step.

The profile representation of the water network solution that achieved the lowest fitness pressure result is presented in Figure 33. The pressure profile for every time period is represented. The range of optimal pressure is represented in the graph as a red line for the upper limit, the reference of high pressure, and the green line for the lower limit which is the reference of minimum pressure desirable in the water network. Table 6 refers to the speed of the PATs referred in the pressure profile of Figure 33.

Table 6 - Rotational speed of the optimized PATs at each given time period referent to Figure 33.

		Rotational speed (RPM) at time:				
	Model	Time 0:00	Time 01:00	Time 02:00	Time 03:00	Time 04:00
PAT – 1	80-200	1700	1100	900	1700	1500
PAT – 2	80-200	1900	2100	1900	900	1900
PAT – 3	80-200	1900	1100	1300	1100	1100
PAT – 4	40-315	2100	2300	1300	1300	1100
PAT - 5	32-200	1500	2100	1300	1100	1900

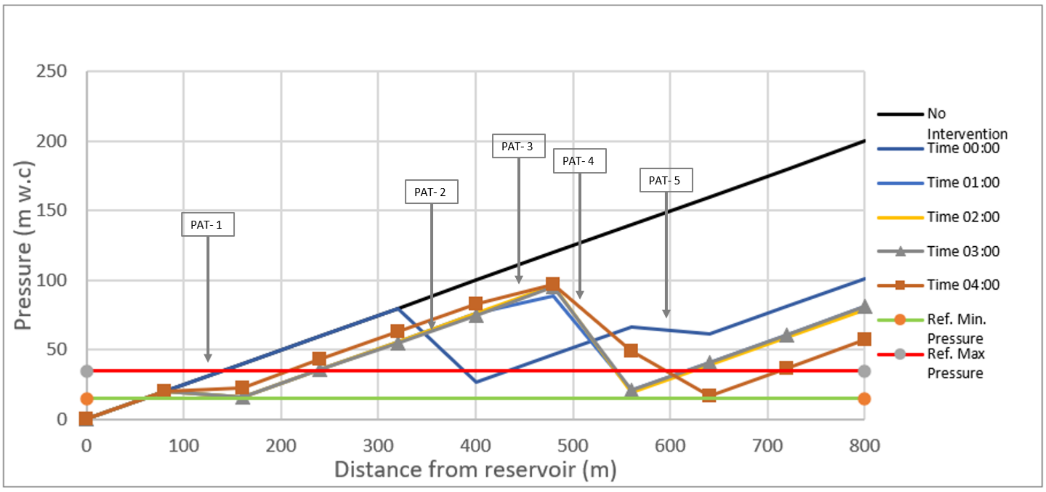

Figure 34 - Pressure profile in the refined network for each time step.

A refinement post optimization of the PAT characteristics was executed for the solution with the best pressure fitness. The pressure profile for the refined solution for each given time step is represented in Figure 34. Small enhancements were made in the PAT rotational speeds. The GA optimization seeks the overall best set of solutions to the water network, therefore, in the final stretch of optimization where the mutation operator is more important, the duration of the conversion may be slower. A fast refinement of the already simplified solution after the optimization process can improve the results that may take multiple generations to improve with the GA. Table 7 details the speeds referent to the pressure profile of Figure 34.

Table 7 - Rotational speed of the optimized PATs at each given time period referent to Figure 34.

		Rotational speed (RPM) at time:				
	Model	Time 0:00	Time 01:00	Time 02:00	Time 03:00	Time 04:00
PAT – 1	80-200	1300	1900	2100	1900	1900
PAT – 2	80-200	1900	1100	1300	1300	1300
PAT – 3	80-200	900	1100	1500	1300	1100
PAT – 4	40-315	900	1500	1700	1500	1100
PAT - 5	32-200	900	1500	1700	1500	1100

By adding the integral of the pressure curve of each time step in the “no intervention” scenario, in the optimized profile of the water network and in the profile refined after the optimization a comparison of the results was made. The “no intervention” case is the benchmark with 100% of the network pressure. The pressure results provided solely by the optimization algorithm provide a reduction to 78% in the network pressure. After the fast refinement of PAT speeds of the optimization results there is a reduction to a noticeable 59% of the original pressure in the water network. A clear improvement was made regarding the original pressure conditions.

6 Conclusions

6.1 Conclusions of the developed research

The control of water scarcity must be made in both ends of the spectrum, controlling and reducing the water exploitation by effectively reducing the losses in transport and at the same time tackling the energy dependent system that is the water supply industry by reducing the need for outsourcing energy that could increment by itself the effects of climate change. The water supply networks and its demands are ever more complex, and the correct management follows the same path.

Multiple methods exist for an adequate control of the excess pressure in water networks. Genetic algorithms have an already proven capability of being able to offer solutions of pressure regulation in water networks by using PRVs. The use of PATs with its multiple regulation methods are also compatible with the concept of optimization introduced by the GAs. The same compatibility applies to the size of the solution space in this type of problem. Multiple methods and alternatives to the GA optimization operators can be studied to enhance the system capabilities.

The use of a pareto front optimization is an excellent method to avoid unnecessary bias by the user of the system and allows it to have a stronger decision power. Avoids an inevitable parametrization of weighted objectives and allows for an optimization of solutions that otherwise could be ignored.

The use of an integral approach, like the one used in this research, to optimize solutions that uses PATs as the base element in a multi-objective problem show a feasible option that could allow for an efficient optimization of large water networks. The fitness functions and restrict constrains showed a good convergence of the solutions, having nevertheless room for improvement by allowing solutions that are in the negative pressure region to reproduce in order to improve the variability of the solutions in the pareto front and possibly the speed of convergence.

The method of combining all the information in the proposed population matrix proved to be a robust option. Allowing for a fast manipulation of the population characteristics and a flawless interaction between each step of the optimization.

The use of all fitness functions developed in this research showed an effective comparison between solutions and allowed for a competitive evolution of the pareto front. The velocity of convergence diminished during the simulation. The lack of reproductive ability of the solutions due to the size of the population or the achievement of a very optimized pareto front by the GA could be a cause for this observation.

The optimization results demonstrate a clear improvement in the pressure conditions. Besides offering adequate solutions that respect the limits of what is the acceptable solution space, it offers a direct improvement after the optimization to only 78% of the original pressure. After a refining of the rotational velocities in the solution, pressure levels of 59% the original pressure were achieved. With the use of

PATs better adapted to the conditions present in the water network, it is possible to achieve even better results.

The methodology used in this research shows effectiveness in the convergence of the pareto front and its adaptation using the evolutionary operators. The use of EPANET-MATLAB Toolkit, despite being a good solution to analyse data from water networks using a powerful mathematical software like MATLAB, is not adequate in performance capabilities to the number of network editions and simulations needed to have results closer to the true pareto and adequate populations and generations in the optimization.

6.2 Future works

The procedure used in this research could be improved in several aspects on future works, many of them already referred but, nevertheless, are here compiled.

The use of a different system to achieve the performance needed to execute an effective GA optimization is essential for a better development, and study, of the methodology presented. The required performance was not reached in this research with the use of EPANET-MATLAB Toolkit. The approach may consist in using an integrated system that deals directly with the hydraulic equations and respective matrix in the same language, without editing a network file. This fast approach solves the lack of generations and population number, ensuring the so important diversity, the possibility to evaluate the optimization in more complex networks and the use of more variables, such as, the complete time-steps and bigger PAT libraries. The use of more detailed cost fitness function can also be of great interest, ensured by also having a cost library for each individual PAT, civil works and components of the electrical regulation.

Further works should be made in the calibration of the evolutionary ratios and rigidity of the solution space limits to enhance the convergence capabilities. Simultaneously, it could be of interest to extend the optimization to solutions that not only compete for viability in the short run, but also for a complete life cycle of the solution. Achieving more simulations, with different regulations and more yearly demands that are predicted to happen in the future.

References

Abdelaziz, M. (2016). Distribution network reconfiguration using a genetic algorithm with varying population size.

Agency, E. E. (2019). Use of freshwater resources in Europe.

Alatorre-Frenk, C. (1994). Cost minimization in microhydro systems using pumps-as-turbines.

Araujo, L. S., Ramos, H., & Coelho, S. (2005). Pressure Control for Leakage Minimisation in Water Distribution Systems Management.

Beasley, D., Bull, D. R., & Martin, R. R. (1993). An Overview of Genetic Algorithms : Part 1, Fundamentals.

Candelieri, A., & Archetti, F. (2014). Identifying Typical Urban Water Demand Patterns for a Reliable Short-Term Forecasting – The Icewater Project Approach.

Carravetta, A., Fecarotta, O., & Ramos, H. (2018). A new low-cost installation scheme of PATs for pico-hydropower to recover energy in residential areas. *Renewable Energy.*

Carravetta, A., Giudice, G. D., Fecarotta, O., & Ramos, H. M. (2012). Energy Production in Water Distribution Networks: A PAT design Strategy.

Carravetta, A., Giudice, G. d., Fecarotta, O., & Ramos, H. M. (2013). PAT Design Strategy for Energy Recovery in Water Distribution Networks by Electrical Regulation.

Chacón, M. C., Díaz, J. A., Morillo, J. G., & McNabola, A. (2019). Pump-as-Turbine Selection Methodology for Energy Recovery in Irrigation Networks: Minimising the Payback Period.

Chankong, V., & Haimes, Y. (1983). *Multiobjective Decision Making.*

Clarke, B. (2010). Managing Water Loss with Pressure Reducing Valves. *Water World*.

Craig R, B., & Andes, K. (2015). Consideration of energy savings in SWRO.

Deb, K. (2014). Multi-Objective Optimization, Springer, Boston

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II.

Derakhshan, S., & Nourbakhsh, A. (2007). Experimental study of characteristic curves of centrifugal pumps working as turbines in different specific speeds.

Derakhshan, S., & Nourbakhsh, A. (2008). Theoretical, numerical and experimental investigation of centrifugal pumps in reverse operation.

Eiben, A., Michalewicz, Z., Schoenauer, M., & Smith, J. (2007). Parameter Control in Evolutionary Algorithms.

Eliades, D., Kyriakoou, M., & Polycarpou, S. V. (2016). EPANET-MATLAB Toolkit: An Open-Source Software for Interfacing EPANET with MATLAB.

Fields, N. (2015, April). Non-revenue water loss: the invisible global problem.

Fonseca, C. M., & Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulatio, Discussion and Generalization.

Fontana, N., Giugni, M., & Portolano, D. (2012). Losses Reduction and Energy Production in Water Distribution Networks.

Fontanella, S., Fecarotta, O., Molino, B., Cozzolino, L., & Morte, R. D. (2020). A Performance Prediction Model for Pumps as Turbines (PATs).

Frosina, E., Buono, D., & Senatore, A. (2017). A Performance Prediction Method for Pumps as Turbines (PAT) Using a Computational Fluid Dynamics (CFD) Modeling Approach.

Frosina, E., Buono, D., & Senatore, A. (2017). A Performance Prediction Method for Pumps as Turbines (PAT) Using a Computational Fluid Dynamics (CFD) Modeling Approach.

Girard, M., & Stewart, R. (2007). Implementation of pressure and leakage management strategies on the gold coast, australia: a case study.

Goldberg, D. E. (1989). Genetics Algorithm in Search Optimization & Machine Learning, Adison-Wesley Longman Publishing Co .Inc

Gonçalves, F. V., & Ramos, H. M. (2008). Controlo económico e energético e proposta de optimização.

H. M. Awad. (2005). Basic Study on Optimal Pressure Regulation in Supervisory Water.

Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammour, A., & Prasath, V. B. (2019). Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach.

Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A., & Prasath, V. S. (2019). Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence,* MIT Press.

Hong, T.-P., Wang, H.-S., & Chen, W.-C. (2000). Simultaneously Applying Multiple Mutation.

Houcque, D. (2005). Introduction to matlab for engineering students, Northwestern University.

Ibrahim, A., Martin, M. V., Rahnamayan, S., & Deb, K. (2006). 3D-RadVis: Visualization of Pareto Front in Many-Objective Optimization.

Ibrahim, D. (2010). Engineering simulation with MATLAB: improving teaching and learning effectiveness .

Ilicic, K., & Kovac, J. (2009). Systematic Approach in Analysis of Pressure/ Burst Frequency Relationship.

Kaya, Y., Uyar, M., & Tekin, R. (2010). A Novel Crossover Operator for Genetc Algorithms: Ring Crossover.

Korejo, I., Yang, S., Brohi, K., & Khuhro, Z. (2013). Multi-population methods with adaptive mutation for multi-modal optimization problems.

Mitchell, M. (1995). Genetic Algorithms: An Overview, Jonh Wiley and Sons .Inc

Muhammetoglu, Ayse., Karadirek, Ethem. Ozen, Ozge. Muhammetoglu, Habib. (2017) Full-Scale PAT Application for Energy Production and Pressure Reduction in a Water Distribution Network

Motwani, K. H., Jain, S. V., & Patel, R. N. (2013). Cost analysis of pump as turbine for pico hydropower plants – a case study.

Nautiyal, H., Varun, & Kumar, A. (2010). CFD Analysis on Pumps Working as Turbines.

Ngarchou, P., Zarei, A., & El-Sharkawi, M. A. (2005). Pareto Multi Objective Optimization.

Novara, D., Carravetta, A., McNabola, A., & Ramos, H. M. (2019). Cost Model for Pumps as Turbines in Run-of-River and In-Pipe Microhydropower Applications.

Perdigão, M. (2018). Implementation of Pump-as-Turbines as Energy Recovery Solutions within Water Distribution and Supply Systems.

Pontius, R. G., Thontteh, O., & Chen, H. (2007). Components of information for multiple resolution comparisson between maps that share a real variable.

Puleo, V., Fontanazza, C. M., & Notaro, V. (2014). Pumps as turbines (PATs) in water distribution networks affected by intermittent service.

Quintela, A. d. (1981). *Hidráulica,* Fundação Calouste Gulbenkian.

Rajakumar, B., & George, A. (2013). APOGA: An Adaptive Population Pool Size Based Genetic Algorithm.

Ramos, H., Covas, D., Araujo, L., & Mello, M. (2005). Available energy assessment in water supply systems.

Reis, L. F., Porto, R. M., & Chaudhry, F. H. (1997). Optimal Location of Control Valves in Pipe Networks by Genetic Algorithm.

Richardson, J. T., Palmer, M. R., Liepins, G., & Hilliard, M. (1989). Some guidelines for genetic algorithms with penalty functions.

Rossman, L. (2000). EPANET 2 User's Manual, U.S. Environmental Protection Agency.

Salgado, R., Todini, E., & O'Connell. (1988). Extending the gradient method to include pressure regulating valves in pipe networks.

Schaffer, J. D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, Proceedings of the 1st International Conference on Genetic Algorithms, Pittsburgh, PA, USA

Sharma, K. (1985). Small hydroelectric project-use of centrifugal pumps as turbines. (K. E. Technical Report, Ed.)

Sen-chun, M., Hong-biao, Z., Ting-ting, W., Xiao-hui, W., & Feng-xia, S. (2020). Optimal design of blade in pump as turbine based on multidisciplinary feasible method.

Štefan, David. Rossi, Mosè., Hudec, Martin, Pavel Rudolf, Alessandra Nigro, Massimiliano Renzi, (2020) Study of the internal flow field in a pump-as-turbine (PAT): Numerical investigation, overall performance prediction model and velocity vector analysis.

Simão, M. M. (2009). Hidrodinâmica e desempenho em turbinas de baixa potência: concepção, modelação e ensaios.

Simpson, A. R. (2011). Comparing the Q-Equations and Todini-Pilati Formulation for Solving the Water Distribution System Equations. Water Distribution Systems Analysis.

Singh, P., & Nestmann, F. (2009). An optimization routine on a prediction and selection model for the turbine operation of centrifugal pumps.

Srinivas, N., & Deb, K. (1994). Multi Optimization Using Nondominated Sorting in Genetic Algrithms.

Stepanof, A. (1957). Centrifugal and Axial Flow Pumps, Design and Aplications, Krieger Publishing Company.

Todini, E., & Pilati, S. (1987). A Gradient Method for the Analysis of Pipe Networks.

Tong Lin, Zuchao Zhu, Xiaojun Li, Jian Li, Yanpi Lin, (2021). Theoretical, experimental, and numerical methods to predict the best efficiency point of centrifugal pump as turbine.

Whitley, D. (1994). A Genetic Algorithm Tutorial, Colorado State University.

Williams, A. A. (1994). The turbine performance of centrifugal pumps: a comparison o prediction methods.

Zidan, A. R., El-Ghandour, H. A., & Elansary, A. (2017). Presssure management in water distributiion networks by multi-objective genetic algorithm.

Zitzler, E., Thiele, L., & Deb, K. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results.

Appendix

Appendix I – Optimization Algorithm

Main

```
clear;
clc;
RCross = 0.5;
RMut = 0.15;
RMutPosition = 0.2;
pop = 80;
gama = 0.8;
Generations = 200;
RefPressure = 15;
HighPressureZone = 0.35;
RefMaxPressure = 35;
run('C:\Users\Tiago Baptista\Desktop\Pasta de Teste EPANET\EPANET-Matlab-'...
    'Toolkit-2.2.0-beta.3\EPANET-Matlab-Toolkit-2.2.0-beta.3\epanet_matlab'...
    '_toolkit\64bit\start_toolkit.m');
d = epanet ('Conceito1Testagem.inp','bin');
d.LibEPANETpath = 'C:\Users\Tiago Baptista\Desktop\Pasta de Teste EPANET'...
    '\EPANET-Matlab-Toolkit-2.2.0-beta.3\EPANET-Matlab-Toolkit-2.2.0-beta.3\'...
    'epanet_matlab_toolkit\64bit\epanet2'
LinkFlow = d.getBinComputedLinkFlow;
NodePressure = d.getBinComputedNodePressure;
lig = d.getBinLinksInfo;
ned = d.getBinNodesInfo;
from = lig.BinLinkFromNode;
to = lig.BinLinkToNode;
ID = lig.BinLinkNameID;
P = lig.BinLinkCount;
Diam  =lig.BinLinkDiameters;
NodeCount = ned.BinNodeCount;
MeanLinkFlow = abs(mean(LinkFlow));
MeanNodePressure = abs(mean(NodePressure));
TNODES = FINDTERMINAL2(from,to,P);
filename = 'CriticalConceito.mat';
save(filename);
[LinkCV, HeadCV] = CoefVariation(LinkFlow, P, NodePressure, ned, Mean...
    LinkFlow, MeanNodePressure);
filename = 'CVConceito.mat';
save(filename);
[NlinkData, counter, NPAT, NPATsize, NlinkDataDim]= NetworkData...
    (MeanLinkFlow,ID,from,to,Diam,P);
[Population]=FirstPopulation(counter,NPAT, NPATsize,pop,gama);

Hor = 0;
Vert = 0;
Gama = 0;
Theta = 0;
Omega = 0;
Phi = 0;
GenCount = 0;
RepeatedCromIdx = [];
%---------------------------------


while GenCount ~= Generations
    GenCount = GenCount + 1 ;
    [NodePressure,ValvePressureFromNode,ValvePressureToNode,...
        ValveFlow, ValveCounter,ValveType]= HidraulicRun3(Population,...
        NlinkData,NodeCount,pop, counter,NlinkDataDim, gama);
    [PressureFitness, CromToDelete] = ObjPressure1(NodePressure,...
        RefPressure,NodeCount,RefMaxPressure,HighPressureZone,...
        RepeatedCromIdx,pop);
    [TotalCostPerKW, HPowerSumAverage] = ObjRentability...
        (ValvePressureFromNode,ValvePressureToNode,ValveFlow, pop,...
        ValveCounter,ValveType, CromToDelete);
    [front, obj, crom] = NONDOMINATIONSORTING ...
        (TotalCostPerKW,HPowerSumAverage,PressureFitness, pop);
    [crom] = CrowdingDistance(crom, front, obj);
    [optpop,ParetoFrontIndex, optpopsize] = SelectPop...
        (crom, TotalCostPerKW,HPowerSumAverage,PressureFitness);
    newpop = selectOpt(optpop);
    [PopulationTemp, PopulationMod] = JoinPopulations(newpop,...
        optpop, Population);
    PopulationMod = Crossover(PopulationMod, RCross);
    PopulationMod = Mutation(PopulationMod, RMut, NPATsize, RMutPosition);
    [Population] = JoinPopulationMod(PopulationMod, Population,...
        PopulationTemp, newpop, optpop);
    [Population,RepeatedCromIdx] = RepeatedCrom(Population,...
        optpop, PopulationMod);

% ----- Data for grafical representation.

for i = 1 : length(ParetoFrontIndex)
ParetoGen(i)= GenCount;
```

```
ParetoPower(i) = HPowerSumAverage(ParetoFrontIndex(i).Index);
ParetoCost(i) = TotalCostPerKW(ParetoFrontIndex(i).Index);
ParetoPressure(i) = PressureFitness(ParetoFrontIndex(i).Index);
end
%-----------------------------------------------------

fclose('all');

    figure(1)
    subplot(2,2,1);
    Hor1 = GenCount + zeros(length(ParetoGen),1);
    Vert1 = ParetoPower;
    Hor = vertcat(Hor, Hor1)
    Vert = horzcat(Vert, Vert1);
    scatter(Hor,Vert);
    xlim([0 Generations])
    xlabel('Generation')
    ylabel('Generated Power')
    linkdata on
    drawnow;

    subplot(2,2,2);
    Gama1 = GenCount + zeros(length(ParetoGen),1);
    Theta1 = ParetoCost;
    Gama = vertcat(Gama,Gama1);
    Theta = horzcat(Theta,Theta1);
    scatter(Gama,Theta);
    xlim([0 Generations])
    xlabel('Generation')
    ylim([0 10000])
    ylabel('Cost/KW')
    linkdata on
    drawnow;


    subplot(2,2,3)
    Omega1 = GenCount + zeros(length(ParetoGen),1);
    Phi1 = ParetoPressure;
    Omega = vertcat(Omega,Omega1);
    Phi = horzcat(Phi,Phi1);
    scatter(Omega,Phi);
    xlim([0 Generations])
    xlabel('Generation')
    ylabel('Pressure Fitness')
    linkdata on
    drawnow

    subplot(2,2,4)
    scatter3(ParetoPower,ParetoCost,ParetoPressure);
    drawnow;


 % ---------------------------------------------------FINAL---------
 if GenCount == 10 || GenCount == 20 || GenCount == 100 ||...
        GenCount == 150 || GenCount == 60
      varname = sprintf('OptPop.GenCount-%u',GenCount);
      save(varname,'optpop');
 end
 %----------------------------------------------------------

    clearvars ParetoFrontIndex ParetoGen ParetoPower ParetoCost...
        ParetoPressure;
    clearvars optpop PressureFitness TotalCostPerKW HPowerSumAverage...
        NodePressure;
    clearvars NodePressure ValvePressureFromNode ValvePressureToNode...
        ValveFlow ValveCounter;
    clearvars PopulationTemp PopulationMod newpop;

 end
```

Hydraulic Simulation

```
function[NodePressure,ValvePressureFromNode,ValvePressureToNode,ValveFlow...
    , ValveCounter]=HidraulicRun(Population,NlinkData,NodeCount,pop...
    , counter,NlinkDataDim, gama)

NodePressure = zeros(25,NodeCount,pop);
ValveCounter = zeros(pop);
i = 0;

for crom = 1 : pop
    for h = 1 : 24
        file  = sprintf('Conceito%d.inp',h-1);
        d = epanet(file,'bin');
        d.LibEPANETpath = 'C:\Users\Tiago Baptista\Desktop\Pasta de'...
            'Teste EPANET\EPANET-Matlab-Toolkit-2.2.0-beta.3\EPANET-Mat'...
            'lab-Toolkit-2.2.0-beta.3\epanet_matlab_toolkit\64bit\epanet2';
        ValveCount = 0;
        tic
```

```
    for i = 1: counter
        show1 = [ 'Link ',num2str(i), ' de ' , num2str(counter),...
            ' do cromossoma n°', num2str(crom), 'tempo -',num2str(h)];
        disp(show1);

        if Population(i,1,crom) >= gama %Apply GPV%
            d.removeBinLinkID(convertCharsToStrings(NlinkData(i,1)));
            name = sprintf('GPV%u',i);
            name = convertCharsToStrings(name);
            d.addBinValveGPV(name,convertCharsToStrings(NlinkData(i,2))...
                ,convertCharsToStrings(NlinkData(i,3)),...
                NlinkDataDim(i,1),(Population(i,2+h,crom)));
            ValveCount = ValveCount + 1;

        end
    end

ValveCounter(crom) = ValveCount;
% 1-ID 2-From 3-To 4-D%
lig = d.getBinLinksInfo;
ValvesIndex = lig.BinLinkValveIndex;
NodePressureTemp = d.getBinComputedNodePressure;
LinkFlowTemp = d.getBinComputedLinkFlow;
ValveFrom = lig.BinLinkFromNode;
ValveTo = lig.BinLinkToNode;
ned = d.getBinNodesInfo;
toc
try
    for node = 1 : NodeCount
            NodePressure(h,node,crom) = NodePressureTemp(1,node);
    end
catch
    i = i+1;
    InvalidCrom(i)= crom;
    for time = 1 : h
        ValveFlow(time,1,crom) = 0;
        ValvePressureFromNode(time,1,crom) = 0;
        ValvePressureToNode(time,1,crom) = 0;
    end
    continue
end

if ValveCount == 0
  for time = 1 : 24

        ValveFlow(time,1,crom) = 0;
        ValvePressureFromNode(time,1,crom) = 0;
        ValvePressureToNode(time,1,crom) = 0;
    end
    continue
end

for valve = 1 : ValveCount
    ValveFlow(h,valve,crom) = LinkFlowTemp(1,ValvesIndex(valve));
    FromNode= ValveFrom(ValvesIndex(valve));
    REFF = convertCharsToStrings(FromNode);
    ToNode = ValveTo(ValvesIndex(valve));
    REFT = convertCharsToStrings(ToNode);
    for node = 1 : NodeCount
        aval = convertCharsToStrings(ned.BinNodeNameID{node});
        if strcmp(REFF, aval)
            ValvePressureFromNode(h,valve,crom)= NodePressureTemp(1,node);
            continue
        end
        if strcmp(REFT, aval)
            ValvePressureToNode(h,valve,crom) = NodePressureTemp(1,node);
            continue
        end

    end
end
end
end
end
```

Fitness function – Pressure

```
function[PressureFitness,CromToDelete] = ObjPressure1(NodePressure,...
    RefPressure,NodeCount,RefMaxPressure,HighPressureZone,...
    RepeatedCromIdx, pop)
PenaltyHighPressure = 0;
PenaltyLowPressure = zeros(pop);
Test = sum(sum(NodePressure,1),2);
for i = 1 : length(Test)
    if Test(i) == 0
        idex(i)= 0;
```

```
    else
        idex(i)= 1;
    end
end
Diference = NodePressure - RefPressure;
for i = 1 : pop
    for t = 1 : size(Diference,2)
        for k  = 1 : size(Diference,1)
            if Diference(k,t,i) < 0
                PenaltyLowPressure(i) = PenaltyLowPressure(i) + 1;
            end
        end
    end
end
PenaltyLowPressure = PenaltyLowPressure - 5;
DiferenceSquare = (Diference.^2)*0.00001;
total = sum(DiferenceSquare,1);
total = sum(total,2);
PressureFitness = (total/NodeCount);
variable = 0;
for t = 1 : pop
    PenaltyHighPressure = 0;
    for time = 1 : 5
        for i = 1 : NodeCount
            if Diference(time,i,t) > RefMaxPressure - RefPressure
                variable = variable + 1;
            end
        end
    end
        variable = variable/(NodeCount*5);
        if variable > HighPressureZone
            PenaltyHighPressure = PenaltyHighPressure + 1;
        end
    if PenaltyHighPressure ~= 0
        PressureFitness(t) = PressureFitness(t)*100;
    end
    if PenaltyLowPressure(t) ~= 0
       PressureFitness(t) = PressureFitness(t)*1000*PenaltyLowPressure(t);
    end
end
    t = 1;
    Delete = 0;
    CromToDelete = [];
    for i = 1 : length(PressureFitness)
        if PressureFitness(i) > 55.90 %Default original pressure.
            PressureFitness(i) = PressureFitness(i).^10000000;
            Delete = 1;
        end
        if Test(i) == 0
            PressureFitness(i) = PressureFitness(i).^2;
            Delete = 1;
        elseif  ismember(i,RepeatedCromIdx)
            PressureFitness(i) = PressureFitness(i).^1000000000000;
            Delete = 1;
        elseif Delete == 1
            CromToDelete(t) = i;
            t = t + 1;
            Delete = 0;
        end
    end

Test = [];
RepeatedCromIdx = [];
Diference = [];
variable = 0;
end
```

Fitness function – Power and Rentability

```
 function[TotalCostPerKW,HPowerSumAverage] = ObjRentability(Valve...
    PressureFromNode,ValvePressureToNode,ValveFlow, pop, ValveCounter,...
    ValveType, CromToDelete)
g = 9.8;
Density = 1000; %kg/m^3%
ErrorListCounter = 0;
load('MATRIXPOWERCURVES.mat');
for crom = 1: pop
    ValveCount = ValveCounter(crom);
   try

        if ValveCount == 0
            for time = 1 : 5
                Delta = 0;
                HPower(time,1,crom)= 0.01 ;% kW
            end

        else
```

```
            for valve = 1 : ValveCount
                for time = 1 : 5
                    for i = 1 : 10
                        if i == 10
                            HPower(time,valve,crom) = 0.01;
                        else
                            Fmin = MatrixPowerCurves(i,(ValveType...
                                (valve,1+time,crom)*2)-1,ValveType...
                                (valve,1,crom));
                            Fmax = MatrixPowerCurves(i+1,(ValveType...
                                (valve,1+time,crom)*2)-1,ValveType(valve,1,crom));
                            Pmin = MatrixPowerCurves(i,(ValveType...
                                (valve,1+time,crom)*2),ValveType(valve,1,crom));
                            Pmax = MatrixPowerCurves(i+1,(ValveType...
                                (valve,1+time,crom)*2),ValveType(valve,1,crom));
                            if ValveFlow(time,valve,crom) > Fmin && ...
                                    ValveFlow(time,valve,crom) < Fmax
                                GPVPower  = Pmax - (((Fmax-ValveFlow...
                                    (time,valve,crom))*(Pmax-Pmin))/(Fmax-Fmin));
                                HPower(time,valve,crom) = GPVPower;
                                break
                            end
                        end
                    end
                end
            end
        end
  catch
       ErrorListCounter =  ErrorListCounter + 1;
        for time = 1 : 5
            Delta = 0;
            HPower(time,1,crom)= 0.01 ;
        end
 end
end


HPowerAverage = sum(HPower,1)/5;
HPowerSumAverage = sum(HPowerAverage,2);
for x = 1 : length(CromToDelete)
    HPowerSumAverage(CromToDelete(x)) =  0.01;
end
% Rentability Fitness Function
    CostTemp = 0;
    ValveCount = ValveCounter(crom);
    Cost = [];
    xtotal = 0;
    try
        for valve = 1 : ValveCount
            x = HPowerAverage(1,valve,crom);
            if x <= 1
                if  x < 0.3
                    CostTemp = 0;
                else
                    CostTemp = (-17512*x.^3)+(38193*x.^2)-28846*x + 9448.3;
                end
            end
            if x > 1
                CostTemp = 1320.7*x.^-0.571;
            end
            Cost(valve) = CostTemp*x;
            xtotal = xtotal + x;
        end
        TotalCostPerKW(crom) = sum(Cost,2)/xtotal;
        if TotalCostPerKW(crom) == 0 || xtotal == 0
            TotalCostPerKW(crom) = 15000;
        end
    catch
        x = HPowerAverage(1,1,crom);
        if x < 0.5
            CostTemp = 10.^10;
        else
            CostTemp = (-17512*x.^3)+(38193*x.^2)-28846*x + 9448.3;
        end
        TotalCostPerKW(crom) = CostTemp;
    end
    for x = 1 : length(CromToDelete)
        TotalCostPerKW(CromToDelete(x)) =  10000;
    end
end
end
```

Non-domination sorting

```
 function [front, obj, crom] = NONDOMINATIONSORTING (TotalCostPerKW,...
    HPowerSumAverage,PressureFitness, pop)
N = pop;
```

```
ind = repmat(struct('np',0, 'sp', []),[1,N]);
[domMat, obj] = calcDominationMatrix(TotalCostPerKW,...
    HPowerSumAverage,PressureFitness, pop);
for p = 1:N-1
    for q = p+1:N
        if(domMat(p, q) == 1)
            ind(q).np = ind(q).np + 1;
            ind(p).sp = [ind(p).sp , q];
        elseif(domMat(p, q) == -1)
            ind(p).np = ind(p).np + 1;
            ind(q).sp = [ind(q).sp , p];
        end
    end
end
front(1).f = [];
for i = 1:N
    if( ind(i).np == 0 )
        crom(i).rank = 1;
        crom(i).Index = i;
        front(1).f = [front(1).f, i];
    end
end
fid = 1;
while( ~isempty(front(fid).f) )
    Q = [];
    for p = front(fid).f
        for q = ind(p).sp
            ind(q).np = ind(q).np -1;
            if( ind(q).np == 0 )
                crom(q).rank = fid+1;
                crom(q).Index = q;
                Q = [Q, q];
            end
        end
    end
    fid = fid + 1;

    front(fid).f = Q;
end
front(fid) = [];
```

Crowding Distance

```
function crom = CrowdingDistance( crom, front, obj)
numObj = 3;
for fid = 1:length(front)
    idx = front(fid).f;
    frontPop = crom(idx);

    numInd = length(idx);
    for i = 1 : numInd
        for t = 1 : 3;
        objTemp(i,t) = obj(idx(i),t);
        end
        objTemp(i,4) = idx(i);
    end
    for m = 1:3
        objTemp = sortrows(objTemp, m);

        colIdx = 4;
        crom( objTemp(1, colIdx) ).distance = Inf;
        crom( objTemp(numInd, colIdx) ).distance = Inf;

        minobj = objTemp(1, m);
        maxobj = objTemp(numInd, m);
        for i = 2:(numInd-1)
            id = objTemp(i, colIdx);
            crom(id).distance = 0;
        end
        for i = 2:(numInd-1)
            id = objTemp(i, colIdx);
            crom(id).distance = crom(id).distance + (objTemp(i+1, m) -...
                objTemp(i-1, m)) / (maxobj - minobj);
       end
    end
    clear objTemp
    minobj = 0;
    maxobj = 0;
    end
```

Selection of the best solutions

```
function [optpop,ParetoFrontIndex, optpopsize] = SelectPop(crom,...
    TotalCostPerKW,HPowerSumAverage,PressureFitness)
optpopsize = length(crom) / 2;
```

```
optpop = crom(1:optpopsize);
rankVector = vertcat(crom.rank);

n = 0;
rank = 1;
idx = find(rankVector == rank);
idxtest = isempty(idx);
numInd = length(idx);
if rank ==1 && idxtest == 1
    ParetoFrontIndex( 1 : 1 ) = 0;
end
if numInd >= optpopsize


    distance = vertcat(crom(idx).distance);
    distance = [distance, idx];
    distance = flipud( sortrows( distance, 1) );
    idxSelect  = distance( 1:optpopsize-n, 2);
    optpop(n+1 : optpopsize) = crom(idxSelect);
    ParetoFrontIndex( n+1 : optpopsize ) = crom(idxSelect);

else

while( n + numInd <= optpopsize )
    optpop( n+1 : n+numInd ) = crom( idx );

    if rank == 1
        ParetoFrontIndex( n+1 : n+numInd ) = crom( idx );
    end
        n = n + numInd;
        rank = rank + 1;
        idx = find(rankVector == rank);
        numInd = length(idx);
end
if( n < optpopsize )

    distance = vertcat(crom(idx).distance);
    distance = [distance, idx];
    distance = flipud( sortrows( distance, 1) );
    idxSelect  = distance( 1:optpopsize-n, 2);
    optpop(n+1 : optpopsize) = crom(idxSelect);
end

end
end

function newpop = selectOpt(optpop)
popsize = length(optpop);
pool = zeros(1, popsize);
count = 1;
for x = 1 : popsize
    d = optpop(x).rank
    if d == 1
        prob = 3;
    elseif d == 2
        prob = 2;
    elseif d > 2
        prob = 1;
    end
    for y = 1 : prob
        ProbPool(count) = x;
        count = count + 1;
    end
end
ProbPoolSize = count - 1;
randnum = randi((count-1), [1, 2 * popsize]);

j = 1;
for i = 1:2:(2*popsize)
    p1 = randnum(i);
    p2 = randnum(i+1);
    result = crowdingComp( optpop(ProbPool(p1)), optpop(ProbPool(p2)));
    if(result == 1)
        pool(j) = (ProbPool(p1));
    else
        pool(j) = (ProbPool(p2));
    end

    j = j + 1;
end
newpop = optpop(pool);
end


function [PopulationTemp, PopulationMod] = JoinPopulations(newpop,...
    optpop, Population)

for i = 1 : length(newpop)
```

```
        PopulationTemp(:,:,i) = Population(:,:,optpop(i).Index);
        b = length(optpop) + i;
        PopulationTemp(:,:,b) = Population(:,:,newpop(i).Index);
        PopulationMod(:,:,i) = Population(:,:,newpop(i).Index);
end
end
```

Crossover Operator

```
function PopulationMod = Crossover(PopulationMod, RCross)
c = size(PopulationMod,3);
for i = 1 : c
CrossVector(i) = rand();
end
t = 1;
CrossParentVector = [];
for i = 1 : c
    rest =  rem(size(CrossParentVector,2),2);
    if i ~= (c)
        if CrossVector(i) < RCross
            CrossParentVector(t) = i;
            t = t + 1;
        end
    end
    if i == (c)
        if rest == 0
            break
        end
        if isempty(CrossParentVector)
            break
        else
            CrossParentVector(t) = i;
        end
    end
end

for i = 1 : 2 : length(CrossParentVector)
    Parent1 = CrossParentVector(i);
    Parent2 = CrossParentVector(i+1);
    finish  = round(rand()*size(PopulationMod,1));
    start = round(rand()*(size(PopulationMod,1)-finish));
    Nvar  = finish - start;
    for xcoord = 1 : Nvar
    tocopy1(1,:) = PopulationMod(start + xcoord,:, Parent1);
    tocopy2(1,:) = PopulationMod(start + xcoord,:, Parent2);
    PopulationMod(start + xcoord,: , Parent1) = tocopy2(1,:);
    PopulationMod(start + xcoord,: , Parent2) = tocopy1(1,:);

    end
end
end
```

Mutation Operator

```
function PopulationMod = Mutation(PopulationMod, RMut, NPATsize, RMutPosition)
c = size(PopulationMod,1);
e = size(PopulationMod,2);
d = size(PopulationMod,3);
Mutation1VectorProb = zeros(c,d);
for z  = 1 : d
    for i = 1 : c
        for x = 1 : e
            Mutation1VectorProb(i,x) = rand();
        end
    end
    for i = 1 : c
        if   Mutation1VectorProb(i,1) < RMutPosition
            if PopulationMod(i,1,z) == 1
                PopulationMod(i,1,z) = 0;
            else
                PopulationMod(i,1,z) = 1;
            end
        end
        if Mutation1VectorProb(i,2) < RMutPosition
            PopulationMod(i,2,z) = round(rand()*NPATsize);
        end
        for x = 1 : e
            if x == 1 || x == 2
            else
                Mutation1VectorProb(i,x) < RMut
                PopulationMod(i,x,z) = ceil(rand()*8);
            end
        end
    end
end
clear  Mutation1Vector;
```

```
end


function[Population,RepeatedCromIdx] = RepeatedCrom(Population,...
    optpop, PopulationMod)

b = length(optpop);
d = size(PopulationMod,3);
c = 1;
RepeatedCromIdx = [];

for i = 1 : d

    for u = 1+i : d

        if Population(:,:,i) == Population(:,:,u)
            t = ismember(u,RepeatedCromIdx);
            if t == 0
            RepeatedCromIdx(c) = u;
            c = c + 1;
        end
    end
end
end
```