1-1-2016

# Enhanced Pump Schedule Optimization For Large Water Distribution Networks To Maximize Environmental And Economic Benefits

Seyed Mohsen Sadatiyan Abkenar
*Wayne State University,*

**ENHANCED PUMP SCHEDULE OPTIMIZATION FOR LARGE WATER DISTRIBUTION NETWORKS TO MAXIMIZE ENVIRONMENTAL AND ECONOMIC BENEFITS**

by

**S. MOHSEN SADATIYAN A.**

**DISSERTATION**

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

**DOCTOR OF PHILOSOPHY**

2016

MAJOR: CIVIL ENGINEERING

Approved By:

Carol J. Miller 3-11-2016

Advisor Date

3-11-2016

3-11-2016

3-11-2016

3-11-2016

# DEDICATION

I dedicate this work to my parents and lovely wife who supported me throughout the process. I appreciate the unstinting support of my devoted parents who never let me down in all steps of my life. I also feel beholden to my wife who always encouraged me to work harder by reducing my self-satisfaction ;)

## ACKNOWLEDGEMENTS

I was so lucky to have an advisor like Dr. Carol Miller and doing this research was not possible without her technical, financial and mental supports. It was an honor for me to conduct this research under her supervision and guidance. I would also like to show my gratitude to all my instructors and thesis committee members, Dr. Shawn McElmurry, Dr. David Szlag, Dr. Caishang Wang and Dr. Yongli Zhang, who increased the quality of this work by their advice. Finally, I would like to thank Dr. Donald Chase and Samuel Dustin Stanley from the development team of the first version of PEPSO, members of LEEM team, Barry LaRoy from the water department of City of Monroe and the Yorkshire Water for sharing hydraulic model of their water networks.

# TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF FIGURES

# LIST OF ACRONYMS AND ABBREVIATIONS

AC: Ant Colony

ANN: Artificial Neural Network

CPN: Commercial Pricing Node

DMA: Demand Manage Area

DS: Darwin Scheduler

EI: Energy Intensity

EPS: Extended Period Simulation

FANN: Fast Artificial Neural Network (C library)

GA: Genetic Algorithm

GHG: Green House Gas

GUI: Graphical User Interface

JPP: Junction Pressure Penalty

LEEM: Locational Emission Estimation Methodology

LMP: Locational Marginal Price

NPW: Negative Pressure Warning (system has negative pressure warning of EPANET)

NSGA II: Non-dominated Sorting Genetic Algorithm II

PDW: Pump Disconnection Warning (system disconnected because of pump warning of EPANET)

PEPSO: Pollution Emission Pump Station Optimization

PFW: Pump Flow Warning (pump cannot deliver flow warning of EPANET)

PHW: Pump Head Warning (pump cannot deliver head warning of EPANET)

PI: Proportional Importance

PPP: Pipe Pressure Penalty

RRS: Relative Rotational Speed

SEM: Standard Error of Mean

TLP: Tank Level Penalty

TTSUI: Total Time Step Undesirability Index

UI: Undesirability Index

VB.NET: Visual Basic.NET

WDS: Water Distribution System

# CHAPTER 1 LITERATURE REVIEW

## 1.1. Introduction

In this dissertation, a multiobjective optimization tool is introduced that is developed for reducing electricity cost and pollution emission (associated with energy consumption) of pump stations of water distribution systems (WDS).

In the first chapter, the background of this field of study is reviewed and necessity of doing this type of research is justified. The general concept of optimization of pump schedules, hydraulic modeling techniques, optimization methods, objectives of optimization, active research groups, and benchmark test cases are main subjects that have been covered in this chapter.

In the first section of the second chapter, the problem that is addressed in this study is defined, and the hypothesis is stated clearly. After that, the developed optimization tool is explained in detail and the test cases and scenarios that are considered with the model are described. All technical details about interface and internal function and procedures of the developed optimization tool can be found in this chapter

The third chapter presents the test results. Raw results are processed and analyzed. The statistical indices and quantitative measures used to describe the test results are presented. In the first set of experiments, different features of the developed optimization tools are tested on two WDSs. In the second set, the optimization tool is compared with a well-known commercially available software package.

Finally, in the last chapter, conclusions of the thesis are presented. In addition, a comprehensive list of opportunities for further investigation and future research is presented. A glossary appendix is included at the end of this document that provides definitions for the acronyms, abbreviations and technical terms that are used in this document. The reader is highly recommended to refer to that section when facing an unfamiliar or unclear phrase or acronym in the text.

## 1.2. Background and Necessity of Optimization of Pump Operation

The pump is a mechanical device for pressurizing fluids. Various types of these devices have been used in almost all fields of human activities that deal with fluids, both liquid and gaseous. In most cases, the required flow rate and pressure of the fluid may include dynamic features, requiring the design to satisfy a range of conditions. The specified operation schedule of the pump must address these dynamic requirements.

Operating schedule of a pumping system defines ON and OFF status of fixed speed pumps (FSP) and, the rotational speed of VSPs. These pump operation schedules or in short pump schedules are an important component of the operational plan for water and wastewater pump station, oil and gas facilities, most of the industrial process that deals with fluids, the air conditioner of buildings, etc. Except some rare cases that water source always has a higher elevation than all consumption points, almost all water distribution systems (WDS) have at least one pump. In many cases, WDSs have more than one pump in multiple pump stations and pumps might become scattered throughout the whole area of the WDS. In some large WDSs (e.g. Detroit Metropolitan WDS) the number of pumps exceeds several hundred. WDS designers commonly use parallel and series pump systems and variable speed pumps (VSP) to cover a broad range of required flow and pressure of different systems. All of these factors make a WDS so complex and dynamic that numerous combination of pump operation plans can satisfy required pressure and flow of the system. Each of possible pump schedules may need different power demand and electricity usage

About 4% of electricity usage in the US is attributed to the supply, conveyance and treatment of water and wastewater at a cost of approximately $4 billion per year (Giacomello, Kapelan et al. 2013). Moreover, due to increasing in urban and industrial water demand and a decrease in access to high-quality water resources, it is predicted that the energy consumption of this sector will increase more than 50% by 2050 (Giacomello, Kapelan et al. 2013). According to the US Department of Energy, approximately 75% of the operating costs of municipal water supply, treatment and distribution facilities are attributed to electricity demand (DOE 2006). Abiodun reported that about 700 million Euros is being expended annually on energy costs of pumping stations in the UK (Abiodun and Ismail 2013).

The high energy demand and increasing trend of the demand and cost of energy is a motivation for water system operators to increase the efficiency of energy usage in this sector. Pumps are the largest energy consumers in water supply, treatment and distribution systems. For instance, in China, the electrical cost of pump operation is about 30% to 50% of the total operational cost of the WDSs (Abiodun and Ismail 2013). The amount of energy used in pump stations depends on the efficiency of pumps and required flow and pressure. Most of the time, . Due to these differences in pump efficiencies (even within a single pump station), energy usage of multiple pumps will not be the same, even if they fulfill the similarly required pressure and flow rate. Moreover, the combination of series and parallel pumps may lead to numerous pumping strategies that can satisfy the required water flow rate and pressure. It was mentioned that each pumping strategy has different energy and power requirements. However, the optimum pump schedule is associated with the lowest energy and power demand while fulfilling all pressure and flow requirements of the WDS. Uncertainties and control limitations increase the tendency of human operators to maintain water pressure higher than the minimum required pressure, which increases energy usage, water leakage and consequently water and energy waste. As noted by several researchers, optimizing pump operation has a considerable effect on water industries, which can offer up to 10% reduction in the annual expenditure of energy and other related costs (Jamieson, Shamir et al. 2007); (Abiodun and Ismail 2013). Based on the water-energy nexus report of US Department of Energy, in the year 2011, 39.2 billion kWh energy used for pumping and aeration of publically available water in the US (DOE 2014). About 80% of this energy is consumed in pumping demands (Copeland 2014). Therefore, the total amount of energy consumption attributed to the pumping of publically available water in the US in the year 2011 was 31.36 billion kWh. If we assume that all pump operators use some optimization techniques and reduce this energy consumption by 10%, 3.14 billion kWh energy will be saved annually. Considering 0.10 ($/kWh) as the average electricity cost in the US (EIA 2016), this optimization can save about 314 million dollars annually. Based on eGRID 2012 data, the average $CO_2$ equivalent, $SO_2$, $NO_x$ and Hg emission rate per kWh of generated energy in the US is 517.98, 0.86, 0.43 and 0.000006 (gr/kWh) respectively (EPA 2015); (Marc Houyoux 2011). Therefore, in each year, it is possible to prevent the release of 1.63 billion kilograms of equivalent $CO_2$, 2.69 million kilograms of $SO_2$, 1.35 million kilogram $NO_x$ and 18.87 kilograms of Hg to the atmosphere by optimizing energy consumption of water pump station in the US. In this calculation, we have solely considered energy optimization in pumps of the water sector. If we calculate the effect of the similar type of optimization in wastewater industry and consider increasing trend of energy consumption in public and private sectors, the amount of the estimated saving will be significantly greater than the above-mentioned value.

There are multiple solutions to a basic WDS optimization problem with constant demand and specified pressure constraints. However, in reality, the flow and pressure demands of a WDS are not constant. So at different times of a day, different days of a week and even different months and seasons of a year, the required pressure and flow rate at points in a network may be vastly different. Furthermore, some physical changes in the network topology may cause changes in required pressure or even flow rate. For instance, adding or removing some pipes, aging, leakage, or breaking of pipes are typical occurrences in the WDS that may change the required flow or pressure of the system. In addition to these physical and topological changes of a WDS, there are some modifications like a change in quantity or types of consumers that may also change the required flow rate and pressure of the system. All these changes of demands require the use of time-variant operation schedules. Based on a possible combination of pumps, it might not be possible to fulfill all pressure requirements of networks at all junctions at the same time. Therefore, we may face some undesired high or low pressure at multiple demand junctions of WDS. Also, in some WDSs that are equipped with pressure reduction or break valves, there is a possibility to operate pumps in a way that results in the buildup of unrequired high pressure behind the pressure or flow control equipment that means excess energy usage and energy waste. These cases show that based on the condition of a system, optimum pump schedule may change. Thus, there is not one optimum pump schedule that answers all needs of a WDS at all times.

The above-mentioned scenarios make it clear that the amount of energy used to meet the demands of a WDS may change over time, and we need to change the operational schedule of pumps to meet unsteady demands. However, there are some cases that even with the constant demand condition and constant energy usage, the energy cost may considerably change. Many power utilities provide energy based on a time-dependent tariff. Based on a time of use a type of tariff, energy price changes over hours of a day (in some cases even months of a year). Therefore, there may be several different price points associated with a specific energy consumption amount, if alternative patterns of consumption are considered. In this case, using elevated storage to deliver water at those hours that energy is expensive, and pumping water to elevated storage during low energy pricing times, will lead to reduced costs for system operation. Moreover, some electricity providers include cost penalties for WDSs that exceed threshold power consumption values. For instance, consider two pump schedules that require equivalent energy. The first pump schedule operates all pumps together for a limited period of the day, while the second schedule uses selected pumps throughout the whole day. Due to the higher power output of the first schedule, the first pump schedule is likely to result in larger operation cost due to the increased power demand charge. Careful planning for operating pumps may lead to a decrease in energy usage, power demand, energy usage cost and power demand cost.

While there have been many previous investigations into the direct (or, *internal*) cost of energy usage in pump stations of WDSs, the *external costs* have received much less attention. An external cost arises when the social or economic activities of one group of persons have an impact on another group and when that impact is not fully accounted, or compensated for, by the first group. These *external costs* are mostly associated with the environmental footprint of energy generation. The external cost of energy usage mostly depends on the source of the fuel and the means of combustion. Various sources of energy emit different types and quantity of pollutants. The environmental effect of the pollutant emission of a fixed amount of energy generated by different methods can also be different. These various effects may lead to a wide range of external cost of energy usage on the environment, society, etc. Electricity distribution systems distribute the energy that has been generated by a combination of many differing energy sources. The mix of energy sources is variable over space and time. So, there are some times that a combination of multiple clean energy generators leads to the generation of less polluting energy with lower external cost. Therefore, the external costs associated with energy consumption are highly dependent on the spatial and temporal features of the energy consumption. Optimum operation of pumps may lead to energy usage at those times that the environmental footprint of energy generation is lower, resulting in reduced external costs of energy consumption.

A pump operation schedule can be optimized to satisfy various goals, such as the amount and cost of energy used, amount and cost of power demand, amount of pollution emission and external cost of energy usage. These goals are not always aligned. For instance, reductions in energy usage may lead to (1) more intense (shorter duration) of energy consumption, (2) consumption during higher energy fee periods, and/or (3) consumption from “dirtier” generators; resulting higher power demand costs, higher energy costs or higher pollution emission and external costs, respectively.

The advantages of an optimized pump operation are evident. However, the means to achieve that optimized schedule are less obvious. At first, let’s investigate the *solution space* of a pump scheduling problem. The *solution space* of this problem is a collection of all possible combination of the operational status of pumps of a system. To understand better the potential size of the solution space, we first consider a very simple pumping system that has just one fixed or constant speed pump (FSP). We also assume that we want to operate the pump for a one day period (24 hours), and that the operational state of the pump (ON/OFF) can be changed only once per hour. The number of combinations of all possible operational states for the first hour is 2 (ON or OFF) and for a two-hour period is $2\times2 = 2^2$ or four combinations. Consequently, the number of possible operational plans of this pump during 24 hours is $2^{24}$. If we have two pumps in the system, the size of possible operational plans will be $2^{24}\times2^{24}=2^{48}$. To understand better the magnitude of this solution space, we can compare to the number of all atoms in the observable universe. This number of atoms (Wikipedia-contributors 2003) is less than the number of possible pump schedules of a pump system with 12 pumps. Similarly, If we imagine that there is a way that we can use a supercomputer with highest theoretical possible computational power ($6\times10^{33}$ operations per second per joule of energy (Wikipedia-contributors 2005) that use the entire available energy of the largest power generator in the world (Three Gorges Dam with total electric generating capacity of 22,500 MW (Wikipedia-contributors 2002), determination of the global optimum pump schedule for a small system with just 6 pumps and find the global optimum pump schedule with 100% certainty would require hundreds of thousands of years.

As explained with the examples above, to solve these types of problems using a deterministic algorithm (Wikipedia-contributors 2004), we need exponential time with respect to variables of the problem (e.g. number of pumps). Therefore, based on computational complexity theory, we can classify it as an EXPTIME problem or exponential time problem (Wikipedia-contributors 2013). In the same way, Yates et al. and Marchi et al. classify the problem of finding the optimum design of WDS as the non-deterministic polynomial-time–hard (NP-H) problem (Yates et al. 1984; Marchi et al. 2014). To summarize, we can say that these types of problems cannot be solved completely by any algorithm in polynomial time. The size of the solution space for even a small pump optimization problem is so large that is not possible to find the optimum pump schedule by evaluating all possible solutions.

In addition to the size of solution space, there is another issue that needs special attention. The relation between change in the status of pumps and change in power or energy demand of system is not linear. The head-flow rate and efficiency-flow rate curves of pumps are usually non-linear. Also, operating parallel and series pumps in a system have a reciprocal effect on the operation of pumps, which increase non-linearity of head-flow rate and efficiency-flow rate relations. This means that turning on a pump that is connected to other pumps in a parallel or series configuration may change the suction or discharge pressure of other pumps, thereby moving the operational point of other pumps on the head-flow rate and efficiency-flow rate curves. In this case, the status of a pump may not change directly, but changing the status of other pumps in the system can modify discharge or efficiency of the pump indirectly. There are many other non-linear aspects to the WDS optimization problem, such as the relation between pump status and pollutant emission or cost of consumed energy. Therefore, it is concluded that the pump operation optimization problem is highly non-linear. In addition to non-linearity, the pump operation optimization problem is a non-convex problem. Non-convexity makes finding the global optimum solution extremely hard. For this type of problems there might be multiple local optimums, and finding the global optimum (if it exists) is not guaranteed.

### 1.3. Optimization Objectives

In Section 1.1 it was explained that optimization of a pump schedule may have various benefits. One of the most important objectives of almost all optimization schemes is cost reduction. Specifically, in the WDS case, the objectives of reducing the capital cost of constructing WDS and reducing the energy usage cost of pumps have been investigated by numerous researchers; Tang, Zheng et al. 2014). Most of the initial optimization efforts for WDS design and operation combined the cost of construction and operation into a single objective. If we assume that the energy price is a constant rate at different times, it is evident that by reducing the amount of energy usage, the associated energy cost will also be reduced. For that case, the amount or cost of energy usage are interchangeable from an optimization perspective. However, for cases that include variable energy costs (time variant energy pricing), the more appropriate optimization objective is cumulative cost of energy consumption. As explained earlier, such variable pricing applications result in multi-valued billing for identical consumption totals. So in practice, when most of the electricity tariffs depend on usage time, it is more meaningful to consider the reduction of energy cost (e.g. $/kWh) as an objective function instead of using the energy consumption amount (e.g. kWh). Accordingly, in most cases researchers considered the change of energy consumption charge during a one day (24 hours) simulation period. Wang et al. considered the hour between 8:00 to 17:00 as peak hours of electricity tariff. So a day was divided to 00:00 to 8:00, 8:00 to 17:00 and 17:00 to 24:00 (Wang, Chen et al. 2013). Baran et al. also used time-dependent electricity tariff that was defined based on-peak (17:00 to 23:00) and off-peak (00:00 to 17:00 and 23:00 to 24:00) hours (Barán, von Lücken et al. 2005). However, Shamir and Salomon used a more complicated electricity tariff. They used the real and complex electricity tariff of Haifa city in Israel which includes three time periods, representing high, medium, and low energy costs. The tariff is different for the weekend and holidays and the various seasons of the year (Shamir and Salomons 2008).

Considering the real electricity tariffs, in many cases, there is a *power demand charge* ($/kW) in addition to the *energy consumption charge* ($/kWh). The cost of energy consumption is added to the cost of maximum required power to determine the total electricity cost of the system. Working multiple pumps at the same time may cause an increase of required power for pumping unit of a WDS. This may increase the total electricity cost of the system. There are some examples that researchers pay attention to the *power demand charge* (Fracasso, Barnes et al. 2014). Fracasso and Barnes included the amount of max power demand (kW) as an objective of the optimization process.

Martinez et al. optimized operation of Valencia WDS and they reported that about 17% reduction of operation cost is possible during a one-year optimization period. Also, it was evaluated that by this amount of saving after 16 months the cost of equipping

Valencia network by SCADA system will be returned. Most of this saving occurred during the high consumption months (Martínez, Hernández et al. 2007).

About four decades ago, when researchers started to think about optimization of WDSs, most of them focused on construction cost (reducing the cost of piping) and operation cost (minimizing the cost of energy usage and power demand of pump station). However, after a while, other objectives like increasing reliability and water quality or decreasing environmental footprint were included in the optimization process. In the last decade, the attention toward the environmental effect of energy usage and sustainability of WDSs increased due to increase in public and scientific awareness of climate change and effect of pollutant emissions from power generation (Wu, Maier et al. 2013). Wang et al. tried to reduce the environmental effect of WDS operation, in term of preventing land subsidence caused by groundwater withdrawal (Wang, Chen et al. 2013). They reported that their suggested algorithm could find an optimum solution with 500 generations using good initial guesses. They claimed that this algorithm converged to an optimum solution very fast (Wang, Chen et al. 2013). However, considering the size of their problem, it is not completely clear that if this algorithm can outperform other algorithms that have been used in other researches (especially for large WDSs). Wu et al. did a comprehensive research on multiobjective optimization of WDS design. Minimizing total life cycle GHG emissions was one of the objectives of their optimization method (Wu, Maier et al. 2013). Including the environmental effect of WDS in optimization process is a new approach and most of the related researches consider only the reduction of GHG in design optimization problems rather than reducing pollutant emission in operation optimization problems. Therefore, the environmental effect of energy usage and pump operation has not been adequately investigated. It was not possible for the author to find any article that includes the environmental effect of WDS design or operation on the optimization process before 2010. In fact, it appears the study Wu et al. were one of the first studies that included greenhouse gas (GHG) emissions as an objective of the WDS design optimization (Wu, Maier et al. 2010). Recently, Stokes et al. suggested a framework for the modeling and optimization of GHG emission associated with energy usage and pump operation of WDSs (Stokes, Simpson et al. 2012). In most of these efforts emission rate of energy usage was considered as a constant value and was linearly related to the amount of consumed energy. However, it is known that most of the time, the source of electrical energy is a mix of various types of power generators. As this combination of generators may change in time, emitted amount of GHG or other pollutants per unit of energy may change. So, consuming the same amount of energy at two different times might result in different effects on the environment. In reality, as water demand of WDS and electricity price may change during an optimization process, emission rate of energy usage may also change. Researchers at Wayne State University recently completed a research project to optimize pump operation of WDS considering the real-time effect of energy usage on pollutant emission. In that project, the LEEM methodology was developed to calculate the amount of pollutant emission associated with energy generation at different points in space and time. LEEM is an acronym for Locational Emissions Estimation Methodology. LEEM is implemented in several products and is offered as streaming data for the industry. Users can connect to its online server and obtain local information of marginal pollutant emission of electricity generation at a different time (lb/kWh). The marginal emission at any location and time is the expected emissions due a unit increase in demand at that location and time that is produced by the marginal generator (Rogers, Wang et al. 2013).

The objective of reducing energy consumption can be expressed in monetary units when represented as energy consumption cost. This allows the energy objective to be added directly to other objectives (such as power demand cost) that are expressed in monetary units. Similarly, it is possible to convert the objective of reducing the environmental footprint of energy usage into a reduction of *external cost* of energy usage. A power station that generates emissions of $SO_2$, $NO_x$, particulates, etc. causing damage to building materials, biodiversity or human health, imposes an external cost. This is because the impact on the owners of the buildings, crops or on those who suffer damage to their health is not taken into account by the generator of the electricity when deciding on the activities causing the damage. Therefore, the environmental costs are ‘‘external’’, although they are real costs to these members of society, the owner of the power station is not taking them into account when making decisions (Streimikiene, Roos et al. 2009). So the idea of using the *external cost* of electricity come from this point that we include the electricity usage cost that is directly related to electricity generation cost as one of our optimization objectives, but there are some hidden and external costs of electricity generation that usually have not been accounted in electricity tariff. Most of these costs are related to environmental effects of air pollution associated with energy production activities. So by including the *external cost* of electricity in the objective function of WDS optimization, we can simply add the environmental effect of energy usage with energy usage cost. However, it should be noted that *external costs* of air pollution vary according to a variety of environmental factors, including overall levels of pollution, geographic location of emission sources, the height of emission source, local and regional population density, meteorology, and so on (Holland and Watkiss 2002).

Wu et al. attempted to include the effect of variable emission rate and electricity tariff on their WDS design optimization efforts. They assumed three scenarios for electricity tariff changes and three scenarios for a change of emission factor (kg$CO_2$e/kWh) in 100 years operation period of a simple transmission line that pumps water from source to three reservoirs with constant and similar head and demands. Three optimization scenarios were completed with variable electricity tariffs, and three other optimizations were done with variable emission factors. These scenarios and changes in electricity tariff and emission rate were created based on probable Australian government policy in future. Results indicated that variation in electricity tariffs have a significant effect on the total cost, but little effect on the total GHG emissions. Also, it was concluded that higher electricity tariffs can remove networks with higher emissions from the Pareto-optimal front, which potentially leads to a final WDS with lower GHG emissions. In contrast, emission factors have no direct effect on the total cost of WDS operation (Wu, Simpson et al. 2012).

Besides the above-mentioned objectives, some constraints seek to direct the algorithm to solutions that satisfy operational requirements of the WDS. For instance, as frequent pump switching (OFF/ON) can cause increased maintenance cost and may damage pumps faster (Wang, Chang et al. 2009), some researchers place a limit on the maximum number of pump switches. Similarly, water pressures at system junctions or water flow rate in pipes can be constrained. Constraints can be handled explicitly or can be converted to objective and handled implicitly during the optimization process. One of the common methods of converting a constraint to an objective is using penalty formula. By this approach, violation from a constraint can be converted to a penalty value and be added to other terms of the objective function. For instance, in a minimization problem, a penalty can be a positive number that has a direct relation to the amount of violation from a constraint. In this case, a solution with more violation from desired limits will have higher penalty value. Adding the penalty value to the amount of objective function increases its value, causing that solution to appear less desirable as an optimum solution of a minimization problem.

In most traditional hydraulic simulators (e.g. EPANET) (Rossman 2000), the user inputs required flow rates at junctions and software solve equations to calculates the pressure at those junctions. In this case, the significant negative pressure at demand junctions shows that required flow of the junction has not been satisfied. So it is common in WDS optimization to constrain the pressure range at junctions and use it as a measure to evaluate the quality and feasibility of the solution. Pressure penalty can be calculated as a function of pressure violation at each junction (deviation above or below a specified allowed maximum or minimum, respectively). Other constraints, such as minimum velocity or the highest number of pump switches can be treated like pressure constraint and included implicitly in optimization process by using the penalty formulation. Zecchin et al. used pressure penalty to add pressure constraint to the objective function of ant colony (AC) algorithms that they used for WDS design optimization (Zecchin, Maier et al. 2007). Wang et al. also suggested a method to calculate the number of pump switches. They just considered the water level in the tank as a constraint and didn't take into account pressure of different junctions. (Wang, Chen et al. 2013). Lopez-Ibanez also investigated the effect of constraint on the maximum number of pump switches and constraints on minimum time interval in time-controlled trigger representation. He found that lower limit of the maximum number of pump switches that does not hinder the search for an optimum solution is related to characteristics of the network. In some cases, three switches can lead to the right results. While assigning a large number as the maximum number of pump switches increases the flexibility of the pump schedule assignment, but as it causes exponentially larger solution space, the process of finding an optimum solution gets much harder (Lopez-Ibanez 2009).

In addition to this implicit methodology, there are some explicit methods for handling the constraints. For instance, Siew and Tanyimboh adopted a pressure dependent analysis approach to simulate both normal and pressure deficient networks (Siew and Tanyimboh 2010). They used their method on some test cases and got good results (Siew and Tanyimboh 2011, Siew, Tanyimboh et al. 2013). Baran et al. also used an explicit heuristic out of main optimization algorithm to evaluate the feasibility of solutions and fulfill technical and hydraulic constraints. Although, even they included the maximum number of pump switches inside their optimization algorithm (Barán, von Lücken et al. 2005).

Most researchers have considered the WDS optimization problem as a single objective problem. Most of them solely focused on the economic side of the problem and considered the cost minimization as the optimization objective. However, some researchers, including Wang et al. used multiobjective methods for optimization of the WDS operation (Wang, Chen et al. 2013). References to the multiobjective optimization of WDSs accounting for network reliability can be traced back to the 1980s, when Walski et al. used the WADISO program to solve WDS pipe sizing problems, considering both cost and the minimum pressure of the network (Walski and Hartell 2012). However, during past decade, by improving multiobjective optimization algorithms, the usage of these methods in WDS optimization increased considerably. In particular, due to the increasing interest of researchers on the environmental effect of WDSs, the use of multiobjective methods for simultaneous optimization of both cost and environmental outcomes has increased.

Optimizing the WDS based on more than one parameter does not necessarily require the use of multiobjective methods. For instance, two objectives can be added together to form an index; then the index can be optimized by using single objective methods. This approach was used for the previously described application involving the monetization of energy usage and power demand. Similarly, adding the cost of pollution emission to the energy and power cost helps us to optimize these three objectives by using a single objective value.

There is another method that can be used for converting a multiobjective problem into a single objective problem. In this approach, the normalized value of objectives can be added together to form a unitless aggregate index. Then the unitless index can be minimized or maximized. In this case, there is no need to convert all values to cost and then add them together. So this method can be used for objectives that cannot be monetized easily. The final amount of objective function that is calculated by this method is not cost and does not have any specific unit. It is just value for evaluation and comparing solutions. Normalizing each value can be done by dividing it by the maximum possible amount of it. It helps to use normalization to convert values of all objectives to a number between 0 and 1. It is important to note that finding the maximum possible amount of an objective is not always easy. For instance, there is not an easy way to calculate maximum possible pollution emission or maximum energy consumption of a WDS. So in some cases, this normalization step can be omitted. Normalized or raw values of each objective can be multiplied by a weighting factor and then added to other terms of the objective function. These weighting factors show the relative importance of various terms of the objective function. For instance, if the weighting factor of normalized pollution emission is two times more than the weighting factor of normalized energy consumption, the effect of pollutant emission on the selection of an optimum solution is two times more than the effect of energy consumption. If we omit the normalizing step and use the raw amount of each objective to calculate the fitness of one solution, possible range and scale of values should be considered in selecting the proper weighting factors. If a possible range of values for objective one is thousands of times smaller than the scale of values for the second objective, the weighting factor of objective one should be thousands of time greater than the second objective to balance the effect of both objectives on the final amount of calculated fitness. It can be seen that even in the previous method that we suggested using cost to unify value of all objectives, the *energy consumption charge* ($/kWh), *power demand charge* ($/kW) and *emission factor* ($/physical unit of pollutant) act as weighting factors. In Bi and Dandy's research on WDS design optimization based on water quality, objective function value was the summation of all pipe cost and net present value of chlorine cost. In this study, the minimum pressure and chlorine concentration were constraints of the problem (Bi and Dandy 2013).

Pollutant Emission Pump Station Optimization (PEPSO) is a software tool developed by the water research team of Wayne State University for optimizing pump schedule of WDS (Miller, Rogers et al. 2014). The initial version of PEPSO uses weighting factors to unify the effect of all different objectives of the optimization process into a single objective function. Wu and Behandish calculated the amount of the objective function by the total weighted cost of energy and amount of three penalties (Periodic water level, Emergency lower-bound and Prevention of overtopping constraints) about water level of tanks (Wu and Behandish 2012). Abiodun and Ismail did a bi-objective optimization that aimed to reduce electricity cost (using time of use electricity tariff) and reducing maintenance problems (reducing the frequency of switching pumps; Abiodun and Ismail 2013). They used both normalizing and weighting factors methods to combine two terms of the objective function. For this purpose, amounts of the objectives were normalized by dividing on the differences between max and min values.

It was explained that a multiobjective problem can be converted into a single objective problem. However, it also was mentioned that calculating a single-objective value by using values of different objectives is not always easy and straight forward. For instance, for calculating the total cost of a solution we need to convert the effect of pollutant emission to the external cost of energy usage and add it to the cost of electricity usage. However, calculating the external cost of energy is not easy, and it depends on many parameters beyond the limits of the optimization problem (e.g. type and the location of the power generator, location, and time of energy consumption, etc.). So it is not easy and always the best method to convert a multiobjective problem to a single objective problem. An alternative way is using multiobjective optimization methods to optimize the multiobjective problem directly and find the *Pareto front*, instead of one optimum solution. *Pareto front* is a set of Pareto optimal solutions that are better than other solutions with respect to all objectives but cannot dominate each other with respect to all different objectives. Figure 1 shows the *Pareto front* and the dominancy concept. It is assumed that the illustrated plane is the solution space of a minimization problem, and each junction represents a solution. Axes (f1 and f2) shows the value of each of the two objectives. We can see that those light color solutions (e.g. point C) do not have any advantage on dark color solutions (e.g. Point A or B) with respect to both objective values. However, comparing darker solution together, we see that there is not any dark point that both of its objective values is less than both objective values of another dark colored solution. Therefore, darker solutions dominated lighter solutions, and none of darker solutions dominated another darker solution. These non-dominated points create a *Pareto frontier*. The final answer of multiobjective optimization methods is a *Pareto frontier* (not one single optimum solution). However, in practice, we need one solution to implement in the real operation plan. So after using a multiobjective optimization method and finding the Pareto frontier, an expert, based on specific needs, can select the proper solution from the group of non-dominated solutions and use that as an optimum practical solution. The selection of one single solution among the solutions of a *Pareto frontier* can be facilitated by using some general rule that shows the importance of each objective with respect to the other objectives and acts as a weighting factor. In this case, one part of the *Pareto frontier* that has some solutions with better values of the objective with a larger weighting factor (or higher level of importance) can be investigated for selecting the best solution.

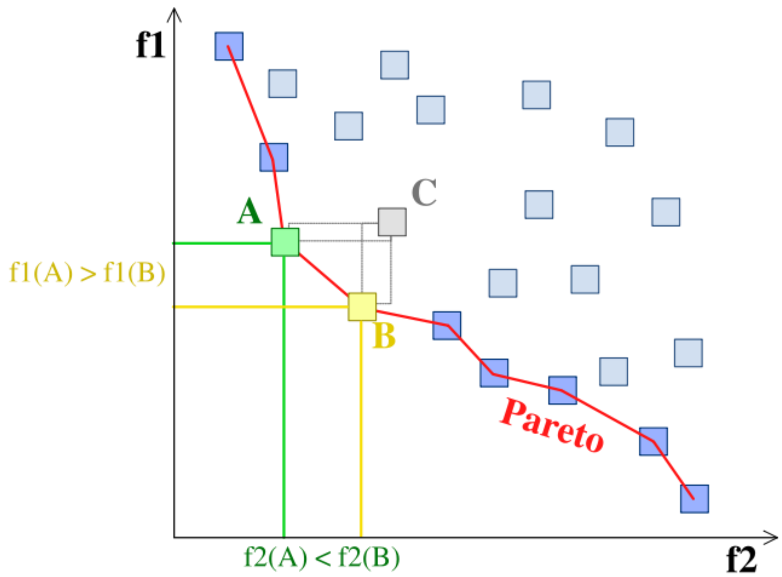



Figure 1- Pareto frontier and dominancy concept (Wikipedia-contributors 2002)

In recent years, multiobjective optimization methods were used in some WDS optimization research investigations. For instance, Baran et al. used six multiobjective algorithms to optimize the operation of water transmission lines based on four objectives. They considered reducing energy cost, reducing maintenance problems, reducing peak power demand, and reducing water level variation in a reservoir (Barán, von Lücken et al. 2005). Fu and Kapelan used a multiobjective optimization method for finding the best design of WDS based on pipe cost and system robustness that was the probability of simultaneously satisfying the minimum pressure constraints at all junctions (Fu and Kapelan 2011). Wu et al. also used a multiobjective method for optimizing the design of a WDS (Wu, Maier et al. 2013). The three objectives that were considered in this study were: 1) minimizing the total life cycle cost of the system, 2) maximizing the hydraulic reliability of the system, as represented by the resilience measure and 3) minimizing total life cycle GHG emissions. For the calculating operating cost, some simplified assumptions have been made. For instance, it was assumed that pumps should be refurbished in average each 20 years (without considering the operation plan). Also, it was assumed that the efficiency of pumps was constant (85%) through the whole simulation. For calculating GHG emission due to energy use for producing pipes and operating pumps, a constant annual rate was used. By using this constant rate, energy consumption was converted to the mass of emitted GHG. The network resilience measure was employed in this article as a hydraulic reliability measure. This measure makes use of the concept of the surplus power factor. It can be used to measure the resilience of a network subject to failure conditions, and thus the hydraulic reliability of the network, on the basis of both pressure and flow.

Converting multiobjective problem to a single objective problem makes the optimization algorithm simpler. Also, its optimum result is a single solution that can be used directly. Multiobjective optimization algorithms are more complicated that the single-objective methods and their result are a group of non-dominated solution that one of them should be selected as a final solution. Therefore, in comparison with single-objective methods, this final selection process is an extra step. Usually using the result of multiobjective algorithm needs human experts to evaluate solutions of the *Pareto frontier* and use their experience or some heuristic to select the optimum practical solution (based on their needs and preference). Despite these drawbacks, using multiobjective methods has some considerable advantages. By using multiobjective methods, finding optimum solutions with respect to one objective do not have any effect on the process of finding optimum value of other objectives. In addition, there is not any need to a normalizing and weighting method to add up the value of multiple objectives. Also, in the end, knowing all possible solutions of the *Pareto frontier*, gives an opportunity to experts to select a solution based on their needs.

Considering all discussed materials regarding optimization objectives and constraints, one example of formulating the problem of optimization of WDS operation is shown here:

$$\min(EPC, PE) \qquad \text{Equation 1}$$

That means, minimize energy and power cost (EPC) and pollutant emission (PE) while,

$\sum Q = 0$ (Conservation of mass at all junctions of WDS)

$\sum H = 0$ (Conservation of energy around all loops or path of WDS)

$H_{min} < H < H_{max}$ (Keeping the water pressure at junctions or water level in tanks in the allowed range)

$V_{min} < V < V_{max}$ (Keeping the velocity of all pipes of WDS in the allowed range)

$D_{min} < D < D_{max}$ (Keeping the operation duration of a pump in the allowed range)

$n < n_{max}$ (Limiting the maximum number of pump switch)

EPC and PE can be calculated by these formulas:

$$EPC = Energy\ usage\ cost + \ Power\ demand\ Cost \qquad \text{Equation 2}$$

$$EPC = \sum_{j=1}^{m} \sum_{i=1}^{n} P_{ij} D_i Ep_i + \max(P_{ij}) \,.\, Pp \qquad \text{Equation 3}$$

Where,

$P_{ij}$ is Power demand of pump j at duration i (e.g., kW);

$D_i$ is Duration i (e.g., hour);

$Ep_i$ is Energy price i (e.g., \$/kWh).

And

$$PE = \sum_{j=1}^{m} \sum_{i=1}^{n} P_{ij} D_i Em_i \ A \preccurlyeq B \ \ iff \begin{cases} f_i(A) \le f_i(B) & \forall i \in 1, \dots, M \\ \exists j \in 1, \dots, M & f_i(A) < f_i(B) \end{cases} \qquad \text{Equation 4}$$

Where,

$Em_i$ is Emission rate of power generation at duration i (e.g., kg/kWh);

Note that $P_{ij} D_i$ is equal to energy usage of pump j at duration i.

The electrical power demand of pump j at duration i ($P_{ij}$) can be calculated as:

$$P_{ij} = \frac{Q_j \rho g H_j}{\eta_j} A \preccurlyeq B \; iff \begin{cases} f_i(A) \leq f_i(B) & \forall i \in 1, \dots, M \\ \exists j \in 1, \dots, M & f_i(A) < f_i(B) \end{cases} \quad \text{Equation 5}$$

Where,

$Q_j$ is flow rate of pump j (e.g. m³/h);

$\rho$ is density of fluid (e.g. kg/m³)

$g$ is gravity of earth (9.81 m/s²)

$H_j$ is water head at pump j (e.g. m);

$\eta_j$ is overall (wire to water) efficiency of pump j (%);

Assuming that the density of water and gravitational acceleration of earth are constants, it can be seen that power demand of each pump is related to flow rate, water head and overall efficiency of the pump. Each pump has two nonlinear equations that relate head and efficiency to flow rate. So linear changes in flow rate cause nonlinear changes of the pump head and efficiency that consequently cause a nonlinear change of power demand and eventually energy usage.

Conservation of mass at each junction and conservation of energy around each loop or path are two implicit system constraints. Allowed pressure range of junctions, allowed velocity range of pipes, allowed duration of working of a pump and the maximum number of pump switches are other constraints. It should be noted that the highest and lowest range of level (volume) of water in tanks can be considered as maximum and minimum range of water head (pressure) at the node of the tank in WDS model. The result of research of Wang et al. reveal that a larger minimum level of tank volume will lead to higher electricity cost. Therefore, the minimum level should be determined carefully and set as low as possible (Wang, Chen et al. 2013). It is possible to add more constraints to the problem formulation to make the final result more practical. For instance, we can consider the minimum rest time between turning off a pump and turning it on again as a constraint. However, it should be noted that increasing constraints make the optimization problem more complicated and decrease the possibility of finding the best solution in a limited time.

If it is wanted to optimize water quality too, it should be considered in formulating the above-mentioned equations. It can be added as a constraint that shows the minimum concentration of chlorine; or water quality can be controlled by water age or even the lowest velocity of water in pipes. Although the optimizing WDS design based on water quality has been studied previously, research on optimizing pump operation based on water quality is not observed by the author in any articles. This lack of research can be explained by considering this fact that water quality is a function of initial chlorine concentration and size of pipes that defines velocity and travel time of water in the network. So pump schedule has a minor effect on the change of the chlorine concentration in the network. If it is not impossible, it is hard to control water quality in WDS by optimizing the pump schedule.

In the above formulation, three factors make this problem a nonlinear optimization. First of all, conservation of energy formula that includes the relation between flow and head is a nonlinear equation. Also, both energy consumption charge and emission factor are nonlinearly changing by time. These three factors do not let us use well-established and straightforward optimization methods that had been developed for linear problems. Beside nonlinearity of this problem, we have a more important issue that makes solving this problem considerably harder. This issue is non-convexity of the solution space of the problem. In almost all real world cases, we face multiple pumps that can be operated in a system with various parallel and series configurations. We know that pumps which are working in series or parallel configuration may affect each other's operational condition and efficiency. This effect may cause non-convexity in the relation between duration of operation of pumps and energy that used for conveying water from a source to a demand point. Also, electricity cost pattern and variation of the pollutant emission rate in time can exist in a shape of a non-convex function. All these factors together, create a non-convex function that relates operation variables (e.g. working periods of pumps) and cost or pollutant emission of operating WDS. Non-convexity of the solution space of this problem, can create a lot of local optimum points and make it impossible to be 100 percent sure of finding the global optimum solution (if it exists). Non-linearity and non-convexity of this problem make a lot of deterministic optimization algorithm inefficient for solving this problem.

## 1.4. Optimization Methods

In comparison with most of the engineering majors, optimization is a new field of study. Scientists, at first, started to use some deterministic techniques to find the optimum solutions of the problems. Some optimization methods like linear programming were created for solving linear problems. In linear problems the relation between variables of problem and optimization objective is linear. Although these mathematically based methods were working very well for linear problems, most of the real world engineering problem are non-linear. Specifically, in the field of water engineering, most of the problems like optimization of the design or operation of WDS were non-linear, non-convex problems. So linear methods were not able to solve these complex problems effectively. At the same time, an increase of engineering activities and limitations of resources encouraged engineers to enhance the effectiveness of their solutions. So needs for optimization techniques that can solve complicated problems with multiple constraints, multiple goals and a large number of possible solutions increased. Other methods like dynamic programming (DP) and non-linear programming (NLP) were used to solve these types of optimization problems. Most of these methods tested on small scale problems and provided good results. However, they were not efficient and successful in large and real size problems. At this period, using powerful computer systems increased the computational power considerably. This progress lets the engineers and researchers create and use new optimization methods that were highly computational demanding, but effective. Many researchers over the last 25 years focused on developing different techniques to optimize WDSs. Within the last two decades, many researchers have shifted the focus of WDS optimization from traditional and deterministic techniques, based on linear and non-linear programming, to the implementation of methods that were based on heuristics derived from nature (Zecchin, Maier et al. 2007), (Bi and Dandy 2013). Accordingly, after using deterministic methods, metaheuristic methods are the second group of optimization methods that are used in WDS optimization problems. In addition to these two major groups of the optimization method, we also can see a group of hybrid methods which are a combination of two above-mentioned groups (Zheng 2013). As these methods are not a separate group of optimization methods, and their name is self-explanatory, we prefer to stick with two-group categorization approach.

By reviewing some parts of previous optimization efforts in this field, and take a chronological look at the previous research effort, we can see the shift from using deterministic methods toward metaheuristic and evolutionary algorithms. We can start from the mid-70s. Alperovits and Shamir (1977) used a linear programming (LP) method to optimize the design of water network to reduce the cost of pipes. This method was inefficient and caused significant computational overhead (Zheng 2013). So Quindry et al. (1981), Calhoun (1981), Stephenson (1984) and Morgan and Goulter (1985) used slightly different LP methods to solve this least cost problem of WDS designing. All of these methods use some simplifying assumption and iterative procedures to convert a nonlinear problem to a linear problem and solve them with LP. However, other researchers started to use some NLP methods to solve this non-linear problem in its original form. Lansey and Mays (1989), Fujiwara and Khang (1990) used multi-step NLP methods to solve this problem. Despite all these early efforts on using deterministic algorithms to solve WDS optimization problems, they could not guarantee to find the global optimum (Zheng 2013). Although they were efficient in the search for the local optimum, they might get stuck in those locations. Most of these methods worked better with the tree shape (branched) networks and could not perform efficiently on medium or large scale looped networks. Also, they struggled to use discrete decision variables. Although early optimization efforts were focused on deterministic methods, a tendency towards them decreased in past decades. One of the latest and boldest research efforts in this category has been made by Samani and Mottaghi (2006). They used a binary linear programming method to solve this problem (Samani and Mottaghi 2006). This approach lets them use discrete decision variables, but even this approach just performed well for solving small problems.

Ulanicki et al. used a dynamic programming method to minimize the objective function mathematically by using gradients, and the calculations have been done by using vector algebra. Two full parameterization and partial parameterization approaches were investigated in their research. A full parameterization approach is one in which the optimal control problem is discretized and parameterized in time and directly solved using an NLP solver. In this case, all variables, including control, state, and algebraic variables are treated by the solver as decision variables. An alternative is a partial parameterization approach. In this method, the optimal control problem is discretized and parameterized in time, and a discrete-time optimal control problem is obtained. Subsequently, the state and algebraic variables are numerically resolved using a system simulator. The reduced gradients of the problem functions with respect to the controls were evaluated using either sensitivity equations or by integration of adjoint equations. In this case, only the control vector represents the decision variables (Ulanicki, Kahler et al. 2007).

In recent years, Evolutionary Computation has proven to be a powerful tool to solve optimal pump-scheduling problems (Barán, von Lücken et al. 2005). The great advantage of metaheuristic algorithms on deterministic methods is that they can be used for almost all types of optimization problems without considering the linearity or convexity of the problem. Metaheuristic algorithms do not require derivability, monotonicity, and continuity of the functions, but only require the objective function values. Metaheuristic algorithms cannot directly tackle the problem of the optimal design or operation of WDSs because the only constraint they handle is related to the range of the decision variables. Therefore, constraints related to the hydraulic behavior of the solution must be checked separately, or constraints can be converted to objectives (Marchi, Dandy et al. 2014). In addition, due to stochastic characteristics of these methods, it is not granted that they converge to the same solution during multiple runs, and also they cannot guarantee to find the global optimum solution (if it exists). The metaheuristic methods and specifically evolutionary algorithms (that are based on the evolution of a population of solution) had been used for solving some optimization problems. However, after introducing genetic algorithm (GA) by John Holland in the early 1970s, using evolutionary algorithms considerably increased in all engineering fields. In the case of WDS optimization, first time Simpson et al. suggested to use GA in the mid-90s (Simpson, Dandy et al. 1994). Although the most optimization efforts regarding WDS are related to reducing the capital cost of construction, there are also considerable researches about finding the optimum operational plan of pumps, finding the optimum location for sensors, calibrating hydraulic models, etc.

Lopez provided a summary table (Table 1) in his dissertation that shows WDS pump optimization efforts in a decade from 1995 to 2004 (Lopez-Ibanez 2009). This period is almost the first decade that researchers demonstrated a tendency to use metaheuristic algorithms for solving optimization problems in water-related engineering problems.

As it can be seen in Table 1, the Genetic Algorithm (GA) is one of the most used algorithm in optimization field and especially in water-related problems (Zheng 2013), (Wang, Liu et al. 2012). In comparison with the old deterministic algorithm, GA showed the better ability to find high-quality optimum solutions. Initially, common binary coding method was used for GA, but Dandy et al. used a gray coding scheme that helps GA to search the surrounding area of a good solution easier (Dandy, Simpson et al. 1996). Also, integer coding was used by Vairavamoorthy and Ali (Kalanithy Vairavamoorthy and Ali 2000). They used a tournament selection method; that prefer a feasible solution with lower pressure violation to other solution even if the objective function value was not the minimum value.

Table 1- Summary of optimization approaches for pump scheduling (Lopez-Ibanez 2009)

| Reference | Optimization algorithm | Tanks | Pumps | Hydraulic model | Representation |
|---|---|---|---|---|---|
| Mackle, Savic &Walters (1995) | Evolutionary algorithm | 1 | 4 | Regression model | Explicit |
| Ormsbee & Reddy (1995) | Nonlinear heuristic | 2 | 2 | Hydraulic simulation | Explicit |
| Nitivattananon, Sadowski & Quimpo (1996) | Dynamic programming | 8 | 10 | Mass balance | Explicit |
| Pezeshk & Helweg (1996) | Adaptive search optimization | 0 | 32 | Hydraulic simulation | Explicit |
| Savic, Walters & Schwab (1997) | Hybrid GA/MOEA | 1 | 4 | Regression model | Explicit |
| Andersen & Powell (1999) | Nonlinear Programing | 15 | 20 | Hydraulic simulation | Explicit |
| Simpson et al. (1999) | Evolutionary algorithm | 1 | 1 | EPANET | Implicit |
| Atkinson et al. (2000) | Evolutionary algorithm | 6 | 7 | Hydraulic simulation | Implicit |
| Goldman & Mays (2000) | Simulated annealing | 3 | 2 | EPANET | Explicit |
| Sakarya & Mays (2000) | Nonlinear optimization (GRG2) | 1 | 1 | EPANET | Explicit |
| Wegley, Eusuff & Lansey (2000) | Particle swarm optimization | 0 | 0 | EPANET | Explicit |
| Boulos et al. (2001) | Evolutionary algorithm | 1 | 3 | Hydraulic simulator (H2ONET) | Explicit |
| Ertin et al. (2001) | Dynamic programming | 1 | 3 | Mass balance | Explicit |
| Kazantzis et al. (2002) | Evolutionary algorithm | 1 | 1 | EPANET | Mixed |
| Sotelo, von Lucken & Baran (2002) | MOEAs: SPEA, NSGA, NSGAII & MOGA | 1 | 5 | Mass balance | Explicit |
| Dandy & Gibbs (2003) | Evolutionary algorithm | 1 | 1 | EPANET | Implicit |
| McCormick & Powell (2003b) | Progressive mixed integer programming | 10 | 35 | Mass balance | Explicit |
| McCormick & Powell (2004) | Simulated annealing | 10 | 35 | EPANET | Explicit |
| van Zyl, Savic &Walters (2004) | Hybrid GA | 2-6 | 3-7 | EPANET | Implicit |

Accordingly, in this method they did not need to include a pressure penalty in the objective function. Wu and Simpson used fast messy GA and it showed considerable improvement in efficiency of optimization in comparison with standard GA (Wu and Simpson 2001)

Figure 2 shows a visual classification of common metaheuristic methods (Wikipedia-contributors 2014). A Large number of these algorithms are inspired by nature, and most of them can be categorized in the evolutionary algorithm group. Most of these algorithms have been designed for single objective optimization. However, almost all of them can be modified to do multiobjective optimization too. Non-dominated Sorting Genetic Algorithm two (NSGA II) is one of the most used version of multiobjective GA.

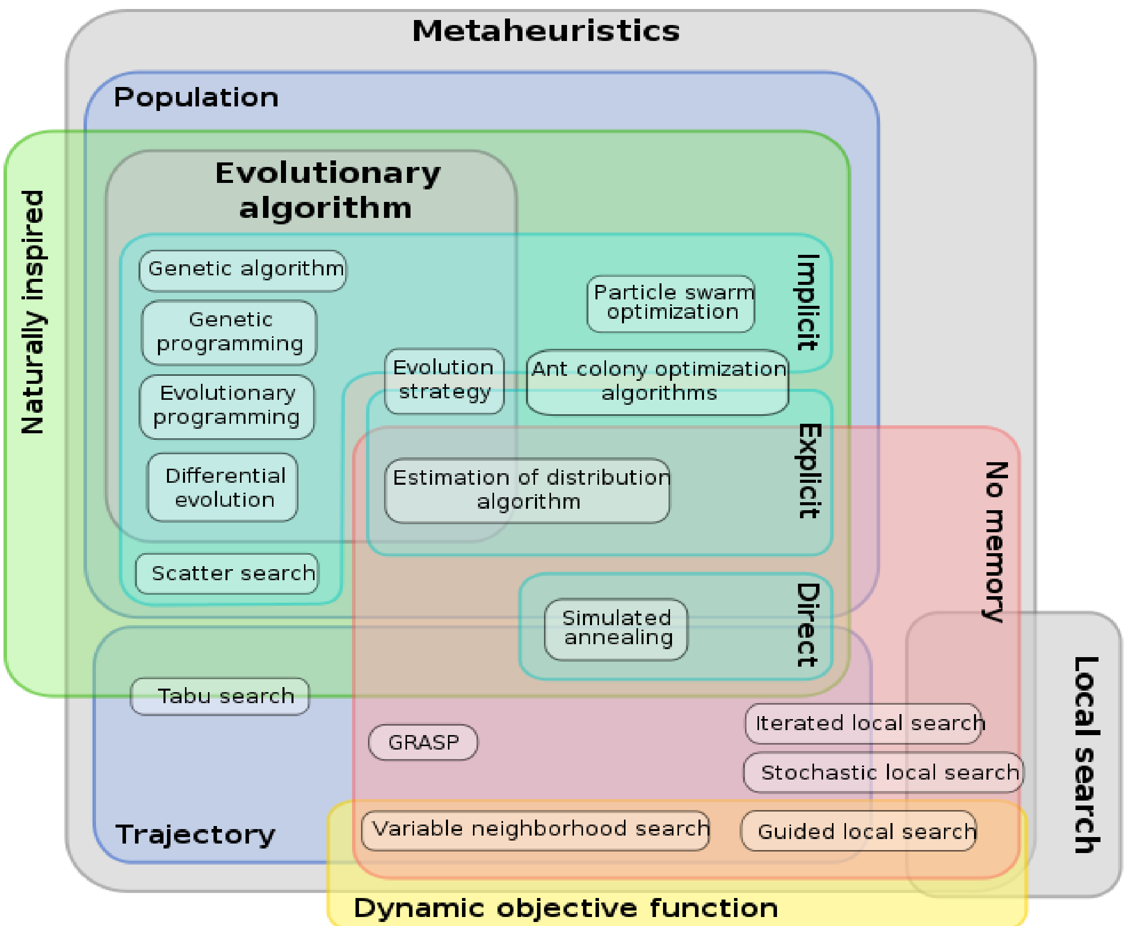



Figure 2- Visual classification of metaheuristic methods

As was mentioned earlier, many of these algorithms have been used for WDS optimization during past two decades. Zheng et al. provided Table 2 which shows the first significant implementation of metaheuristic algorithms for optimizing WDS through the past decade (Zheng, Zecchin et al. 2012). Zheng concludes that metaheuristic algorithms are better than deterministic algorithms in the case of WDS optimization, because: 1) They are better on exploration; 2)They can handle discrete search space better; 3) They can handle multiobjective optimization directly (Zheng 2013).

However, the efficiency of a metaheuristic algorithm will decrease by increasing the number of decision variable and expanding the solution space. As most of these algorithms are population-based, the whole optimization process needs significant computational resources and time as evaluations are required of each member of the population.

Table 2- First significant research efforts on usage of metaheuristic algorithm for optimizing the WDS design problem (Zheng 2013)

| Algorithm | First reference |
|---|---|
| Genetic algorithm (GA) | Simpson et al. (1994) |
| Simulated annealing (SA) | Loganathan et al. (1995) |
| Tabu search (TS) | Lippai et al. (1999) |
| Harmony search (HS) | Geem et al. (2002) |
| Shuffled frog leaping algorithm (SFLA) | Eusuff and Lansey (2003) |
| Ant colony optimization (ACO) | Maier et al. (2003) |
| ANN metamodels | Broad et al. (2005) |
| Particle swarm optimization (PSO) | Suribabu and Neelakantan (2006) |
| Scatter search (SS) | Lin et al. (2007) |
| Cross-entropy algorithm (CE) | Perelman and Ostfeld (2007) |
| Differential evolution (DE) | Suribabu (2010) |
| Honey-Bee Mating Optimization (HB) | Mohan and Babu (2010) |
| Genetic Heritage Evolution by Stochastic Transmission (GHEST) | Bolognesi et al. (2010) |

Although GA was the most used metaheuristic algorithm to optimize WDSs, there are also many studies in the past decade that adopted other metaheuristic algorithms. For instance, the particle swarm optimization (PSO) algorithm has received considerable attention in the literature, and differential evolution (DE) is one of the latest techniques which is applied to this problem (Marchi, Dandy et al. 2014). Also Zecchin et al. applied five types of Ant Colony (AC) algorithms (which are based on the foraging behavior of ants) to four WDS design problems: Ant System (AS), Ant colony system (ACS), Elitist Ant System (ASelite), Elitist-Rank Ant System (ASrank) and Max-Min Ant System (MMAS). They compared their results with other researchers' results in the same test cases. In comparison with other algorithms in the literature, ACs and specially ASrank and MMAS, showed very promising results. Some of these ACs performed better for a small problem and some for a large problem. Compared with MMAS, ASrank was more efficient, but ASrank did not perform as well as MMAS in the bigger and more challenging case studies. These abilities related to exploring (the ability of the algorithm to search vast areas of the solution space) and exploiting (the ability of the algorithm to search more thoroughly near areas where good solutions have been found previously) abilities of the algorithm. MMAS act better in those cases due to its greater ability to explore (resulting, however, in longer search time), while still exploiting the best information (Zecchin, Maier et al. 2007). Similarly, Lopez-Ibanez states in his Ph.D. thesis that there are some successful implementations of the common evolutionary algorithm in optimizing pump schedule, but there is a lack of experimental analysis of comparing another alternative algorithm for doing this task (Lopez-Ibanez 2009). So he tried to test AC algorithm for optimizing some water networks. The two ant colony algorithm was compared with single and multi-objective GA algorithms. The optimization goal of the single-objective test was to reduce energy usage, and objective of the multiobjective test was to reduce energy usage and pump switches. It was stated that AC outperformed all common evolutionary algorithm in literature and this work, for Richmond WDS. However, this conclusion cannot be made completely for the second network (Van Zyl network; Lopez-Ibanez 2009).

Chu et al. used an immune algorithm that is inspired by the biological defense process of the immune system to solve New York City tunnel design problem and found the least-cost design. They also combined immune algorithm with GA to get a better result. They found that in comparison with GA, the immune algorithm can find the optimum solution in less number of iterations (Chu, Lin et al. 2008). Bagirov et al. used particle swarm optimization, an artificial bee colony, and firefly algorithms to optimize pump operation of a small WDS with two tanks and three pumps. They also compared the results by using three criteria: the “optimal solution" obtained; (b) the efficiency; and (c) robustness. Their tests showed that the artificial bee colony is the most robust and the firefly is the most efficient and accurate algorithm for optimizing pump operation in small systems (Bagirov, Ahmed et al. 2012).

Moreover, Simulated Annealing (SA), Honey-bee Mating Optimization (HBMO) and Gene Expression Programming (GEP) have been used in past for designing and selecting the optimum pipe diameter for water distribution networks (Wang, Liu et al. 2012).

In most test cases, the new algorithms could not outperform GA. However as it was mentioned previously, recently other algorithms like ant colony (AC) and differential evolution (DE) were used for WDS optimization, and they showed that can produce high-quality results with high efficiency. Although some contradictions might be related to specific test cases or selection of parameters of the algorithm (Zheng 2013). So, in general, it can be stated that up to this point GA could provide acceptable results in the case of WDS optimization, but other algorithms like DE and AC that were adjusted properly for a specific problem could produce a better result.

In addition to the general algorithms and methods that can be used for solving this type of optimization problems, some optimization parameters and factors might affect the efficiency and effectiveness of the optimization process. In the following paragraphs, some of these parameters and factors have been reviewed briefly and some studies that changed these parameters to get better results are mentioned.

On most of WDS optimization researches, the optimization horizon is typically chosen as 24 hours to take account of daily demand patterns and electricity tariff structure. A choice of optimization cycle less than 24 hours will not take full advantage of cheaper tariff periods (Zheng and Morad 2012). The operational planning horizon defines the optimization horizon. Forecast of the demands for an operational planning horizon, which in most urban system ranges from a minimum of 24 hours up to a maximum of one week, depending on the size of the storage relative to the demands (Shamir and Salomons 2008).

To define the minimum time interval of pump operation, we should consider the demand change in time and relation between decreasing the time interval and increasing optimization efforts. It was investigated that intervals higher than one hour prevent algorithm to find optimum solutions. On the other hand, time intervals less than one hour make the searching process for optimum algorithm longer so one-hour time interval is suggested as a moderate and efficient value (Lopez-Ibanez 2009).

Most of researches in this field are focused on finding an optimum pump schedule for fixed speed pumps (FSP). It is understandable that researchers initially focused on

FSPs, as these types of pumps can be found in almost all pump stations. However, variable speed pumps (VSP) are as common as FSPs in new WDSs. In addition, FSPs have just two possible states (ON or OFF) and has smaller solution space in comparison with VSPs that may have a various operational state with multiple rotational speeds that provide more energy reduction and operation optimization opportunities. Despite these facts, few researches worked on optimization of pump operation plan of VSPs. From limited researches on optimizing operational plan of VSPs; we can point to Wu et al. research. They did a WDS design optimization by using GA and including VSPs. They reported that comparing the same optimization process with FSPs showed that using VSPs can reduce the total cost and GHG emission from WDS (Wu, Simpson et al. 2012). Similarly, Hashemi et al. used VSPs instead of FSPs in their pump operation optimization and stated that using VSPs can lead to up to 10% reduction in pumping energy cost. (Hashemi, Tabesh et al. 2013).

In addition to the wide variety in optimization algorithms that have been used in this field, other factors make each of individual research in this field different from others. For instance, there are many different approaches that researchers have taken to pump optimization in WDS. Numerous researchers tried to find the best pump schedule that was represented by on and off blocks of time that show operation status of a pump during a predefined time interval (e.g. one hour; Sadatiyan Abkenar, Stanley et al. 2014). Van Zyl used the tank-level controlled triggers, and Lopez-Ibanez tried to find the best string of trigger that defines the start and end time of a working period of a pump (van Zyl, Savic et al. 2004), (Lopez-Ibanez 2009). The first method that is mostly used by researchers called the binary representation of pump schedule and can easily be used by the evolutionary algorithm like GA that looks at each on and off blocks of time like a gene or specification of the organism (solution or pump schedule). The second methods of coding are mostly called level-controlled trigger and time-controlled trigger. These solution representation methods successfully used by Lopez Ibanez in GA and AC (Lopez-Ibanez, Prasad et al. 2008). Lopez-Ibanez investigated various representations of pump schedule in his thesis and suggested that time-controlled trigger representation can lead to a better result and ensure maximum limit of switches per pump in comparison with level-controlled trigger representation. However, his result also showed that time-controlled trigger based representation did not have considerable advantages on the common binary representation (Lopez-Ibanez 2009). An example of pump operation plan that is represented by the binary and time-controlled trigger is shown in Figure 3.

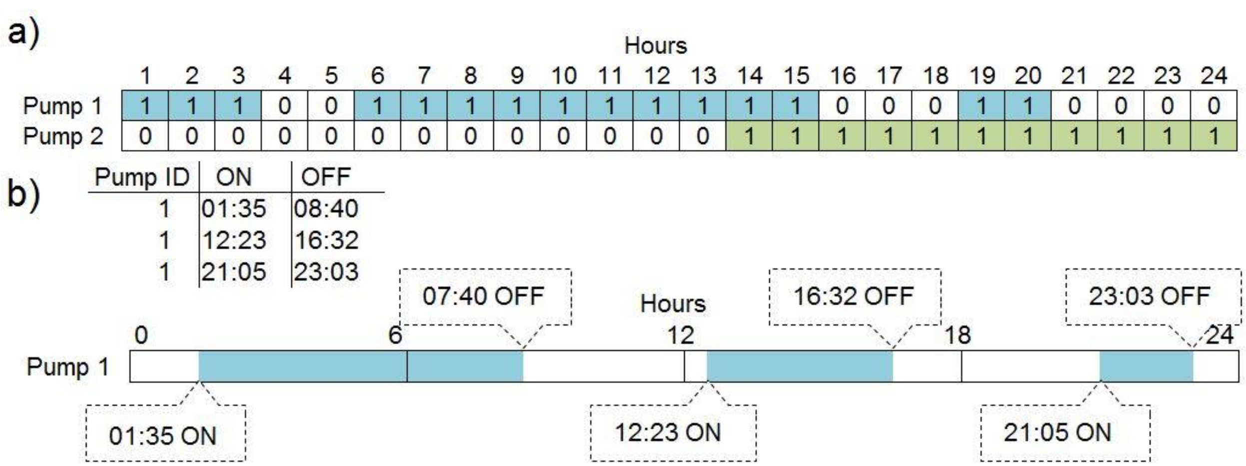



Figure 3- Pump schedule representation by (a) binary and (b) time-controlled trigger methods (Sadatiyan Abkenar, Stanley et al. 2014)

Abiodun and Ismail used a multiobjective weighted sum Genetic Algorithm with a discrete method of coding and 120-bit memory space for each pump (Abiodun and Ismail 2013). As the main core of optimization process, a Genetic Algorithm with the real-number chromosome (continuous) was used. During crossover step, the whole schedule of a pump gets exchanged (not just the time intervals). Shamir and Salomons used optiGA (a general purpose VB GA) in their Haifa WDS operation optimization research. optiGA's built-in option can be used to optimize binary, real, and integer variables in case of optimizing a network that has VSP (Shamir and Salomons 2008).

Metaheuristic algorithms highly depend on the adjustment of parameters of the algorithm for a specific problem. Tolson provided a table (Table 3) that shows the number of parameters of some famous metaheuristic algorithms that have been used for WDS design optimization (Tolson, Asadzadeh et al. 2009).

Table 3- Number of parameters of algorithms (including penalty function parameters and excluding stopping criteria) (Tolson, Asadzadeh et al. 2009)

| Optimization Algorithm | Number of parameters |
|---|---|
| GA (GENOME) | 8 |
| MSATS7 | 8 |
| PSO | 6 |
| PSO variant | 5 |
| SFLANET | 5 |
| HS | 5 |
| MMAS ACO | 4 |
| CE | 3 |
| HD-DDS | 1 |

Wang et al. used a gene expression programming (GEP) method for optimizing the design of Hanoi WDS. They also used a range of mutation and crossover probability in their research. The crossover probabilities vary from 1.0 to 0.7 with each 0.05 interval, and the mutation probabilities differ from 0 to 0.05 with each 0.01 interval. The best parameter values adopted were population size of 100, 0.9 probability of crossover, 0.03 probability of mutation and the maximum number of generations was set to 500 (Wang, Liu et al. 2012). The obtained optimum solution compared with six other studies that solved the Hanoi benchmark problem previously. The optimization result showed that this algorithm was as good as other algorithms, but not considerably better.

Abiodun and Ismail used a GA algorithm with a population size of 100 and 5000 generations. Crossover and mutation rates of 0.4 and 0.05 were used, respectively (Abiodun and Ismail 2013). Wu and Behandish used a GA with 50000 generations and population size of 100. Also, they considered the periodic water level, emergency lower-bound and overtopping constraints. (Wu and Behandish 2012). Zheng and Morad used a GA with 50000 generations and population size of 100 (Zheng and Morad 2012). Wu et al. also performed a multiobjective optimization of WDS design. They used an NSGA II algorithm with 3000 generation, 500 population size and crossover and mutation rate of 0.9 and 0.03 respectively (Wu, Maier et al. 2013). Wang et al. used a GA with the population size of 100 and 1000 generation. Crossover and mutation rate of 0.9 and 0.5 were used respectively. High mutation rate in this study is justifiable; as they were using local search and did not need minor mutation for finding the local optimums (Wang, Chang et al. 2009).

Baran et al. used and compared six multiobjective evolutionary algorithms for optimizing operation of a small water transmission line. These algorithms are non-dominated sorting genetic algorithm (NSGA), strength Pareto evolutionary algorithm (SPEA), non-dominated sorting genetic algorithm two (NSGA II), controlled elitist non-dominated sorting genetic algorithm (CNSGA), niched Pareto genetic algorithm (NPGA), and multiobjective genetic algorithm (MOGA). Comparing Pareto front of all algorithms by using six different comparison factors shows that SPEA was the best algorithm and after that NGSA II produced good results (Barán, von Lücken et al. 2005). Besides all these algorithms, in order to transfer general solutions to a feasible solution, a heuristic constraint algorithm was used. The population size of 100 and 20000 generation and crossover and mutation rate of 0.8 and 0.01 were used respectively in this study.

Gibbs, Maier, et al. realized that adjusting parameters of optimization method is related to a characteristic of problem and parameter of optimization method should be selected for each specific problem to get the best result. So they reviewed many methods in computer science field to calibrate GA and selected and tested two methods that seem useful but have not been used in a practical field like WDS optimization. Two parameter setting methods with one base condition and commonly suggested parameters for GA were implemented to optimize chlorine injection in Cherry Hill-Brushy network, and the results were compared. About probability of crossover and mutation, two separate tests were done. In one set of test probability of crossover and mutation kept constant and in the other test self-adoptive parameters were used. Three different max generation stopping criteria were used for each case and each test combination repeated 13 times with random initialization. The average result of runs with various stopping criteria for each parameter setting method was used for comparison. The results compared with t-test and they were considered different if they showed significant changes with 95% confidence. The result demonstrated that constant crossover and mutation parameter answer slightly better than self-adaptive parameters. However, both calibration methods were better than selecting a common parameter for GA.

Hernandez et al. used a software package called Dynamic Real-time Adaptive Genetic Algorithm-Artificial Neural Network (DRAGA-ANN). Some practical constraints (e.g. tank overtopping and emptying of tanks, the maximum power usage of the pump station, etc.) were considered and infeasible solution penalized during calculation of fitness. A very complex electricity tariff with six different rates during a day and for various months of the year was used. GA was modified a little bit. So instead of increasing the probability of selecting the fittest solution as a parent, that solution selected directly to become parents. Also, the optimum solution of the previous simulation was used as an initial guess of the next optimization to reduce the required time of optimization process. GA with 2000 generations and crossover and a mutation probability of 0.765 and 0.002 were used respectively. The optimization goal was finding a pump schedule that uses minimum energy and utilize as much as possible from a cheaper source of water and without violating any of the operational constraints (Martínez, Hernández et al. 2007).

As it was explained previously, in addition to the two general categories of deterministic and metaheuristic algorithms, there is another hybrid group of algorithms that have been proposed by some researchers for optimizing WDSs (Tolson, Asadzadeh et al. 2009), (Zheng, Simpson et al. 2014), (Giacomello, Kapelan et al. 2013), (Liu, Yuan et al. 2011), (Milan Čistý and Bajtek 2009). These hybrid methods are combining deterministic algorithm with metaheuristic algorithm. Typically, one algorithm finds the promising regions of the solution space and another algorithm performs the search of those sections to find the best solution. So it can be stated that, in general, two separate algorithms play exploration and exploitation roles. Conceptually this idea is interesting (that is, enlisting the strength of two different algorithms for each of the exploration and exploitation tasks). However in practice, such an algorithm has not yet found its way to the optimization software market. Using local search with metaheuristic algorithm can be seen in researches about two decades ago when Savic et al. enhanced their GA by a local search. This idea improved the possibility of that GA can find the optimal solution (Wang, Chang et al. 2009). One of the common problems of the proposed hybrid algorithm is their high computational demand, especially in dealing with large size problems (Zheng 2013). As a hybrid algorithm, some local search methods can be used for polishing the final solution of metaheuristic algorithms. For instance Bi and Dandy slightly reducing the size of pipes and changing concentration of chlorine to gain the best solution around the founded optimum solution of metaheuristic algorithm for WDS design optimization. As another example, beside of multiobjective weighted sum Genetic Algorithm, a Greedy algorithm was used by Abiodun and Ismail to generate near optimum initial solutions. Also in the final step of each generation, local search is used to improve the result and find the best solution in a neighborhood. It helps to find local optimums among members of a generation, before crossover and mutation of the next generation (that may throw solutions to another part of the solution space before finding the optimum in the current search area) (Abiodun and Ismail 2013).

### 1.5. Hydraulic Modeling

In all optimization methods, solutions should be evaluated during the optimization process, and finally, the best solution can be reported as the optimum (or near optimum) result. For the evaluating solution, we need to know the response of WDS to any change of decision variables. There are various methods for evaluating the effect of suggested operational plan on network status. Empirical models (e.g. mass balance, process-based model), simplified network hydraulic models, and complete network simulation models are some examples that can be used to calculate the effect of changes of decision variable on responses of the system (Rao and Alvarruiz 2007). The first two groups of methods are relatively fast, and although they have been used for small WDSs previously, they cannot provide accurate results for large networks with a lot of non-linearity. Wang et al. used a simple test case with one tank and four FSPs. They used some formulations to relate tank level and pump status to flow, or pressure of pipes and junctions and they did not use complete hydraulic model (Wang, Chang et al. 2009, Wang, Chen et al. 2013). Abiodun and Ismail modeled a considerably smaller water transmission system. Therefore, it was possible to form an array of a finite combination of the initial status of the system and the effects on pressure and tank level, energy usage, etc. (Abiodun and Ismail 2013). Similarly, the method that Baran et al. used did not include hydraulic analyzing of the model. They simply considered various combinations of 5 pumps and one tank and calculated energy usage of each case. Tank level was calculated based on mass balance, so extended period simulation (EPS) was not used (Barán, von Lücken et al. 2005).

In some deterministic optimization methods, equations that relate decision variables to the status of the network can be implemented explicitly in optimization formulation. In deterministic methods, unlike most of the evolutionary algorithms and other metaheuristic algorithms, the optimization part of computer code and hydraulic simulation part of the code are not separate. For instance Blaszczyk et al. explicitly implemented the mass balance equation in their non-linear optimization formulation (Błaszczyk, Karbowski et al. 2013). Few researchers even tried to include the hydraulic simulation part explicitly in the evolutionary algorithm. For instance, Wu et al. used the mass balance at joints and energy balance at loops and considered them as optimization constraints of their multiobjective GA (Wu, Maier et al. 2013). They did not use hydraulic modeling software (e.g. EPANET) and the EPS model implicitly and separate from the main core of the optimization algorithm. However, most of the recent optimization efforts, which used metaheuristic methods, separate hydraulic simulation part of computer code from the optimization algorithm.

One of the most frequently used methods is creating a high-fidelity computer model of WDS that inputs operational orders and initial status of the system and after solving hydraulic equations provides the final status of the system after a defined period. Over the past decades, a lot of researchers try to improve this high fidelity and realistic modeling methods. They created some computer programs that by a user-friendly interface let the user creates the WDS model and run the simulation to get final results. One of the most famous free and publicly available software in this group is EPANET2 that is published by the US EPA (Rossman 2000). Also, some commercial software is available that are widely used in WDS design and rehabilitation projects. (Bentley 2014). Lopez-Ibanez reviewed about 20 articles between 1995 to 2004 and reported that most of the researchers used complete hydraulic simulation to evaluate the effect of decision variables on the status of the hydraulic network (Table 1) (Lopez-Ibanez 2009). Researchers try to create models with high physical detail and calibrate them to get accurate results. If these hydraulic simulation computer models get calibrated very well, they can provide accurate results that are very close to the real condition of the system. For instance, Preis et al. modeled a large water network in Singapore with more than 20000 pipes and 19000 junctions and equipped it with eight pressure and flow rate sensors. Measured data were used to calibrate the hydraulic model of the system to reduce the modified square error of pressure and flow rate. Finally, after three months, cross-validation of the calibrated model with some 24 and 48 independent measurements of flow and pressure showed that pressure result of the model was in agreement with direct measurements (especially for locations that system has more sensors). (Preis, Allen et al. 2011). However, most of the time these models need to solve a large matrix of hydraulic equations to find numerous unknowns of the WDS. Therefore, using the high-fidelity computer models are computationally demanding. In the case of optimization, it may be necessary to iterate the hydraulic simulation of the WDS thousands of times. This computational burden of high fidelity hydraulic models can slow down the optimization process considerably, preventing good optimization methods to be used for large and real size optimization problems.

The speed of the optimization process can be increased by various approaches. Certainly, the use of high-efficiency optimization algorithms can be useful. In addition, reduction in the size of the solution space and search space allows the algorithm to reach the optimum point more rapidly. Since the solution space of large WDS is so huge and vast, even reducing the size of solution space might not help a lot. However, there is some possibility to reduce required computational demand to finish an optimization iteration. So another method that works effectively for algorithms that need a lot of iteration (e.g. evolutionary algorithms) is reducing the required time for each iteration. As the required computational time of hydraulic simulation is the bottleneck of reducing the required time for each iteration, parallel processing has been used for reducing the time of this part of the calculation (Guidolin, Fu et al. 2012, Wu and Behandish 2012). Lopez-Ibanez explained in his thesis that the computation time of hydraulic modeling part of a real size problem is some order of magnitude greater than the required computation resource for the optimization algorithm by itself (Lopez-Ibanez 2009). In parallel processing approach, multiple processors simulate hydraulic network simultaneously. So if for one iteration of an optimization process, a WDS needs to be simulated multiple time, the required time can almost be divided by the number of processors that are doing the simultaneous calculation. Lopez-Ibanez used parallel computing to reduce the required time for hydraulic simulation part. In this regard, they changed the code of EPANET to be able to use that for parallel modeling of WDSs (Lopez-Ibanez 2009). Although parallel processing can reduce the required time for each iteration, it does not diminish the total required computation. Therefore, the use of this method requires computers that have multiple processors with high computational power.

Another option for reducing optimization time is using some modeling methods, which need less computational power, but can provide results with acceptable accuracy. Using surrogate modeling methods that sometimes called as metamodeling techniques is the alternative way. For instance, one of the techniques that used in this field is using artificial neural networks (ANN) to create a metamodel for WDS. Wu and Behandish used both parallel processing and ANN based metamodel and found that, although parallel processing can improve the speed of WDS optimization, it is not as effective as using metamodel that is created by an artificial neural network (ANN) (Wu and Behandish 2012). Using ANN is one method from a larger group of methods that Razavi et Al. called them surrogate modeling methods and thoroughly investigated them in the water resources field in one of their papers (Razavi, Tolson et al. 2012). They express that although various motivations may cause to use surrogate models, the main reason is reducing computational resource demand of the computationally intensive modeling process. A large group of the most relevant papers in the water resource field selected by Razavi et al. (48 articles) to investigate the usage of surrogate modeling in this field. They classified surrogate modeling into two main groups of response surface modeling and lower fidelity modeling. Except artificial neural network (ANN), other response surface modeling uses data-driven function approximation techniques to approximate the model response. Usually, each response surface can be used to approximate one response of the system. However, lower fidelity modeling is a physically simplified model that needs lower computational resources and respectively may produce some results with lower accuracy. The simplified model can be used to find all required results of the system at the same time.

Three methods can be used to create the low-fidelity physically based surrogates: a) models with reduced numerical accuracy, b) approximation by model order reduction and c) reducing the physical model with lower details. Two first methods most of the time produce better results. Most of these methods have widely been used in other fields, but in water science, they have not been used extensively. However, some well-known WDS modeling programs could be found in the market that has a skeletonizing component to reduce the physical model and form a lower fidelity and fast surrogate models (Bentley 2014). Generally with respect to the response surface surrogates, lower-fidelity surrogates expected to emulate better unexpected regions of the domain of original model, and they can result more accurate in extrapolating. There are some approaches to reduce discrepancies between low-fidelity and high-fidelity models (i.e. correction functions, space-mapping, and hybrid strategies).

Shamir and Salomons used a reduced model of Haifa WDS for operation optimization. The results of network reduction algorithm depended on the demands since the properties of equivalent pipes (which are created when a junction is removed) depended on demand at that node. However, it was claimed that within the range of variation of demands over time (factor of 2 between high and low demands), the reduced network reproduced results of the full model with very high fidelity. Calculating reduced network just took three seconds, so in the case of any change to the network (e.g. pipe breakage) reduced model could be modified very fast and does not interfere real-time optimization process. The validity of reduction of the model was measured by checking the similarity of tank levels over time in the reduced model and the full model. Also, the resulted control routine was used again in the full network model, to compare the pressures at the junction with reduced model results (Shamir and Salomons 2008). They compared the result of optimization of whole and reduced model of Haifa WDS. The full model calculation showed 12% reduction in energy cost, but reduced model genetic algorithm (RM_GA) showed about 10% reduction in cost. Comparison of full and reduced model result revealed that tank level and pressure results had less than 2% difference. Optimization with RM-GA for a 31 day period took about 8 hours (almost 15 min for each day). RA-GA reduced the simulation time by a factor of about 15. Although Shamir and Salomons showed that RM-GA was considerably faster than GA and full hydraulic model, in the following paragraphs, when we start to talk about the result of research on ANN, it still can be seen that reduced model is slower than ANN+GA. Therefore working on reduced model is not as promising as working with ANN metamodel on the speed of optimization.

As it was explained reducing the time of each hydraulic modeling can lead to reduced time of one iteration of the optimization process and consequently can reduce the whole optimization time. In this regard, using response surface modeling techniques received much attention during past ten years. Machine learning methods like artificial neural network (ANN) or support vector machine (SVM) can be used to generate metamodels. Razavi et al. also concluded that ANNs is the most commonly used response surface modeling techniques. Support vector machines (SVM) are almost new learning technique in this field (Razavi, Tolson et al. 2012). In WDS optimization researches, these response surface modeling techniques most of the time referred as metamodeling techniques (Martínez, Hernández et al. 2007), (Broad, Maier et al. 2010), (Behandish and Wu 2014). Metamodels do not use energy and mass conservation equation to find the effect of a new pump schedule on the pressure at junctions and level of water in tanks. However, they can be generated for each network to map data from inputs (e.g. current tank level, pump schedule, water demand, etc.) to outputs (e.g. tank level after a defined time interval, energy usage of pumps, etc.). Metamodels are much faster than complete network simulation models, and their usage for real-time optimization of WDS increased trough past decade. The main result of using metamodels instead of high fidelity models is speeding up the optimization process and make the real-time optimization of medium and large WDS possible. Although due to the nature of metamodeling some accuracy will be lost while using them and it may cause some difficulties in the process of finding the optimum solution. For instance, a small error of metamodel may cause the feasible solution to be considered as infeasible and vice versa.

In most of the reviewed works ANNs with one hidden layer were used to create the surrogate model. It has been proved mathematically that ANN is capable of representing arbitrarily complex, nonlinear processes which relate the inputs and outputs of any system via a finite number of nodes on hidden layers (Rao and Alvarruiz 2007). For a problem with n decision variable, we need an ANN with one hidden layer of n-1 neurons or two hidden layers of (n/2)+3 neurons to fit the original function. It was mentioned that although ANN with more than one hidden layer is theoretically able to model most of the complex problems, because they are more prone to fall into poor local minima in training, using them do not guarantee to get better results. Selecting a proper structure of ANN is the most important things that users should do. Figure 4 is a schematic of an ANN that receives input information from input layer (layer of blue nodes on the left side) and process them through the hidden layer (layer of orange nodes in the middle) and gives the result via nodes of the output layer (layer of green nodes on the right side). Each ANN has one input and one output layer, but it can have multiple hidden layers. While training the ANN, a group of known input and corresponding output data will be presented to the ANN and will be used for adjusting the weight of connections between neurons (nodes). When an ANN is trained, it can receive input data and pass them through the network and generate output data that are similar to the real outputs of the system. All training techniques try to adjust the weight of connections between neurons to reduce the final error between calculated outputs and real outputs of the system. As it is shown in Figure 4, input values of an ANN for WDS can be initial tank levels, status, and speed of pumps, the status of valves and water demands. Outputs also can be the final water level in tanks after the simulation duration (e.g. one hour), energy use of pumps and pressure at junctions of WDS, etc.

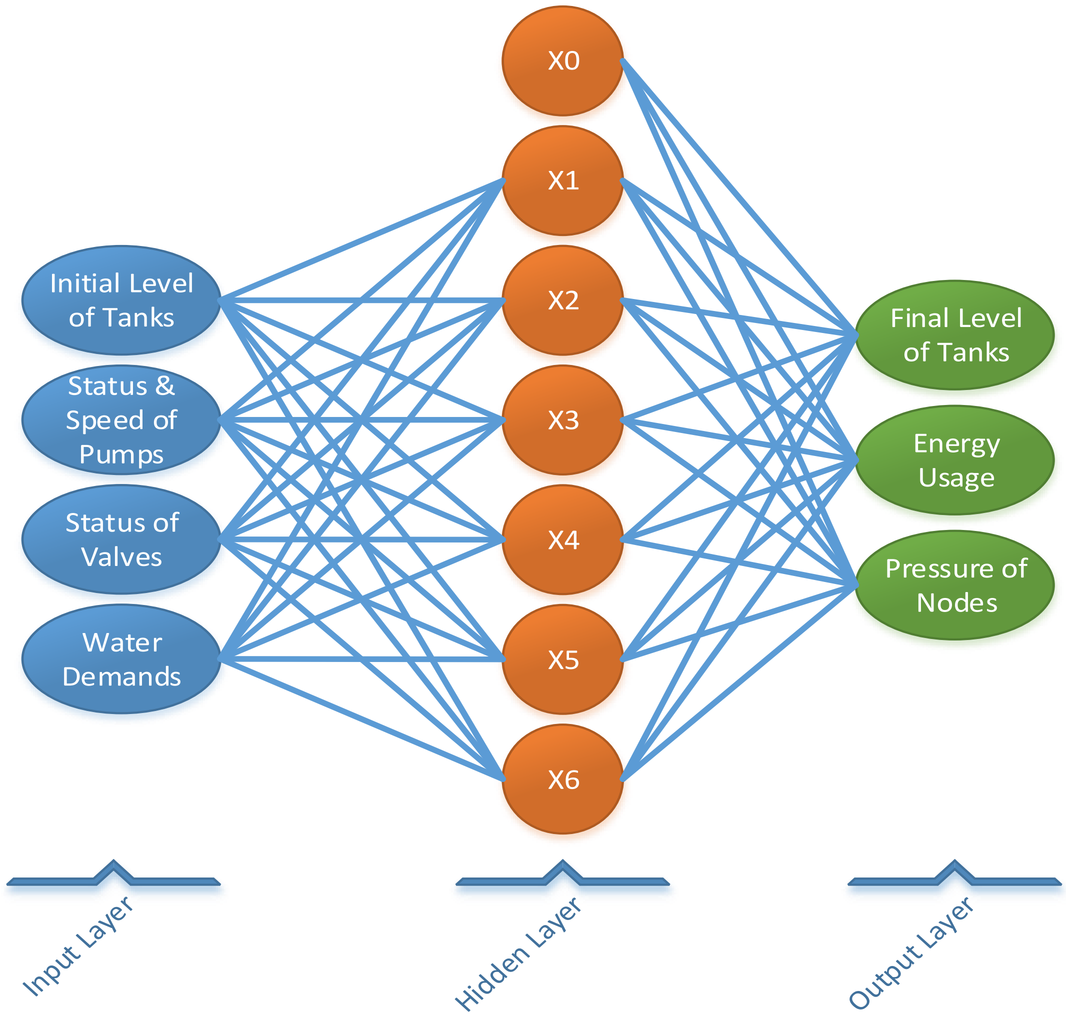



Figure 4- Schematic of an ANN for WDS

Razavi et al. explained that they think the design of experiment (DoE) is necessary to be sure to create a good metamodel unless the initial set is large enough that can cover the whole function domain. They believe that the DoE is in fact required, and a sufficiently large, well-distributed initial set of design sites to develop the metamodel is a key factor in the success of a metamodeling practice. The minimum size of DoE that is suggested by various formulas in this paper are almost small, but it was stated that the optimum size of DoE depends on the condition of the problem and available computational resource. Minimum theoretical design sites required for ANN and SVM are higher than other methods. The Minimum number of design sites for ANN in reviewed articles were about 150 to 300, but other methods could be used with the much smaller design set for the same problem. More design sites help to fit original function more effectively, but it is more computationally demanding. ANN receives some calculated outputs and corresponding inputs (design sites) and by back propagation method tries to adjust parameters that relate inputs to outputs and minimize calculation error. The frequency of updating metamodel with new information was related to the available computational resource.

It was mentioned that ANN is one of the most computational expensive methods for training. Using surrogate modeling for a problem with constraint (and especially constraints that included in the objective function by penalty) the accuracy of the metamodel is really important and defines the feasibility and infeasibility of the solution. It was suggested to check the feasibility of solution time by time by original model and train metamodel on both feasible and infeasible spaces to improve its accuracy. Creating metamodel for a high dimensional problem is more challenging. It is possible to create a better model.by:

- Reducing the number of decision variable and filtering out unimportant decision variables
- Reducing the size of solution space
- Decomposing problem into a set of smaller scale sub-problems

Except ANN and SVM, most of the methods do not have internal validation process. In contrast, flexible models like ANN created models, are highly prone to overfitting. Emulating multiple outputs are possible with ANN and correlation between outputs can be modeled with it. However, other methods mostly can emulate just one output at a time. Although emulating multiple outputs at a time has some advantages and can take into account the correlation of outputs, but sometimes it makes the training of ANN very hard.

Razavi et al. categorized the meta-model enabled analysis frameworks in four groups: a) Basic Sequential Framework, b) Adaptive-Recursive Framework, c) Metamodel-Embedded Evolution Framework and d) Approximation Uncertainty Based Framework. (Razavi, Tolson, et al. 2012). The Basic Sequential Framework is the simplest framework and includes three major steps for a) design of experiment (DoE), b) fitting meta-model c) substituting meta-model for the original model in performing the analysis of interest. Adaptive-Recursive Framework is similar to the first framework, but the points that have been found in the third step will be evaluated by original model and added to the set of design to update meta-model. Metamodel-Embedded Evolution Framework is almost similar to Adaptive-Recursive Framework, but it does not have the first step for DoE. At first, a population-based optimization algorithm run initially on the original function for a few generations. All the individuals evaluated by the original function in the course of the first generations are then used as design sites for metamodel fitting. In the following generations, individuals are selectively evaluated by either the metamodel or the original model. Approximation Uncertainty Based Framework seems the most sophisticated framework and added and extension to the adaptive recursive framework and it also includes the uncertainties associated with the approximation of the original model. Different methods can be used to calculate the approximation uncertainty, but specifically for ANN, Bayesian neural networks provide the variance of prediction.

As it was mentioned previously, using, ANN based metamodels increased in the WDS optimization field. For instance, Broad et al. used ANN based hydraulic metamodels for optimizing operation of large and a real system to have least pressure violation, energy use and quality issues (Broad, Maier, et al. 2010). Rather than directly calling the hydraulic model for each GA trial evaluation, Wu and Behandish employed ANN based surrogate models to replace the hydraulic simulator in optimization (Wu and Behandish 2012). Bi and Dandy created an ANN by 5000 training sites with 40 junctions in the hidden layer and used it to optimize WDS design. Their ANN has 22.inputs. However, each of the five output (4 pressure and one chlorine concentration of 5 critical junctions) created with separate ANN (Bi and Dandy 2013). Roa and Alvarruiz explain that ANN trained with a set of known input and outputs that have been produced by running an EPANET model by randomly generated initial condition. Then it was tested by test set that was almost in the size of 20% of the training set and was produced by the same method. The designed ANN has an input layer, one hidden layer, and one output layer. The input set contains the combination of pump/valve settings, demands and initial water levels in storage tanks while the output set corresponds to the power consumption of pumps, resulting water levels in storage tanks, pressures and flow rates at critical locations throughout the network. By trial and error, it was decided to put 20 neurons in hidden layer to get the best results. Also, it was realized that training set that has more than 2000 training sites does not improve ANN results. They showed that Using ANN provides a good result for hydraulic simulation and could produce almost accurate results and decreased computation time by a factor of 10. ANN have been used for water quality modeling too, but its quality modeling results were not as accurate as hydraulic simulation results. It was mentioned that chlorine residue (as a factor of water quality) mostly depends on the size of the network and is not strongly depended on the input values of ANN, so ANN could not find the good relation between inputs and water quality in this case. So as an alternative (and not completely good one) minimum velocity of water in some important pipe have been modeled and investigated (Rao and Alvarruiz 2007). Hernandez et al. also used a three-layer, feed-forward ANN (with 24, 100, 15 neurons in input, hidden and output layers respectively) to gain the domain knowledge of a hydraulic model. 2500 training samples and 800 test samples were used, and the resulting root mean squared error (RMSE) for training and test set were 1.2% and 1.3% respectively. Max elevation of tank increased about two meters to produce some infeasible and out of range conditions for training more accurate ANN. Pumping status, valve setting, demands of DMAs and storage tank level were used as input values and power consumption, Flow rates of selected location of DMAs, pressures of entrance points of DMAs and storage level considered as outputs. ANN structure designed based on trial and error in multiple steps (Martínez, Hernández, et al. 2007).

The previous studies showed that process of using The ANN to create a hydraulic metamodel takes some times and needs many trial and error. It also is related to understanding the interaction of valves completely. It was shown that in some researches, selecting the ANN structure, number of neurons in the hidden layer and required number of training samples need a lot of trial and error that is time-consuming (Rao and Alvarruiz

2007), (Martínez, Hernández, et al. 2007), (Bi and Dandy 2013). If the process of shaping ANN gets automatic, it can be used to optimize networks even while it has pipe breakage.

Jamieson et al. explain that as ANN is trained by samples that have been simulated by EPS, it might not provide results that are completely similar to the real WDS. The base assumption in training ANN is to have a calibrated and accurate EPS model. Many parameters (e.g. accurate age and roughness of pipes, etc.) are unknown, so having completely accurate EPS model is not always the case. Therefore, ANN can be improved by using SCADA data from the real system. ANN can be trained further with real data to provide some results that are even better than the EPS model (Jamieson, Shamir, et al. 2007).

Similarly, Zheng and Morad coupled ANN based meta-model with GA to find the optimized pump schedule of demand monitoring zone (DMZ) of Oldham in Greater Manchester in England on a real-time basis (Zheng and Morad 2012). They found out that ANN needs some improvement. Because it cannot provide a good result when it faces overtopping of some tanks (as it does not have any physical sense similar to EPS). So they suggested using real data from SCADA to overcome the problem of aggregated tank level error caused by using ANN metamodel. They compared results of optimization method with actual operation data and claimed that, using this method, caused about 15-20% energy saving. For training ANN, 1,000×24-hour scenarios with one-hour time-step were used. ANN tested at snap-shot testing level and then verified by extended period simulation (EPS). In this study, the ANN equipped with an external constraint to prevent overtopping of tanks.

Fu and Kapelan explained that offline ANN needed a large number of training examples to cover the whole solution space, and its accuracy did not increase around the optimum solution. So they suggested to use online ANN and combine it with NSGAII to train an ANN with a smaller number of examples at first and then retrain it with adjusted frequency to provide a better approximation of the original model around the optimum solution of each generation. To keep the training time constant new training examples were replaced with old examples. At first EPANET 2 used to solve the hydraulic model for some first generations and after collecting enough examples, trained ANN replaced with EPANET. Pareto front of each generation simulated again with EPANET and the results were used instead of old training data in the training set (Fu and Kapelan 2011). Although it should be noted that replacing data of the old train set with some training examples around Pareto front might decrease the accuracy of ANN far from Pareto front that consequently decreases exportability of the optimizer. New York Tunnel Problem (NYTP) was used as a case study of this research. ANN has 21 inputs for the diameter of 21 pipes and one hidden layer with ten nodes and one node in output layer to result in the fitness of each solution. Training accuracy was 0.0001 with the maximum epoch of 100. They investigated the effect of characteristics of the training set on ANN and reported that NYTP needs a training set of 1000 to 2000 examples and new training cases of 100. Higher new training cases (about 500 and 1000) caused some instability. ANN of NYTP was almost small, and its training only takes about two seconds. Anytown was another case study in this research. This network was larger than NYTP. Based on selected parameters in this research, Anytown network needed a higher frequency of retraining to get good results in the direction of optimization path. So the used training size was 300.

Bi and Dandy tried to use on-line metamodels for hydraulic and quality modeling in their research and used differential evolution (DE) for WDS design optimization (Bi and Dandy 2013). ANN toolbox of MATLAB with its default parameters have been used for creating the metamodel. To make sure that the optimum solution in each generation is feasible EPANET checked it. For defining retraining process, these parameters have been considered: the size of training dataset, the number of generations between retrainings and the total number of retrainings. As ANN based metamodel cannot effectively and efficiently model pressure and a chlorine concentration of all junctions, so some critical junction should be selected. Bi and Dandy suggested a 5-state statistical method for selecting critical junctions in WDS. The five stages of selecting critical junctions are: data range check, demand check, dominance check, correlation check and frequency of critically check. They also used a method for generating high-quality training data for ANN.

Broad et al. state that ANN had been used successfully for optimizing small example of WDS but in large cases it has some difficulties to approximate the main model accurately. These problems might cause some main issues during optimization. Since a small error of a metamodel may cause a feasible solution to be considered as infeasible and vice versa. These types of problem can be reduced by checking best solutions by the high fidelity model and using local search to improve the quality of the final solution. Also, other methods like skeletonization and decomposition or Gaussian elimination can be used to simplify WDS and reduce training time of ANN and complexity of network for approximation purposes. Duration of simulation and simulation resolution should be adjusted correctly to decrease inaccuracy, especially in the case of water quality modeling. In that case, simulation duration should be longer than water age. Control duration also should be large enough to minimize any numerical irregularities. (Broad, Maier, et al. 2010).

It can be seen that just a handful of researches have been done in this regards. Some of them just used the basic sequential framework and created an off-line ANN that was trained one time before optimization process. However, there are few other researches which used an adaptive recursive framework or metamodel-embedded evolution framework used on-line and repetitive training of metamodel. These studies mostly focused on hydraulic modeling not quality modeling, and most of them used hydraulic modeling for WDS design optimization, not operation optimization. However, new researches during the last three years show the interest of researchers in this field and possibility of improvements.

Pump operation optimization result is sensitive to the accuracy of the hydraulic model and predicted water demand. If the hydraulic model is not calibrated or the predicted water demand pattern is not similar to the real water demand, optimum pump schedule might not be able to answer required pressure or demand of the system in real practice. So it is recommended to use model calibrator tools and water demand prediction techniques to create high-quality inputs for the optimization process that helps to provide practical and optimized results. Investigating different calibrator tools and demand prediction techniques are beyond the scope of this study. Here it is assumed that a calibrated hydraulic model with accurate water demand pattern is available as an input to the pump operation optimizer tool.

## 1.6. Test Cases and Benchmark Problems

By reviewing previous researches, it can be seen that most of the studies focused on small scale WDS or water transmission lines. These networks have a handful of pipes, junctions, pumps and occasionally one or two elevated tanks. For instance, Wang et al. used a small water transmission line with one tank and 4 FSPs (Wang, Chen et al. 2013). Similarly, Abiodun and Ismail used a small water transmission line with one tank and 5 FSPs for their bi-objective optimization. Baran et al. also used a small transmission line with one tank and 5 FSP in their research. Wu et al. used two small water transmission lines with one tank, 1 FSP, 36 pipes and 16 junctions in the first system and two tanks, 1 FSP, 41 pipes and 19 junctions in the second system (Wu, Maier et al. 2013). Wu, Simpson, et al. also used a small system with three tanks, 3 FSPs, eight pipes and five junctions in their WDS design optimization (Wu, Simpson et al. 2012). A small portion of real systems are similar to small test networks of these researches, but most of the time we face large networks with a couple of hundred pipes, junctions, and a considerable number of pumps, valves, tanks, etc. There are a few studies that tried to optimize real and large size WDS. For instance, some large WDS were used in optimization test cases of POWADIMA projects of the European Commission. In this project, a combination of GA and ANN were tested on a modified model of Anytown and two real WDNs (Haifa and Valencia). Haifa was selected as a small and hilly network with multiple tanks and Valencia as large, but almost a flat network (Jamieson, Shamir et al. 2007). Rao and Alvarruiz used Anytown benchmark WDN for their study, and they changed it a little bit to form Anytown Modified WDN. This network contains three tanks, 3 FSPs, 41 pipes and 19 junctions (Rao and Alvarruiz 2007). Martinez et al. studied Valencia WDS and reported promising results. Valencia WDS has two water treatment plants with significantly different production costs and serves 1.2 million people. The hydraulic model includes two tanks, 17 pumps, ten valves, 725 junctions, 772 pipes and 6 DMAs. They showed that by using an ANN based hydraulic metamodel, the optimization of Valencia WDN with 17 pumps takes about 10 minutes. The ANN was about 94 times faster than EPANET simulation for Valencia WDS (Martínez, Hernández et al. 2007). Another large scale system worth mentioning from the WDS optimization literature is the large and real system demand monitoring zone (DMZ) of Oldham in Greater Manchester in England. This case was investigated by Zheng and Morad and also Wu and Behandish in their optimization research (Zheng and Morad 2012), (Wu and Behandish 2012). Oldham WDS is composed of more than 3200 pipes, 3500 junctions, 420 valves, five reservoirs, 12 storage tanks and 19 fixed speed pumps (FSP). Wu and Behandish tested two optimization methods on demand monitoring zone (DMZ) of Oldham (Wu and Behandish 2012). In addition to real WDSs, some researchers developed an abstract model of a WDS and used in the hypothetical system for their studies. For instance, Marchi, Dandy, et al. introduced a rural network with 476 pipes and 98 loops (Marchi, Dandy et al. 2014).

As explained in the introduction section of this thesis, the size of the solution space is directly related to the number of pumps. Therefore, networks with more pumps need exponentially more iteration, time and computational power to find the optimum solution. In addition, larger networks with a higher number of pipes and connection are harder and more time consuming to evaluate hydraulically. Even one iteration of a solution of a larger network is more computationally demanding than the small network. Although Wang et al., Abiodun et al., Van Zyl et al. and others reported good results in implementing some methods for optimizing a small WDS (Wang, Chen et al. 2013), (Abiodun and Ismail 2013), (van Zyl, Savic et al. 2004), it does not mean that the same method works for solving the same type of problem with a larger network.

There are a considerable number of articles that optimized benchmark WDSs. The advantage of these benchmark systems is that they are used by many researchers during the past decades and the best known near optimum solution of them are available for comparison with the result of new studies. Some famous benchmark networks are D-town, Anytown, Hanoi and two loop networks. Hanoi and two loop WDSs are simple systems without pumps that mostly are used for design optimization. The Hanoi water distribution network has 32 junctions and 34 pipes structured in 3 loops. No pumping facilities have been considered since only a single fixed head source at an elevation of 100 m is available. The minimum head requirement at all junctions is fixed at 30 m. The set of commercially available diameters is [12, 16, 20, 24, 30, 40 inches]. The Hanoi system has been used in a lot of WDS design optimization researches like Wang et al. researches(Wang, Liu et al. 2012). D-town and Anytown are almost large scale (or, at least, simplified version of large scale networks) that can be used for various purposes. None of the above-mentioned networks have variable speed pumps (VSP). D-town has been used as the benchmark WDS of a series of scientific competition with the name of Battle of Water Networks (BWN). The D-town network has 11 pumps in 5 pump stations. It also has seven tanks. Although this network has some loops, it is a branched network in general and has different separate pressure zones.

NYTP, double NYTP, HP are famous case studies, but they are small in comparison with real problems. The Balerma system that contains 454 pipes and 3 loops created to simulate a more realistic problem. However, even this system was almost hydraulically simple. Hanoi problem is just good to check if an algorithm can find a good feasible solution or not. As the feasible solution space of this problem is limited. Zucchin et al. used these four WDSs in their research on WDS design optimization by AC: the two reservoir problem (TRP), the New York tunnel problem (NYTP), the Hanoi problem (HP), and the doubled New York tunnel problem (2-NYTP) (Zecchin, Maier et al. 2007). Bi and Dandy did some tests on design optimization of the New York tunnel problem (NYTP), modified New York tunnel problem (MNYTP) and Jilin network (JN) (Bi and Dandy 2013).

Figure 5 shows the schematics of some of these WDSs. Also, Table 4 provides some information about them that can be used to gain an idea of the size and characteristics of each test case.

The Haifa-B model which serves a population of some 60,000 and ranges in elevation over 450 meters was used by Shamir and Salomons. The full model WDS includes 987 pipes, 867 junctions, nine storage tanks, eight pressure reducing valve, 17 FSPs in 5 pumping stations and six demand manage areas (DMA). The reduced model contains 77 junctions and 92 links and maintains all pumps and tanks (Shamir and Salomons 2008). Ulanicki et al. used a WDS in the south of France, as a case study. This system includes five pump stations. One of the pump stations has VSPs, and others have FSPs. The WDS also has three valves, 48 junctions, 37 pipes and seven tanks (Ulanicki, Kahler et al. 2007).

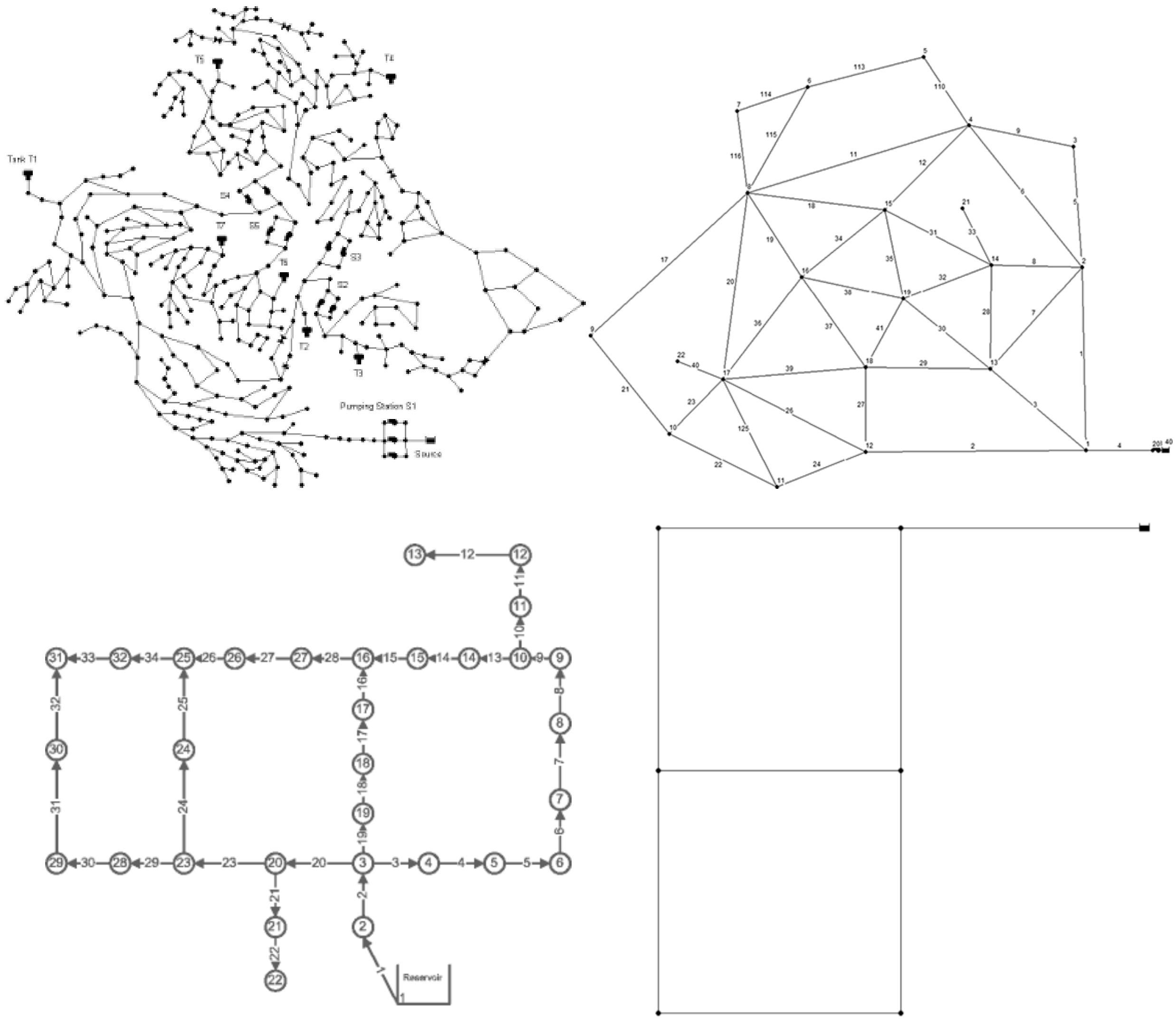



Figure 5- Schematics of some of the benchmark WDNs: D-town (top left), Anytown (top right), Hanoi (bottom left), Two loops (bottom right)

Morley created a framework for evolutionary optimization in WDSs (Morley 2008). He used most famous benchmark WDSs in his dissertation and result of his work, and others are published in the Exeter University of UK. It formed a reliable database of hydraulic models of WDSs and categorized them to be used in five groups of researches: a) Expansion, b) Layout, c) Operation, d) Sensor Placement, e) Design / Resilience. The collected WDS hydraulic models have been provided for public use and are available through the website of the center for water system (CWS) of Exeter University. Some of the famous benchmark WDS of this database are Wolf-Cordera Ranch, EXNET, Gessler 1985, Anytown, New York tunnel, D-Town, Hanoi, Richmond (CWS 2014). Wang et al. (who are also from CWS group of Exeter University) did an extensive two-objective optimization test on 12 small to large WDS (Wang, Guidolin et al. 2014). They put the hydraulic model file of these network on the CWS website for public use. The provided hydraulic models are used, and their information is listed in Table 4. They were sorted based on the number of junctions in WDS. So by the first look at Table 4, we can see the size of the network is increasing from top to bottom. However, beside the number of junctions in a network, there are other parameters (e.g. number of pipes, loops, etc.) that show the size of the system. By taking a look at the pump and tank column, we can see that some benchmark systems does not have any pump and tank and most of them just have one source of water.

These networks mostly were used as a WDS design, resilience or expansion optimization test cases. The main usage of each benchmark system based on its characteristics and previous usages in the literature is categorized that can be seen in the last column of Table 4. Although some of these networks like New York Tunnel, Anytown or Gessler 1985, had been used for more than two decades in WDS related optimization researches, and they have almost well-established known global optimum result, most of them had been created and can be used for WDS design optimization and do not have enough number of pump and tank that can be used for WDS operation optimization (Simpson, Dandy et al. 1994), (Laurie J Murphy, Dandy et al. 1994). Accordingly, D-Town, Richmond, and Monroe WDSs can be used as operation optimization test cases. Anytown, Wolf Cordera Ranch and skeletonized version of Richmond network also are simpler WDSs that can be used for this type of researches.

Table 4- Benchmark WDSs for optimization tests

| WDS | Junction | Pipe | Loop | Pump | Tank | Reservoir | Valves | Main Usage |
|---|---|---|---|---|---|---|---|---|
| Two-Loop Network | 7 | 8 | 2 | 0 | 0 | 1 | 0 | Design |
| Gessler 1985 | 12 | 14 | 3 | 0 | 0 | 2 | 0 | Design |
| Two-Reservoir Network | 12 | 17 | 6 | 0 | 0 | 2 | 0 | Design |
| New York Tunnel | 20 | 21 | 2 | 0 | 0 | 1 | 0 | Design-Expansion |
| GoYang | 23 | 31 | 9 | 1 | 0 | 1 | 0 | Design |
| Anytown | 25 | 47 | 23 | 3 | 2 | 1 | 0 | Design-Expansion |
| Blacksburg Network | 31 | 35 | 5 | 0 | 0 | 1 | 0 | Design |
| Hanoi | 32 | 34 | 3 | 0 | 0 | 1 | 0 | Design |
| BakRyan | 36 | 58 | 23 | 0 | 0 | 1 | 0 | Design |
| Fossolo | 37 | 58 | 22 | 0 | 0 | 1 | 0 | Design |
| Richmond - Skeletonized | 48 | 51 | 4 | 7 | 6 | 1 | 0 | Operation |
| Pescara | 82 | 100 | 19 | 0 | 0 | 3 | 0 | Design |
| Modena | 276 | 317 | 42 | 0 | 0 | 4 | 0 | Design |
| D-Town | 399 | 443 | 45 | 11 | 7 | 1 | 5 | Design-Expansion |
| Balerma Irrigation Network | 451 | 454 | 4 | 0 | 0 | 4 | 0 | Design |
| Richmond | 879 | 965 | 87 | 7 | 6 | 1 | 1 | Operation |
| Monroe | 1540 | 1971 | 432 | 13 | 3 | 1 | 0 | Operation |
| Wolf Cordera Ranch | 1790 | 2005 | 216 | 6 | 0 | 4 | 4 | Design-fire hydrant |
| Exnet | 1896 | 2469 | 574 | 0 | 0 | 2 | 2 | Design |

The Richmond water distribution system is owned by Yorkshire Water in the UK, and the owner gave permission for this system to be used in academic studies. It was used in a Ph.D. research project on operational optimization of water distribution systems by Kobus van Zyl (van Zyl, Savic et al. 2004). It Also was used by Giacomello to test a fast hybrid optimization method (Giacomello, Kapelan et al. 2013). So it is suggested by CWS as a benchmark system for the operational optimization problem. Reducing the energy cost is the objective of the optimization of the Richmond WDS. This WDS has seven pumps in 6 pump stations. It also has six tanks that each of them is connected to one pump station. In the original problem, the water level in tanks is controlled by pumps, so there is not any constraint on the pressure at junctions. Each pump station has unique off-peak and on-peak electricity tariff. The best operational cost found in the literature (excluding penalty cost of £0.15 per pump switch) was £33 982 from Van Zyl‘s studies (van Zyl, Savic et al. 2004).

Except Monroe system that has two variable speed pumps (VSP), none of the other benchmark networks have VSPs. Although it should be mentioned that Monroe WDS have not been used as frequently as other benchmark networks.

In addition to above-mentioned benchmark systems that had been used previously, Jolly et al. recently selected 12 real WDSs in Kentucky and formed a database that can be used in WDS researches and especially optimization efforts (Jolly;, Lothes; et al. 2014). These WDSs classified based on their characteristics. For instance, three configurations defined for categorizing topology of networks: a) Branch, b) gridded and c) loop configurations. Total demand of network has been distributed based on pipe diameter, and some adjustment has been made for transmission mains. As all systems were in almost the medium range size, a typical diurnal demand curve published by the American Water Works Association was used as the demand curve. Jolly et al. expressed that they tried to create these models as close as possible to a real world system, but because of security issues, they intentionally modified some part of the models. Also, some sensitivity test has been done on the network to calibrate them before publishing.

### 1.7. Analyzing Results and Comparison Methods

The difference in objectives, methods, study cases, etc. provides a broad outlook regarding optimization of WDSs. However, these differences make it hard to compare results of various studies and select the most effective method for solving this type of problems.

In early studies, researchers just tried to suggest an optimization method, test it on a WDSs and report the results. However, after a while, new studies tried to use the result of previous studies (specifically the best known near-optimum result of a benchmark WDS) and compare their new results with previous studies. Although comparing the quality of new results with the results of previous studies can be a good base for comparison, multiple factors involved in optimization makes it hard to find an entirely suitable case of comparison. In fact, many factors can influence the final results: the problem characteristics, the number of function evaluations allowed, the variable coding method, the nature of the objective function, the specific algorithm operators and the range in which the algorithm parameters were tested (Marchi, Dandy et al. 2014). Besides, almost all metaheuristic algorithms and population-based methods have some stochastic characteristics that cause them to produce slightly different results even within multiple runs of the same algorithm. So it is really important to use statistical indicators and tests that show if the observed difference in the results of different methods are based on the stochastic nature of the algorithms or if it shows a meaningful difference between the methods. Bi and Dandy conduct 30 runs with each calibrated algorithm in their researches, and average, min, max and standard deviation of results were used for comparison between the results of various methods (Bi and Dandy 2013). Lopez-Ibanez also used the statistical method in his thesis, for comparing the result of tests with different parameters (Lopez-Ibanez 2009).

The efficiency of different methods can be compared in terms of required time or computational efforts to find the optimum or a near optimum solution. To have a fair and accurate comparison, it is important to include all computational demand of the whole optimization process. Also, required time is highly dependent on the computational power of used hardware and software. These issues can interfere with reporting the comparison result and cause some faulty conclusions. Zucchin et al. researched on WDS design optimization by different types of AC algorithm. 20 runs were conducted with each algorithm for each test case, and statistical results of these runs were compared with the results of other algorithms of previous researches on the same test cases. The performance of the algorithms was measured based on solution quality (i.e., the best cost that is the minimum cost found in a run) and search efficiency (i.e., search time that is the number of function evaluations required to find the best cost for each run). Also, solutions feasibility of the results of this study and other's results was assessed by EPANET (Zecchin, Maier et al. 2007).

Marchi, Dandy, et al. tries to come up with a methodology for comparing evolutionary algorithms for optimizing WDSs. The general proposed comparison methodology has these steps: a) selection of the EA techniques to be compared; b) selection of appropriate test problems; c) calibration of each EA algorithm for each test problem; d) final runs of each EA method for each test problem; and e) analysis of the results. Also, it was suggested that all algorithm should use the same hydraulic solver. It is proposed that, for the first step, at least, one of these algorithms should be included in comparison: Genetic algorithms (Simpson et al. 1994; Dandy et al. 1996; Savic and Walters 1997; Reca and Martínez 2006), differential evolution (Suribabu 2010; Vasan and

Simonovic 2010; Zheng et al. 2011a) and ant colony optimization (Maier et al. 2003; Zecchin et al. 2005) due to their numerous successful applications to WDS problems. They suggested using the range of parameter of each algorithm found from literature and also check algorithm with a variable amount of parameter to calibrate it for a specific problem. Then multiple runs should be conducted with a calibrated algorithm and average, standard deviation and percentage of result that found global optimum should be calculated. For avoiding the weight pressure penalty to be another calibration parameter, this policy suggested to be followed: For comparing solutions always feasible solutions are better than infeasible solutions and among infeasible solutions, solutions with the lowest violation are better (Marchi, Dandy et al. 2014).

In the case of multiobjective optimization, the comparison is even more complicated. As the result of a multi-objective optimization method is a Pareto frontier of non-dominated solutions, it is not possible to compare the value of a single optimal solution with the best known near-optimum solution. In this regard, Baran et al. suggested some metrics to compare Pareto frontier result of various algorithms (Barán, von Lücken et al. 2005). Fu and Kapelan determined the optimal value with S-metric and diversity metric. S-metric indicates the closeness of the solution to the theoretical Pareto Front and spread of solutions over objective space (Fu and Kapelan 2011).

## 1.8. Gaps in Research

In both cases of increasing efficiency of optimization process for optimizing real size water network in a short period and incorporating environmental goals in the optimization process, researches are ongoing, but it was not possible for the author to find comprehensive research efforts that could do the optimization process of a medium or large size WDS effectively and include environmental objectives besides energy usage and cost objectives.

Initially, researchers used deterministic methods (e.g. linear programming) to optimize the design and operation of WDS. They mostly were focused on reducing the cost of building and operating networks. Although during past two decades, researchers inclined toward using metaheuristic and an evolutionary algorithm. However, most of the research in this area had been done on some small and simple WDSs.

Up to the present day, few researchers have tried to include the environmental effects of water networks in the optimization process. Some of them considered the amount of pollution caused by producing pipes and pollutant emission of the energy usage of pump stations. In most of these researches, a constant value was used to convert the amount of used pipe or energy to greenhouse gasses (GHG) emission. Considering a constant emission rate for using energy at any time and location is not a realistic assumption. This method mostly relates the emission reduction to energy reduction and does not take into account the change of pollutant emission by using energy from different sources at different location and time.

The operating schedule of pumps from the past does not affect the future solution. However, it seems that to control the max number of pump cycle and max working time of pump, the previous working condition of pumps should be taken into account, and there is not any research that comprehensively investigated this issue.

Although optimization pump scheduling solutions have been done for some real WDSs, there is still a lack of a robust and general pump scheduling methodology that is efficient and effective for medium and large size water distribution systems. Such systems typically contain dozens of pumps, resulting in a vast solution space; and multiple storage tanks, resulting in highly non-linear hydraulic constraints. In addition, evaluating each of the trial solutions usually requires an extended period hydraulic simulation. A typical optimization run requires thousands of scenarios to be separately evaluated before a near-optimum solution is obtained. As a result, the optimization run times for real-world systems can exceed several hours or even days (Wu and Behandish 2012).

Even commercial optimization tools on the market that are mostly using the result of recent studies in this field can optimize a medium of large size WDS in a reasonable amount of time. For instance, Darwin Scheduler part of WaterGEMS software that is one of the most well-known pump operation optimizer in the market can handle less than 200 controls effectively (Bentley 2014). It means that it can effectively optimize the operational plan of WDS with at most eight pumps during a 24 hour simulation period with one hour time intervals (8×24=192<200). Although this product uses fast, messy GA (which is one of the states of art optimization algorithm in this field), it takes about one day for this tool to optimize a medium size WDS (Alighalehbabakhani, Abkenar et al. 2014). It also just able to optimize $CO_2$ emission of used energy based on a constant emission rate

It was mentioned in previous sections that some researchers defined constraints to limit the number of pump switches or control water level in tanks, etc. However, it was not possible for the author to find any comprehensive study that takes into account all essential factors of providing a pump operation plan that fulfills all practical requirements of a pump operation and can be used directly in real condition. It means that result of most of the current optimization methods cannot be used directly in an actual operation, and they need to be modified with an operator to become more practical. Here is a list of items that have not comprehensively been studied in previous research and need more attention in future studies:

- Adjusting exploration and exploitation abilities of the optimization algorithm and using effective local search or other more intelligent heuristics beside main optimizer algorithm to improve the efficiency of optimization
- Conducting optimization of medium and large scale WDSs with practical details like using VSPs and valves
- Using time and location dependent rate of emission instead of fixed emission rate values
- Defining better measure for optimizing based on maintenance objective and adding constraints about working hours and cycles of pumps
- Making clear the method of providing efficient ANN for creating WDS metamodels. Defining various causes of the inefficiency of ANN, investigating the reason of aggregated tank level error, while using metamodel and suggesting possible solutions for improving the metamodel creation process
- Using real data beside the results of calibrated hydraulic model to train ANN and using other machine learning algorithm for producing metamodel
- Define a methodology for creating online and live ANN that keeps using newer training sets always to maintain the metamodel updated
- Doing more investigation on the shape of the Pareto front and the effects of different parameters on that
- Defining a guideline to help the user to select the best solution among the solutions of Pareto frontier

- Investigating more about the effect of optimization parameters (e.g. crossover and mutation rates) on optimizing efficiency
- Defining a clear process to improve metamodel and prevent it from producing infeasible results and understand overtopping or emptying of tanks
- Adding more intelligence to optimization method to select solutions that are more applicable to real operation from a practical point of view. Moreover, considering operational constraints during the optimization process, as much as possible

# CHAPTER 2 RESEARCH HYPOTHESIS AND METHODOLOGY

## 2.1. Statement of the Problem and Research Hypothesis

It was explained in the previous chapter that although considerable research has been completed on pump operation optimization during past decades, this type of problem remains an immense challenge for application to real networks of mid- to large-size. Calculation of the near optimum solution for a medium or large size network (more than 10 pumps) requires a considerable amount of time. Another complication is that the pollutant emission caused by energy usage has not been studied extensively yet. Even in the few studies that pollutant emission was considered as an objective of optimization, the pollutant emission was aligned directly with the amount of energy used. Therefore, the locational and temporal variations in pollutant emission is a novel area for research. Finally, although in different studies various types of constraints and heuristics were used to find the near optimum solution that meets practical needs of pump operation (e.g. number of pump switches, tank level control, running time of pumps, etc.), there is no available optimization tool and method that consider all requirements, for finding a practical optimum pump schedule. Accordingly, the objective of this study is to develop a pump operation optimization tool for WDSs that has the following characteristics:

- Applies multiobjective optimization to reduce energy consumption, electricity usage cost and pollutant emission in a spatially- and temporally-sensitive manner.
- Efficient and precise computational algorithms.
- Ease of use – with limited supervision and training.

The primary hypothesis of this study is:

It is possible to develop a pump operation optimization tool that decreases both energy usage and related pollutant emissions for real WDSs within a reasonable simulation time period.

This hypothesis has three components:

- It is possible to develop a pump operation optimization tool that can find near optimum solution for medium and large size WDS in a relatively reasonable time (fraction of an hour to fraction of a day).
- It is possible to develop a pump operation optimization tool that can find a near optimum solution which is practical and can be used directly and with minimum expert supervision for operation of the WDS
- It is possible to develop a pump operation optimization tool that can find a near-optimum solution which decreases energy usage and related pollutant emission and is sensitive to time and location of generating energy.

As described in previous sections, there are multiple benefits of this research. In particular, the “value added” of this project includes:

- Decreasing the required computational resource for optimizing pump operation and decreasing the required time for finding the near-optimum solution.
- Collecting and compiling various heuristics and optimization details that had been used previously in pump operation optimization studies to increase the efficiency of the main optimization algorithm and increase the quality of the final near optimum solution.
- Making the final solution of optimization process practical so that it can be used directly in real operation with minimum expert supervision.

- Adding pollutant emission calculation and optimization function (to the WDS pump operation optimization.
- A modular design of the optimization tool that allows each part to be replaced with alternative codes for future studies to improve its efficiency without the need to create a completely new optimization tool from scratch.

It was mentioned in Chapter 1 that a few recent studies have used metamodeling or parallel computation to decrease required time of pump operation optimization. Also in some studies, reducing the pollutant emission and environmental footprint of pump operation plan have briefly been studied. But the author could not find any studies that unify all above mentioned ideas and considers pollutant emission optimization besides energy usage optimization while the required time of optimization is reduced by using metamodeling methods. Moreover, in previous studies, there is a relatively low emphasis on developing an optimization method and tool that can generate the practical type of result which needs minimum edits by experts, before getting used in the real operation plan. As a result, not even experimental optimization tools, but also commercial pump operation optimization tools in the market are not able to generate practical outputs for the operational plan of medium and large size WDSs within a reasonable time. So one of the practical advantages of this study is taking one step toward developing a completely automate, real-time and online tool that can optimize the operation process of WDS based on a couple of economic and environmental objectives. Another practical advantage of this dissertation is its realistic point of view regarding the pump operation optimization needs in the market and trying to take one step toward creating an optimization tool that can be accepted by operators of WDS and decrease reliance on the individual human judgment in real WDS operation. The theoretical advantage of this study is preparing a fast and modular multiobjective optimization platform that each part of its structure can be changed in future to improve its efficiency and practicality. Also, this study tries to prove that it is practically possible to optimize the operation of pumps of a WDS and decrease energy consumption cost while the related pollutant emission is also decreased.

## 2.2. Tool Development

As part of the present investigation, a computer program has been developed and tested for application to the problem of optimizing pump operation of WDSs. This computer program is named PEPSO, which stands for Pollutant Emission Pump Station Optimization. In this section, all components of PEPSO are introduced, and their functionalities are explained.

Visual Basic (VB 11.0) programming language was used along with Microsoft .Net Framework 4.5 to develop PEPSO as a modular software with graphical user interface. Modular structure makes modifying and future enhancement of PEPSO easier for developers. PEPSO also is designed with a multithreading structure that uses the capability of multi-processor computer systems to speed up some part of the optimization process. PEPSO uses the multithreading capability to separate the optimization calculation from other time consuming graphical process of the user interface. In addition to 16000 lines of code that are written, PEPSO uses code libraries of EPANET Toolkit V2.0.12 (Rossman 1999, Rossman 2008) for hydraulic simulation, FANN V2.2.0 (Nissen 2003, Nissen 2003) for ANN training and MATLAB Runtime V8.5.0 (MathWorks 2015) for 3D plotting.

### 2.2.1. Graphical User Interface

PEPSO was aimed to work as a user-friendly software that WDS designers and operators with an average knowledge of hydraulic and WDS operation can use.

Therefore, the inclusion of a strong graphical user interface (GUI) was a key element in PEPSO's development. In order to achieve this aim, the following detail's were considered in design of the GUI (Wikipedia-contributors 2015):

- Clarity and Concision: All different types of elements like labels, icons, and colorful plots were used to make the interface as clear as possible for the user. Other elements like tabs, tables, boxes and borders were used to separate and categorize input and output sections and create a tedious interface that is clear and concise at the same time.
- Familiarity: Windows user interface guidelines were used to design a familiar interface for even those users who use PEPSO for the first time (MSDN 2015). Moreover, logical color coding and standard signs and icons were used to provide a familiar interface to all users.
- Responsiveness and Efficiency: Various shortcuts and menus were used to make importing data, defining and running the project and receiving results as efficient as possible. Also, it was tried to reduce required time for retrieving data from saved or downloaded files
- Consistency: The whole interface has been designed based on a comprehensive logic. Therefore, after working with the first form, the user can implement the learned logic for interacting with other forms.
- Aesthetics: By selecting appropriate size and type of component of interface and using colors and shapes it was tried to make the interface like modern software that is attractive to user's eyes.

- Forgiveness: Various exception handling methods with a fully explained warning and error messages were provided to prevent the process from crashing due to a bad input or user interaction.

PEPSO has seven major forms that allow the user to define an optimization project and execute it. Figure 6 illustrates the process flow enabled by PEPSO's interface. Each of the steps and related forms is explained in the following section.


Figure 6- PEPSO Flowchart

### 2.2.1.1. Main Form

The Main form is the first form that appears on the user's screen upon execution of PEPSO. It provides access to all forms via menus and tool strip. It also shows a summary of all defined or loaded project information. During the optimization run, the main form provides run-time information and statistics of the optimization process. Figure 7 provides a screenshot of the main form that is displayed at one point for a test simulation.

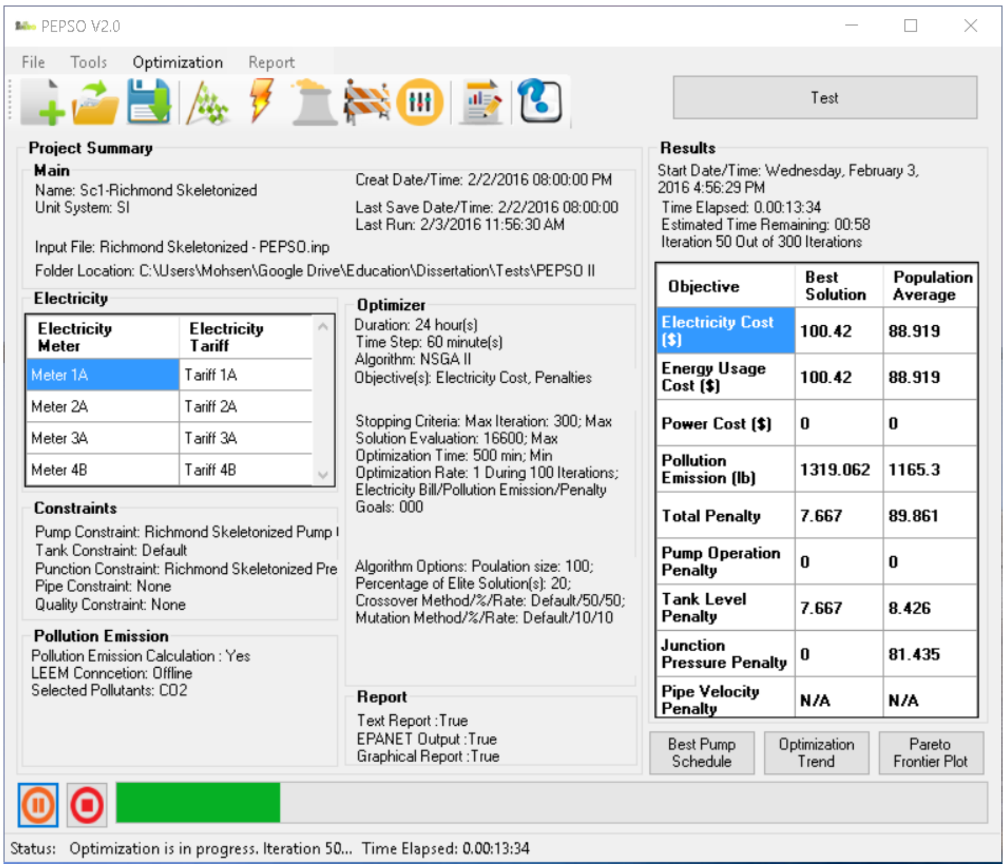



Figure 7- Main form of PEPSO

### 2.2.1.2. Project Configuration Form

The project configuration form is the initial point for defining a new project. It also can be used for changing some basic information of a loaded project, such as name, project folder address, and hydraulic model file address. Through buttons of this form, users have access to all other forms for adjusting project parameters before running the optimization process (see Figure 8).

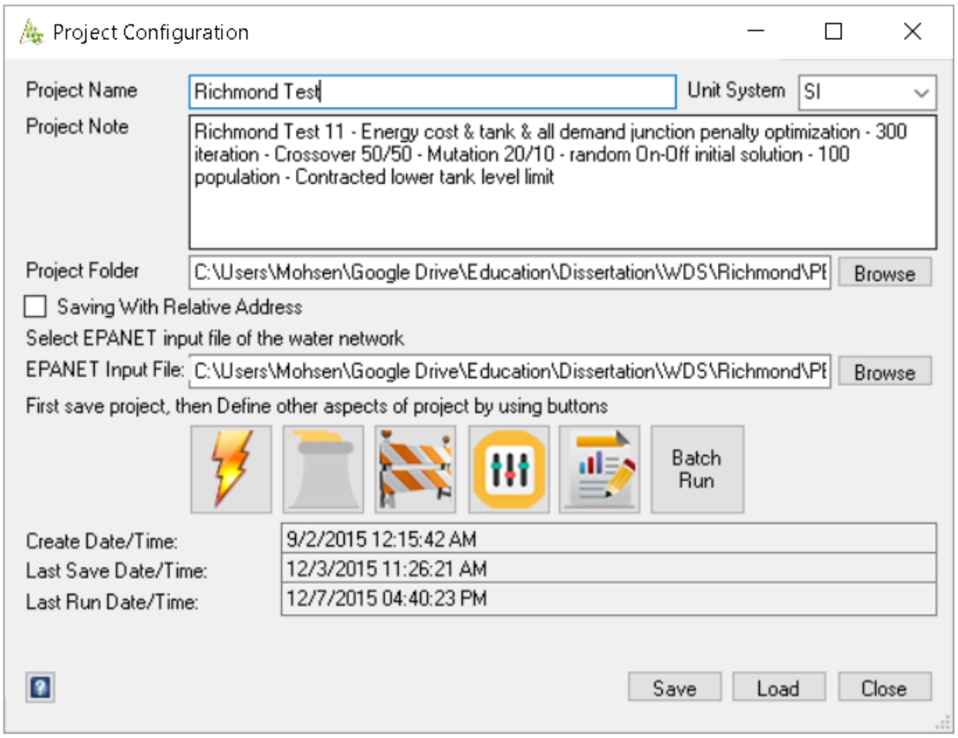



Figure 8- Project Configuration form of PEPSO

After initializing a project using the configuration form, a suite of additional forms can be accessed to further define the project. These additional forms include the electricity, pollution emission, constraints, optimization, and report options form. All of these forms have been designed with the same logic to create a consistent user experience. This ensures that multiple scenarios of electricity tariff, pollution emission, pump, tank, junction and pipe constraints and optimization options can be defined, saved, loaded and selected as an active scenario by using the same logic.

#### 2.2.1.3. Electricity Form

Most of the industrial electricity tariffs have two parts: a) *energy consumption charge* and b) *power demand charge*. The *energy consumption charge* ($/kWh) should be multiplied by the amount of consumed energy (kWh) to calculate energy consumption cost ($). Similarly, *power demand charge* ($/kW) should be multiplied by *peak power demand* to calculate the power demand cost ($). The *peak power demand* of an electricity meter can be calculated as a maximum power demand of the electricity meter during a defined billing period (e.g. one month) that is measured in a defined time intervals (e.g. 30 minutes intervals). Total electricity cost is electricity consumption cost and power demand cost of all electricity meters.

The electricity from has two tabs.Users can input various types of electricity tariffs in the first tab. The tab can accommodate electricity tariffs that have a constant rate *energy consumption charge*, as well as time-variant rates ($/kWh). Also, *power demand charge* ($/kW) and duration and intervals of calculating *peak power demand* can be defined via this tab. Note that it is possible to define and use multiple electricity tariffs in an optimization scenario for different electricity meters. However, each electricity meter can have only one electricity tariff.

After defining, at least, one tariff, the second tab can be accessed to define electricity meters and assign the defined tariff to them. Most of the time a pump station has one electricity meter.However, it is possible to define multiple electricity meters for pumps that are physically located in one pump station. Each electricity meter should have a list of the connected pumps. *Peak power demand* and energy consumption of the connected pumps to an electricity meter will be added up before calculating the electricity cost. Note that each pump can be connected to only one meter. Figure 9 shows tariff (left) and meter (right) tabs of the electricity form. Latitude and longitude of electricity meter are necessary input parameters if the user plans to use the pollution emission calculation or optimization routines. Location of the electricity meter will be used to retrieve the emission factor values from the LEEM subroutines.

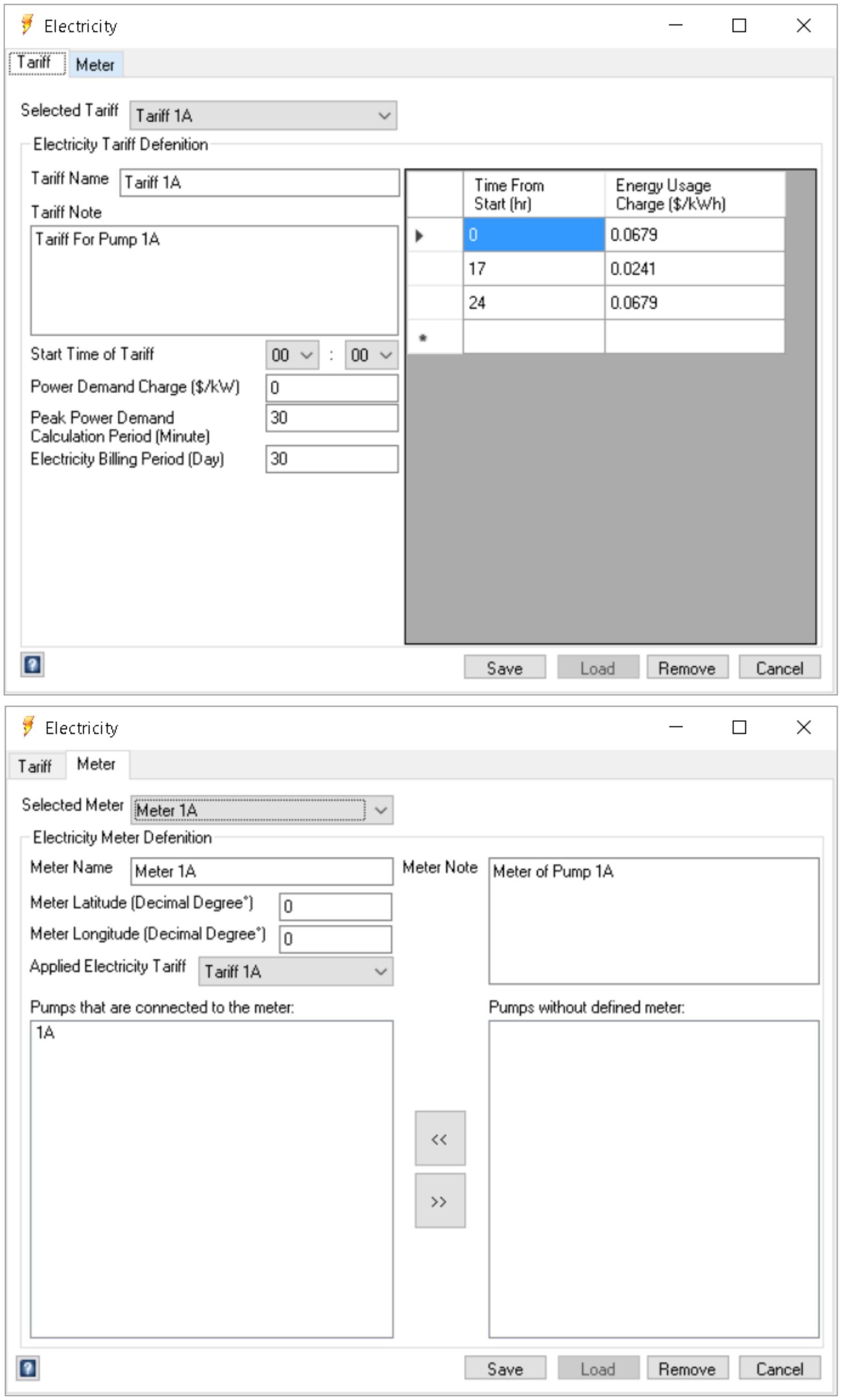



Figure 9- Electricity tariff (Top) and electricity meter (Bottom) tabs of the electricity form of PEPSO

### 2.2.1.4. Pollution Emission Form

One of the unique characteristics of PEPSO in comparison with other WDS operation optimization tools is its ability to use the emission factor report of LEEM to enable real-time spatially-explicit emission reduction optimization. The pollution emission form is the interface for user-defined pollution emission calculation scenarios. Each scenario may include one pollutant or a user-defined pollution index that is a linear combination of multiple pollutants. Users can elect to receive emission factor values from the LEEM server via internet or use an offline LEEM report. The offline LEEM report option is useful when the user wants to compare results of different optimization runs and wants to prevent unwanted changes due to receiving different reports from LEEM during different optimization runs (due to the time-sensitive nature of the emission factors). Figure 10 provides a screenshot of the pollution emission form.

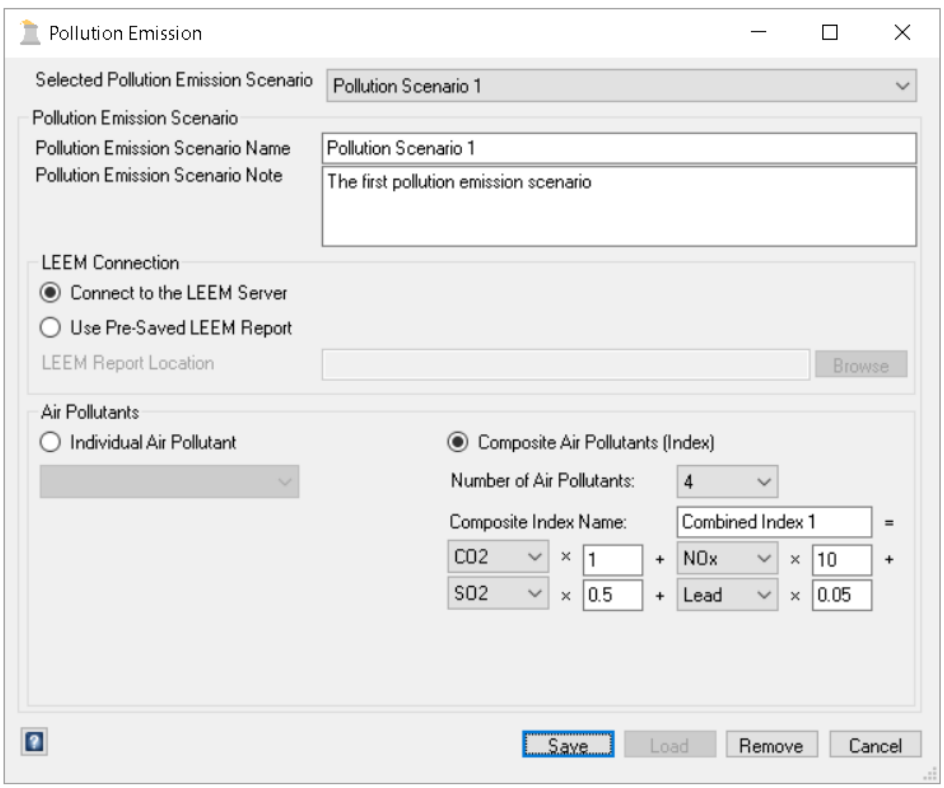



Figure 10- Pollution emission form of PEPSO

It should be noted that only those pollutants that their information can be obtained from the LEEM server or an offline report can be selected in this form. Currently, LEEM 2.5 server provides emission factor (lb/kWh) of five pollutants ($CO_2$, $NO_x$, $SO_2$, Hg, and Pb). PEPSO uses the user-specified location of each electricity meter (from the electricity meter tab of electricity form) and time of optimization, to query emission factors from LEEM server.

#### 2.2.1.5. Constraints Form

The constraints form has four tabs that allow users to define customized constraint scenarios for pumps, tanks, junctions, and pipes. It also is possible to select the default constraint scenario that PEPSO automatically defines based on characteristics of the WDS model. Although it is not recommended, it is possible to turn off constraint scenarios, allowing network optimization in the absence of any constraint on the operation of pumps, the water level in tanks, pressures of junctions or water velocity.

As shown in Figure 11 (top), the first pump tab of the constraints form allows the user to define whether a pump is a variable speed or fixed speed pump. For a variable speed pump, the user can input the pump's minimum possible *relative rotational speed* (RRS). RRS is a number between 0 and one that 0 means the pump is off, and 1 means the pump is working with its maximum rotational speed. Based on the pump affinity law, the power demand of a pump is directly proportional to the cube of the RRS (Pelikan 2009). When RRS of a pump is 0.5, it only can push water with ($0.5^3$) = 12.5% of its nominal power, so it is not practical to use a number less than 0.5 as the minimum RRS of the pump. Note that the maximum RRS of all pumps is considered 1 (100% of the maximum rotational speed of the pump). Other constraints for pump operation include a) a maximum number of switches in a day, b) a minimum duration of time between pump shut-down and start-up, and c) maximum continuous period of operation for the pump. When a pump operation schedule violates these limits, a penalty will be calculated and added to the total penalty of the pump schedule.

Users can define these limits as hard constraints or not. This means that in addition to calculating penalties, these limits can be used for defining an acceptable or unacceptable pump schedule. If the user decided to define these limits as hard constraints, violation from them completely discredits the pump schedule from being selected as the optimum final result. PEPSO will not use these hard constraints during the optimization process. However at the end when a solution should be selected among the Pareto frontier as the optimum solution, these hard limits will help to filter out unacceptable solutions. Using a hard constraint during the optimization may limit the ability of PEPSO to explore the solution space for the optimum solution.

In the constraints form for the tank (Figure 11 bottom), users can define allowed and desired minimum and maximum level of water in the tanks. Note that the minimum and maximum tank levels that are defined in the EPANET model file are physical limits and EPANET does not let water level to go beyond these limits. However, the desired minimum and maximum levels that are defined by users are soft constraints. Water level can go beyond desired limits, but this violation causes some penalties. By default, the minimum and maximum desired water level constraints of a tank are 15% higher and lower than the bottom and top level of the tank, respectively. The minimum and maximum allowed water level that can be defined by users are hard constraints and like hard constraints of the pump operation tab, will not be used during the optimization process. However at the end of the process, they will help to filter out all unacceptable solutions from the final Pareto frontier. By default, the minimum and maximum allowed tank levels are equal to the minimum and maximum tank levels of the EPANET hydraulic model, respectively.

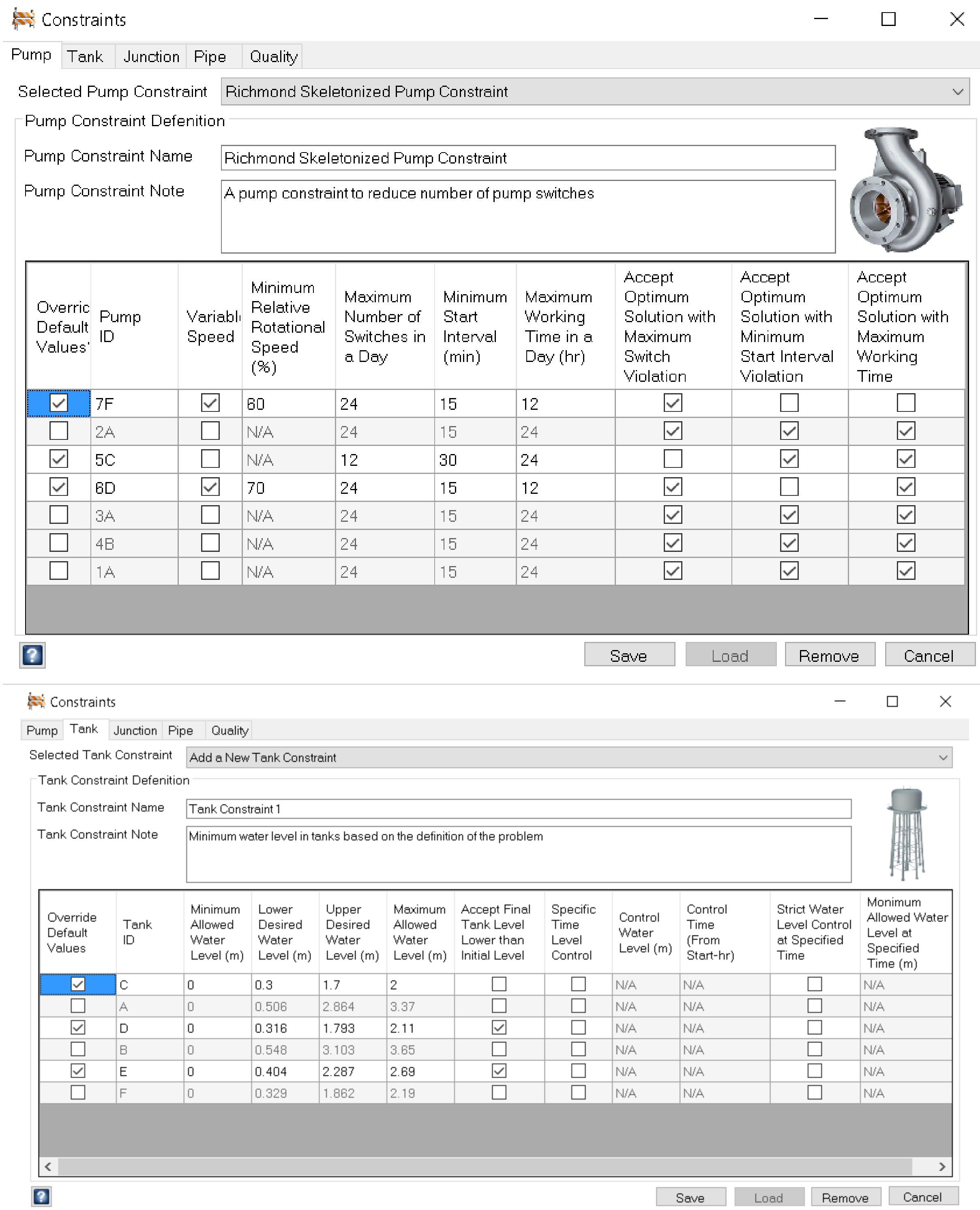
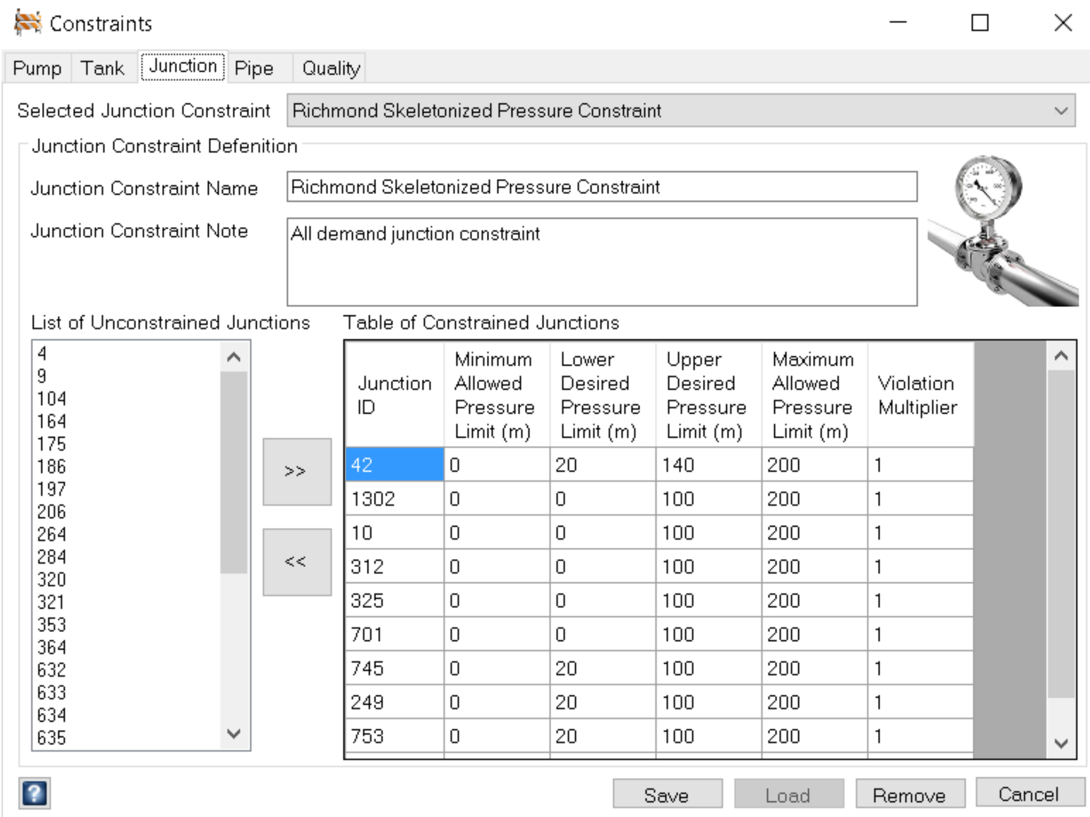



Figure 11- Pump constraint (top), tank level constraint (bottom) tabs of constraint form of PEPSO



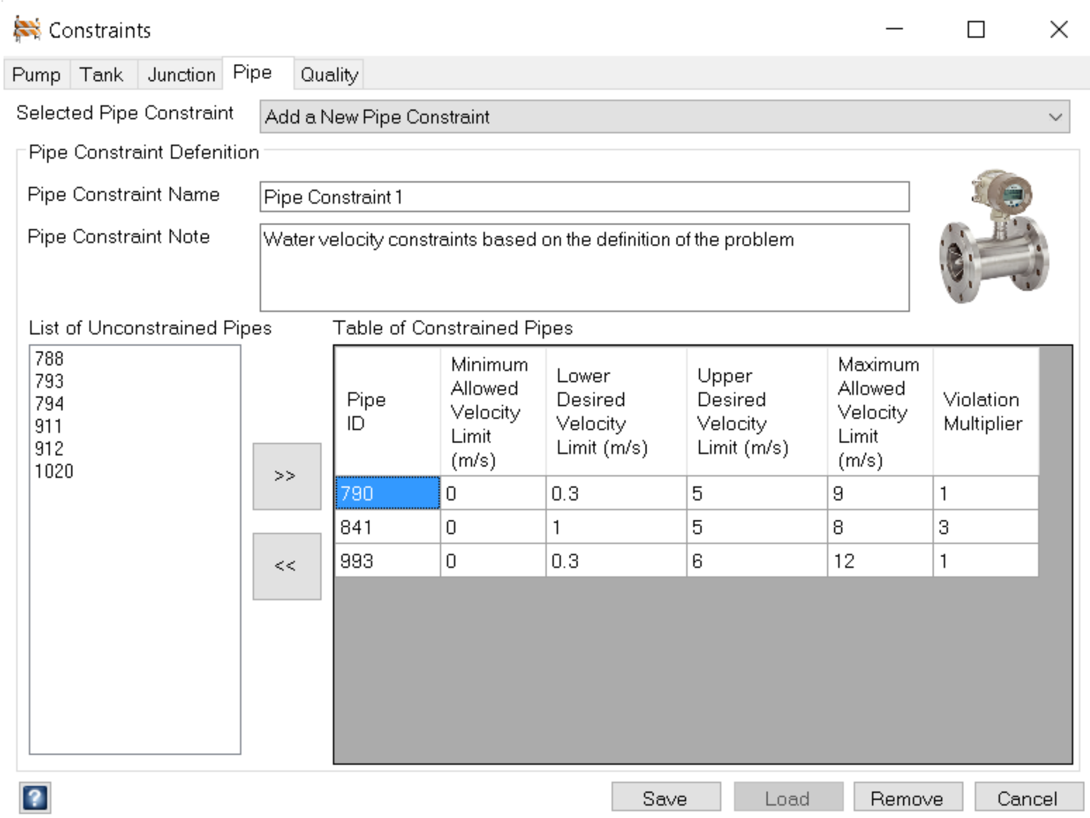




Figure 12- Junction pressure constraint (top) and water velocity constraint (bottom) tabs of constraints form of PEPSO

Five last columns of the tank constraint table can be used for constraining water level in the tank at a specific time. For instance, if the operational requirement of a WDS dictates that a tank should be 50% full at 7:00 AM, this part of the table can be used to define constraint water level and time. Like desired minimum and maximum level, this is a soft constraint and will be used only for calculating penalties. However, if users select the “strict water level control at specific time” option, it will be used as a hard constraint for filtering out the unacceptable solution at the end.

The junction and pipe tabs of the constraints form that are shown in Figure 12 (top and bottom respectively) allow the user to select strategic junctions and pipes from the list of all junctions and pipes of the WDS and assign the minimum and maximum allowed and desired pressure or velocity limits to each of them. It also is possible to indicate the relative importance of each junction or pipe in respect to others by defining the *constraint importance multiplier*. By default, these multipliers are equal to one for all junction and pipes, resulting in equivalent penalty associated with the violation of pressure or velocity limits of all selected junction and pipes. However, changing the *constraint importance multiplier* of a junction increases the penalty associated with pressure violation of that junction with respect to others. Like the tank level constraints, the desired limits define soft constraints. The pressure or velocity violation from these limits increase the calculated penalty. It is important to know that violation from each of the minimum and maximum limits of water level in the tank, water pressure at junction or water velocity has different meaning and PEPSO stores these violations separately. PEPSO will use them separately to discover promising ways of changing the pump schedule for improved results. The pressure and velocity allowable limits are stricter hard constraints and will just be used for filtering out acceptable solutions from the final Pareto frontier at the end of the optimization process.

#### 2.2.1.6. Optimization Options Form

Users can open the first tab of optimization options form to define optimization algorithm parameters and objective functions (see Figure 13, top). In the upper part of this tab, three objectives of optimization (electricity cost, pollution emission, and penalties) can be selected. Electricity cost is composed of energy consumption cost and power demand cost ($). Pollution emission is the weight of a single emitted pollutant (lb) or values of the user-defined pollution emission index. Lastly, the penalty value is total penalty formed from pump operation constraint violations, tank level violation, pressure violation and velocity violation. Here users also can define relative weights of each selected objective. This weight will not be used during multi-objective optimization process of PEPSO that optimizes each objective independently. However, at the end of the process and before reporting the final optimum solution, it will be used to select the optimum pump schedule among all the acceptable solutions in the final Pareto frontier.

The middle section of the options tab defines stopping criteria. Optimization can be stopped under any of 5 user-defined conditions: (1) elapsed computation time, (2) the maximum number of iterations, (3) the maximum number of or solution evaluations, (4) when a set of predefined objectives is reached, or (5) a maximum number of stagnant iterations. The bottom section of this tab gives users some options to adjust the optimization algorithm options and hydraulic simulation method. For instance for NSGA II optimization method, users can define the number of solutions in each population, *crossover* and *mutation* percentage and rate, and the number of elite solutions of each population. It also is possible to select EPANET or ANN model for hydraulic simulation.

The *crossover* and *mutation percentages* define the portion of the population that should be used in *crossover* (reproduction) process or should be mutated, respectively.

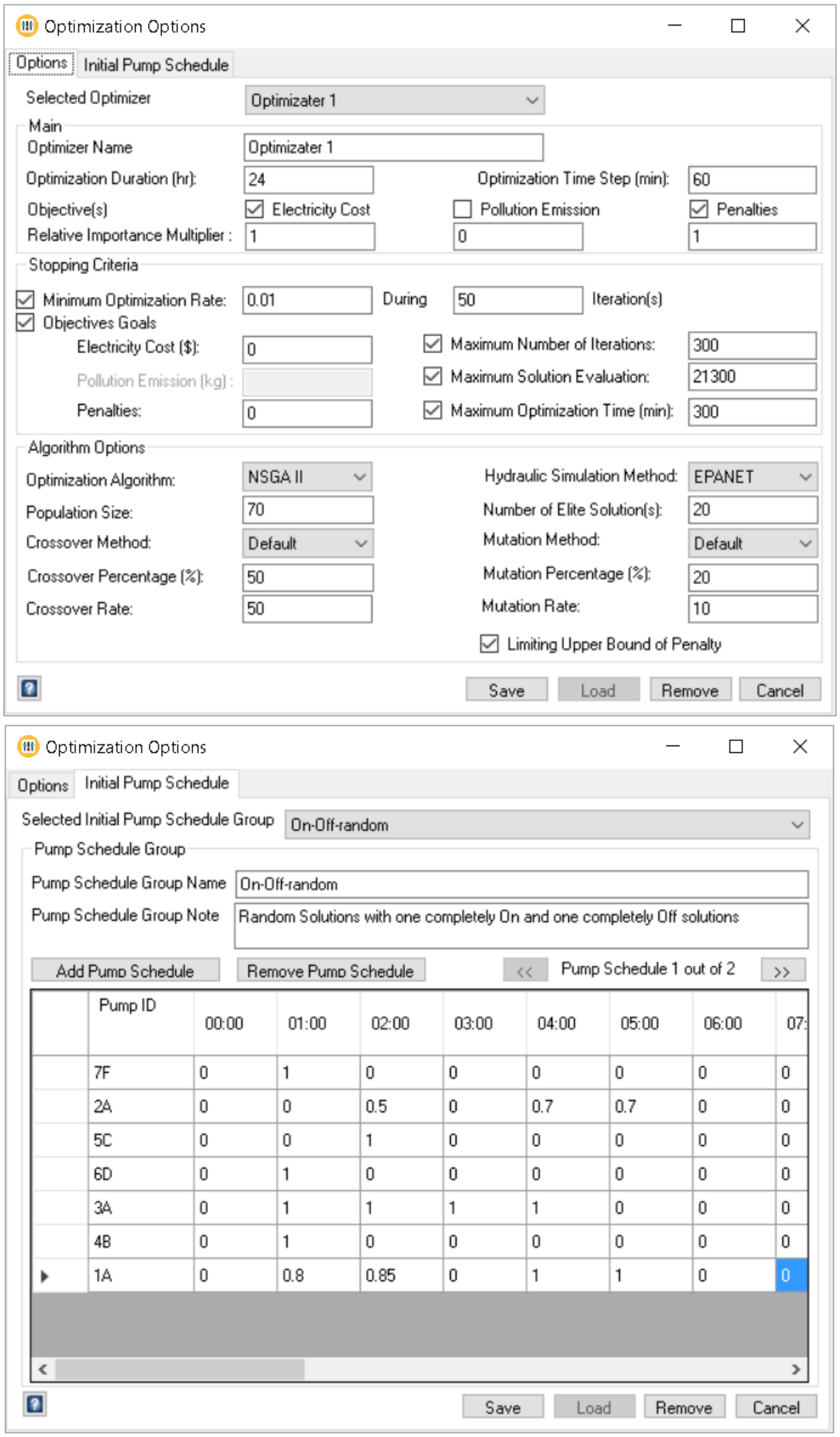



Figure 13- Optimization option (top) and initial pump schedule (bottom) tabs of the optimization options form of PEPSO

The *crossover* and *mutation rates* define the portion of a selected pump schedule which should be changed during *crossover* and *mutation* processes, respectively. By default, both the *crossover percentage* and rate are 50%. The *mutation percentage* and rate by default are 5% and 10%, respectively. High *mutation* and *crossover rates* may change the selected pump schedule drastically that may aid PEPSO's exploration of the solution, but may also decrease the efficiency of exploitation process and fine tuning the near-optimum solution.

The second tab of the optimization options form that is shown in Figure 13 (bottom) helps users to customize start point of optimization and define an initial population of solutions. By default, PEPSO forms the initial population by a group of randomly created pump schedules. It also adds two extreme pump schedules to the population to catch two extreme points of solution space. In one of those two extreme pump schedules, all pumps are off and in the other one, all pump are on. However, in addition to the default initial population, users can define some initial pump schedules and use them to replace all or part of the initial random population. This option is especially useful when it is desirable to initiate an optimization run from the result of a previous optimization run. It also can be used for comparing different optimization scenarios when users want to keep the initial population of all scenarios the same.

#### 2.2.1.7. Reporting Option Form

The reporting options form provides all options that users need to customize reports of PEPSO. The top section of this form allows users to select different types of reports that should be included in the text output. A field in the middle section of the form is provided to define the file name for the optimized EPANET model. The bottom section of the form shows all options for customizing the graphical report. Users can select different types of graphical reports, their updating frequency, and detailed adjustments about label or scale of axes of the graphs. Figure 14 shows a snapshot of this form. The text report, “Richmond Test_Optimized.inp”, with optimized model and graphs, will be saved in the project folder. Users can select to show (during optimization) and save a) the best practical pump schedule, b) the optimization objectives trends, and c) the 3D Pareto frontier graph at defined iteration intervals.

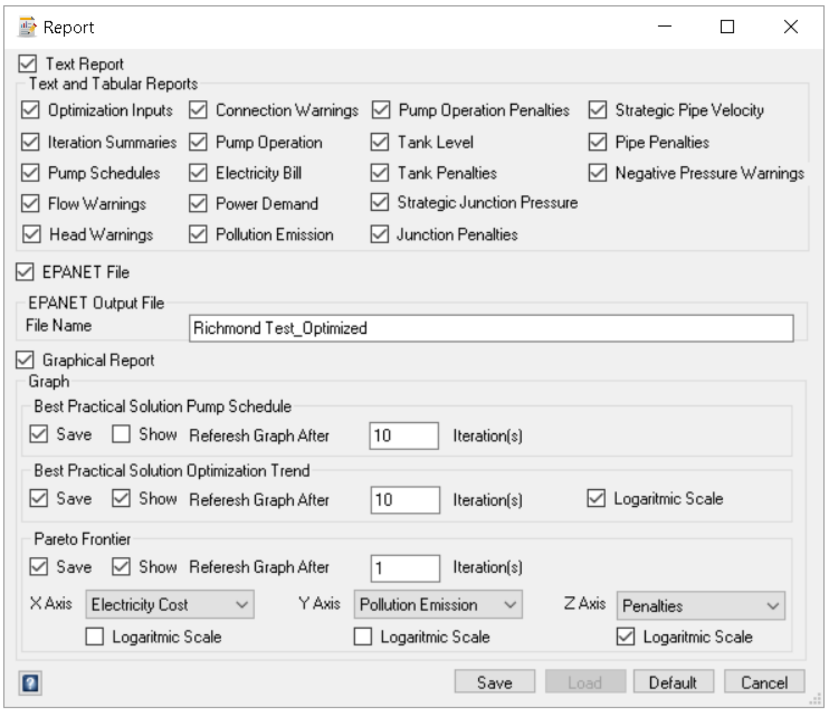



Figure 14- Report option form of PEPSO

### 2.2.2. Optimizer

After defining the optimization project and saving the project file, the user can press the run button to start the optimization process. The optimization process can be broken down into three main phases: a) pre-optimization b) iterative optimization and c) post-optimization (finalizing and reporting). In the pre-optimization phase, the optimization run will be initialized and the initial population will be created, evaluated and edited. Flow chart of Figure 15 shows the pre-optimization process.

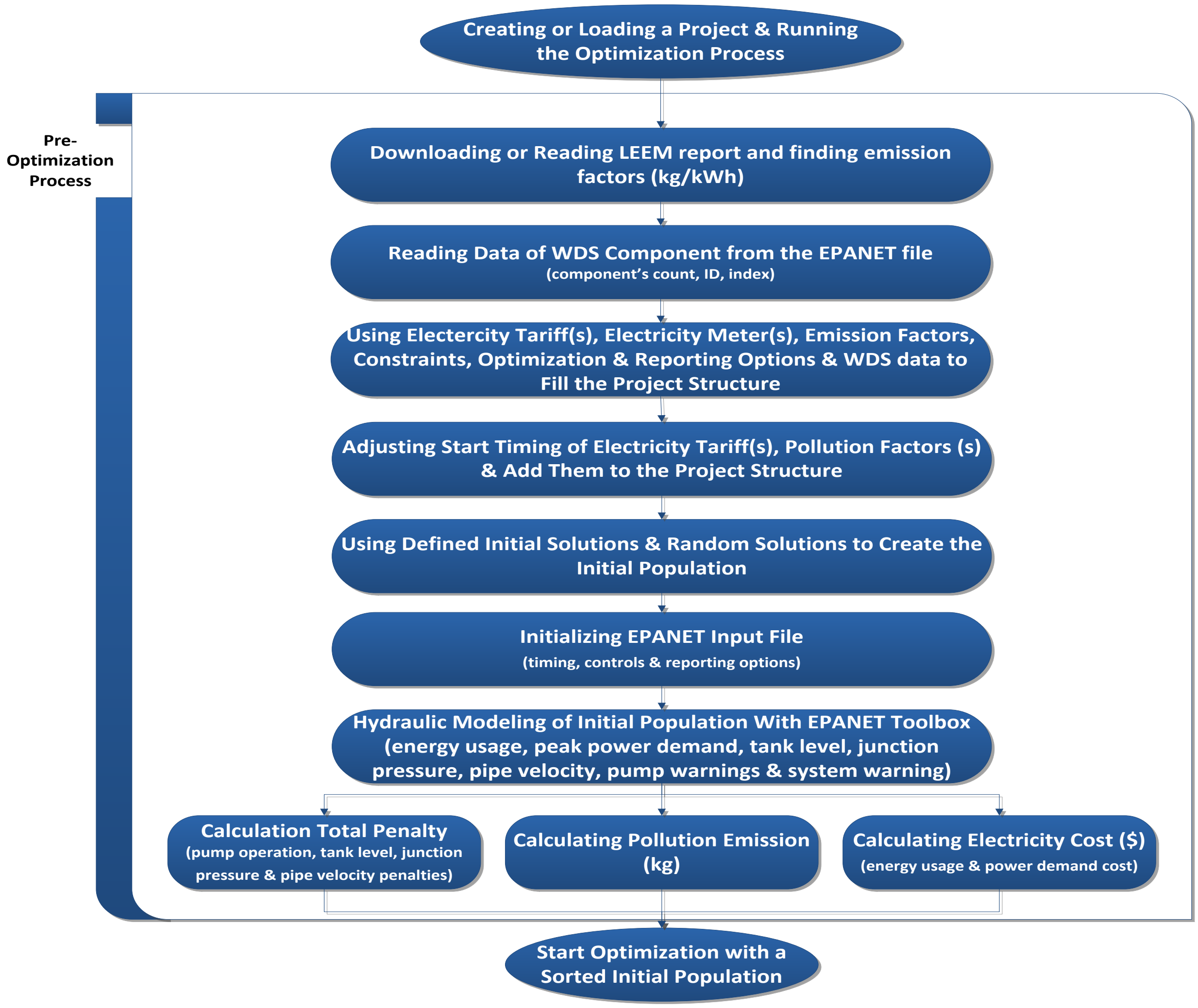



Figure 15- Flowchart of pre-optimization process

In the iterative optimization phase, *crossover* and *mutation* steps will be used to generate better solutions. Then new solutions will be evaluated, and a new generation will be formed from the available elite solutions. This process will be repeated until meeting a stopping criterion. After stopping the iterative process, the program starts the post-optimization phase. In this final phase, PEPSO selects the best solution and prepares and stores all requested reports in the format of text, EPANET model, and graphics.

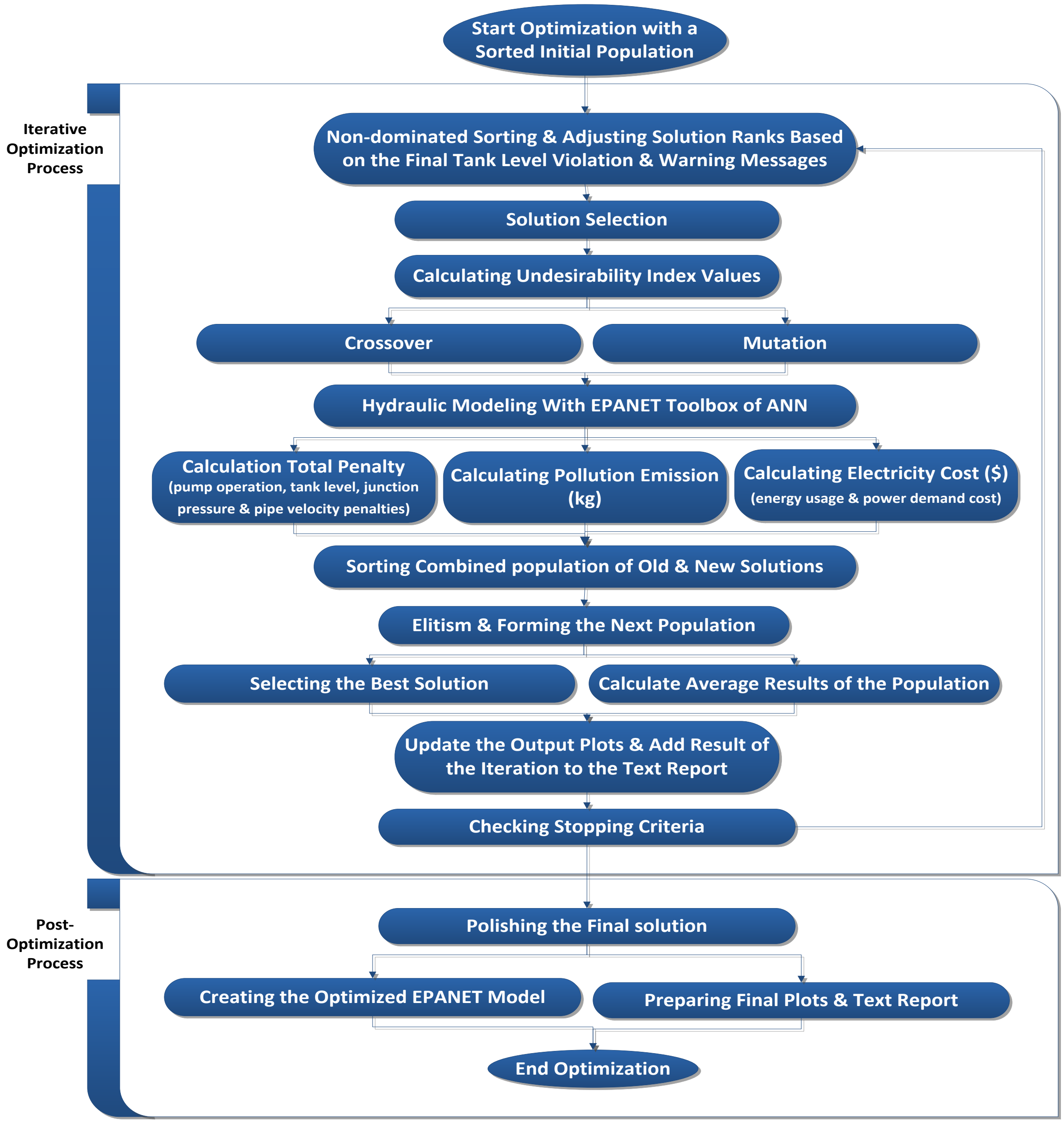



Figure 16- Flowchart of iterative optimization and post optimization processes

The flowchart of Figure 16 shows different processes of the iterative optimization and post-optimization phases. The section 2.2.2 and its subsections explain main modules that are used in these three phases of optimization. Hydraulic solver and output reporter modules that are primarily used in the second and third phase will be explained separately in sections 2.2.3, 2.2.4 and their subsections.

### 2.2.2.1. Optimization Initializer

The optimization initializer module of PEPSO reads all required data from the project file and stores it in a suitable structure for optimization. It reads and stores all information that is provided through the forms that were introduced in Section2.2.1, in addition to information of WDS that is provided by the EPANET hydraulic model and emission factors that are provided by the LEEM report. A copy of all imported data into the optimization data structure will be written into the first section of the optimization text report. It helps users to better understand characteristics of the optimization run when they are using the final text report. The optimization project data structure keeps the address of the project folder and name of all input and output files. By default the project folder stores the project file (*.prj), ANN models of the WDS (*.net), text output (*.txt), graphical outputs (*.fig and *.jpg) and EPANET optimized model (*.inp).

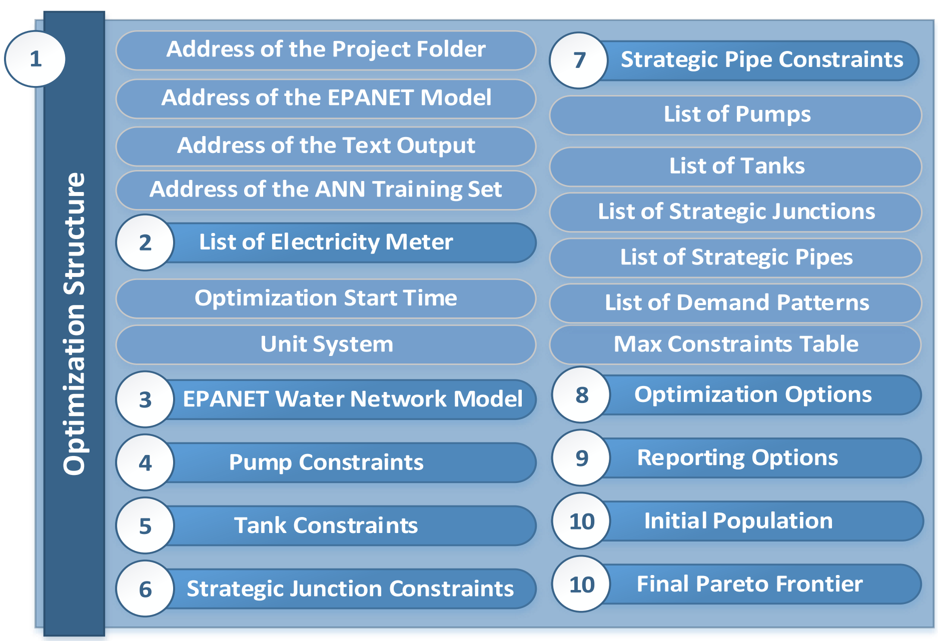



Figure 17- Diagram of the optimization data structure (for more information see Appendix B)

As previously mentioned, the optimization structure also read and stores information of the EPANET hydraulic model, including the number of WDS components, pattern, report and start times, and unit system that is used in the EPANET hydraulic model. Figure 17 shows the main diagram of the optimization data structure. All numbered items in this diagram are sub-structures that have been expanded in Appendix B.

All inputs, including water demand pattern, pump status, electricity tariff, emission factors and all outputs in the form of time series must have the same time reference, time step and duration as the optimization time reference, time step and duration. The optimization initializer module uses the start time of optimization, optimization time step, hydraulic model time step, hydraulic pattern start time and time steps, and hydraulic simulation start time to adjust the timing of all inputted energy cost and emission factor patterns and prepares them for use in the optimization process.

In the current version of PEPSO, the optimization duration and time step, which are defined by users via the optimization options form, dictate duration and time step of all the above mentioned time series. However, the start clock-time of the EPANET hydraulic model defines the time reference and start point of the time series, including the pump schedule. If the duration of the defined energy cost pattern is less than the duration of optimization, the same pattern will be repeated to cover the whole optimization duration. In the case of the emission factor, this correction is a little bit more complicated. LEEM 2.5 currently provides between 6 to 37 hours of emission factor prediction (based on time and location of data query). Therefore, if the duration of predicted emission factors data is shorter than the optimization duration, emission factors of the same clock time of the previous day will be used to fill the lack of predicted emission data. It is assumed that emission factors of the previous day are not considerably different from emission factors of the same clock time of the next day and are acceptable candidates to fill the lack of prediction data without changing the final result drastically. However, errors are expected especially when this assumption covers days that are holidays, weekends, drastic temperature/climate changes, and so on.

If the optimization time step is larger than energy cost pattern or emission factor time steps, all energy cost and emission factor values that fall into an optimization time step will be averaged based on their contribution time. For instance if the optimization time step is one hour and for 15 minutes of an hour energy price is 0.1 ($/kWh) and for the other 45 minutes is 0.16 ($/kWh) the weighted average energy price of that time step is (15 / 60) × 0.1 + (45 / 60) × 0.16 = 0.145 ($/kWh). If the optimization time step is shorter than the energy cost pattern or emission factor time steps, these will be broken into the smaller time steps with equal length to the optimization time step. For instance, assume that the optimization time step is three hours and emission factor of $CO_2$ at the first hour is 1.5 (lb/MWh) and for the next two hours is 0.9 (lb/MWh). The emission factor value of that three hour time block is calculated as: (1 / 3) × 1.5 + (2 / 3) × 0.9 = 1.1 (lb/MWh). Sometimes it may happen that LEEM cannot provide an emission factor value for a time step. In this case, PEPSO fills the missing value with average emission factor during the optimization period.

All PEPSO calculations use the time unit of seconds. So all non-second input time values will be converted to second. All requested non-second output values will also be converted before reporting. In addition to the time unit, all other physical units of PEPSO calculations are SI units. Although it is possible for users to use the US customary unit system for inputting and receiving outputs, PEPSO converts these units to SI system, before using in the calculation and converts the result back again to the US customary before reporting. Pressure unit in PEPSO calculation is the meter of water head, and discharge unit is cubic meter per second. However, for input data, PEPSO accepts all units that are accepted by EPANET V2.0.12 (Rossman 2000).

The optimization initializer module also creates an initial population of solutions and appends it to the optimization structure. This function that creates the initial population is explained in the following section.

#### 2.2.2.2. EPANET Input Initializer

In the pre-optimization phase, PEPSO initializes the EPANET model of WDS. This initialization process prevents some potential error caused during the optimization process and provides a standard format for simulating and reporting by EPANET. In this process, at first, water demand pattern of junctions will be replaced with modified demand pattern that has the same duration and time step that is defined by the user. After this, start time of patterns, pattern time, reporting time step and the hydraulic time step of the EPANET file will be adjusted based on the optimization run requirements. Report status option of the EPANET file will be changed to “YES” to make sure that EPANET simulation report has all required information (including warning messages). By default, the number of hydraulic simulation trials and accuracy of convergence will be changed to 40 and 0.001, respectively. The lower trial number and accuracy may result in a faster hydraulic simulation but may increase the probability of receiving a system unbalanced warning and decrease the accuracy of calculation that might affect the efficiency of optimization.

The goal of PEPSO optimization is a determination of an optimized pump schedule. So PEPSO should start the optimization process with a hydraulic file without any predefined pump controls, rules, and operation patterns and find the best set of pump controls to meet the objectives. Therefore, before starting the optimization process, all pump rules and controls of the input EPANET model should be removed, and initial status of all pumps should be reset to off (by default). Similarly, all variable speed pump patterns should also be removed. To be able to change pump schedule of the EPANET file during the optimization process, some initial pump control lines are needed in the control section of the EPANET input file. Number of these controls should be equal to the number of pump schedule cells (number of pumps × number of time steps). So an initial and pump control line will be repeated by the number of pump schedule cells and added to the control section of the EPANET input file.

These initial control lines are only placeholders of pump control lines that will be created by PEPSO during the optimization process. These initial control lines should not have any effect on the operation of pumps, so the EPANET initializer module of PEPSO creates them by using EPANET pump ID of the first pump that is off at one time step after the final time step of the optimization process. For instance, if we are going to optimize a pump schedule for a 24 hour period with one hour time intervals, "PUMP 1 IS OFF AT TIME 25" can be the placeholder pump control lines. This definition for the placeholder pump control line shows that this initial control will not have any effect on optimization results and is created for filling the required lines of the control section of the EPANET input file.

After making all these changes on the EPANET input file, the file will temporarily be saved for the optimization purpose. At the end of the optimization process, this file will be overwritten with the optimum solution.

#### 2.2.2.3. Initial Population Creator

As explained in section 2.2.1.6, users can choose to start the optimization process from a randomly created population of solutions, or they can define a population (completely or partially) as a starting point. For creating a random pump schedule, the status of the constant speed pumps at each time step will be changed to on or off randomly. For variable speed pumps, on or off status of the pump will be defined in a similar way. However, for the “on” variable speed pumps, the RSS will be selected as a random number between the minimum RRS and 1 (100% as the maximum relative speed). The RRS values will always be rounded to two decimal places (which results in a 1% accuracy in rotational speed).

Within the optimization initializer module, the maximum possible junction pressure penalties that will be used later during the optimization steps are calculated. Two extreme conditions that may result in maximum junction pressure penalties are 1) turning all pumps off (to create maximum low-pressure violation) and 2) turning all pumps on (to create maximum high-pressure violation). PEPSO automatically adds these two extreme pump schedules to the initial population to include results of these two extreme conditions. So, the number of initial solutions that can be defined by users is equal to the size of the initial population minus two.

#### 2.2.2.4. Objective Calculators

Three separate modules of PEPSO are used to calculate independently three objective values: electricity cost, pollution emission and total penalty of each solution. Before calculating objective values, solutions should be simulated hydraulically and the energy consumption, *peak power demand*, water level in tanks, water pressure and velocity at junctions and pipes and pump schedule characteristics (e.g. number of pump switches) at all time steps are stored in a temporary file.

The electricity cost calculator module receives total energy consumption of all the pumps that are connected to a meter at each optimization time step. The calculator then multiplies the energy consumption value with the corresponding energy consumption charge to calculate the energy consumption cost associated with that electricity meter. It similarly uses the *peak power demand* of all pumps that are connected to the meter to calculate associated power demand cost.

The pollution emission calculator module uses the energy consumption of each electricity meter at each optimization time step. It multiplies the energy consumption value (kWh) by the corresponding emission factor (lb/kWh) to calculate the emission pollution weight (lb). It should be noted that values of emission factors depend on the time of energy consumption and location of the electricity meter (pump station). LEEM reports the marginal emission factor which is equal to the amount of pollution emission due to one unit increase in energy consumption of the whole region. We cannot multiply the total energy usage in the region by marginal emission factor to calculate the pollution emission of the region. For this purpose, we need to use pollution emission data of the all the energy generators that provide energy of the region (not only the marginal generator). However, it is justifiable to assume that the total energy consumption of a WDS is relatively negligible in comparison with the total energy consumption of a region and is not able to change the marginal generator. In this case, the marginal emission factor can directly be multiplied by the total energy consumption of the WDS to calculate its pollution emission. Theoretically the resulted emission value is not equal to the total pollution emission of the real system. However, the calculated emission value by this method (by using marginal emission factors) can be used for comparing different operational scenarios. The difference between resulted emission values of scenarios shows change of the total emission of the real system due to change of the operational scenario.

The third module calculates associated penalties of each solution (pump schedule). Based on the user request, the total penalty may include pump operation penalty, tank level penalty, junction pressure penalty and velocity penalty. So it is possible for users to turn off a constraint and the associated penalty will not be calculated and will not affect the optimization process. For each time step, the penalty is calculated as a violation value of the parameter raised to a predefined factor. By default, the factor is two. This helps to amplify the importance of the deviation from the acceptable range as the deviation increases. For instance, if the acceptable range of tank level is from 1 to 5.5 meter, and if the tank level goes up to 7 meters the violation can be calculated as $|7 - 5.5| = 1.5$ meter and the penalty is $1.5^2 = 2.25$. If, at another time step, tank level is 0 meter the violation is $|0 - 1| = 1$ meter and penalty is $1^2 = 1$. It can be seen that although the violation of the first case is 1,5 times more than the violation of the second case, the penalty of the first case is 2.25 times more that the second case. This way of using the power factor to increase penalty when the violation is large, help ensure that PEPSO will recognize unfeasible solutions. For instance, if there are 100 junctions in a system with 3 meters excess pressure for each of them, this solution is physically more acceptable in respect to the same system that has pressure violation on just one junction, but the amount of violation is 200 meter. Although $100 \times 3 = 300$ meters violation is more than $1 \times 200 = 200$ violation, but the 200 meters violation may cause pipe breakage, so the second scenario is not as feasible and acceptable. In this case raising the penalty to a power greater than one (e.g. two) let PEPSO see that $100 \times 3^2 = 900$ is way smaller than$1 \times 200^2 = 40000$.

Although penalty power factor of two is a default value of PEPSO, some simulation showed that sometimes the effect of the power factor of two is considerably severe, and 1.5 may be a more reasonable value that may better guide PEPSO to discover more feasible solutions.

Penalties that are associated with low limit violations are treated differently than penalties that are associated with high limit violations. These are treated differently and stored separately because they have different implications, and different policies should be implemented to reduce them. For instance, if PEPSO faces a high penalty value that is associated with the excess pressure, it might need to turn some pumps off to reduce the pressure of the system. On the other hand, if the same amount of penalty is related to insufficient pressure, PEPSO might need to turn some pumps on to increase pressure and tackle the issue.

PEPSO provides a flexible option for calculating penalties of strategic junctions and pipes that allows users to control the effect of violation of pressure or velocity of each component on the optimization process. For instance, In a WDS, controlling pressure of one junction might be more important than the others. In this case, users can increase pressure *constraint importance multiplier* of the desired junction to increase the effect of its penalty on the optimization process. As it was explained in section 2.2.1.5, these multipliers can be adjusted for each strategic junction and pipe and will be multiplied by calculated penalty of each junction or pipes before adding them up to calculate the total penalty. By default, these multipliers are 1.0, which means the violation of all strategic junction and pipes have the same effect on the optimization process.

If pump operation shows some violation regarding the defined pump constraints (e.g. number of pump switches in a day), the total penalty value will be increased one unit. Most of the time other penalties like tank level or junction pressure penalties are more important than pump operation penalties. So small pump operation penalties will not affect the optimization process unless other penalties are negligible.

.

#### 2.2.2.5. Undesirability Index Calculator

The undesirability Index calculator module is a very unique aspect of PEPSO that cannot be found in other pump operation optimization tools. This module calculates a value for each pump and at each time step that indicates if the pump status is desirable or not. A high absolute value of *Undesirability Index (UI)* shows that the status of the pump at that time step caused some problem (e.g. high pressure, low tank level). This pump status suggests that this pump schedule is a good candidate for modification to make the next iteration of pump schedule one step closer to the optimum pump schedule.

The calculation of UI value of a pump schedule at a specific time step requires knowledge of the tank level penalties, junction pressure penalties, negative pressure warnings and flow, head or connectivity warnings of the pump at that time step. For instance, if junction pressure penalty shows insufficient pressure at a time step, increasing the probability of turning on pumps at that time step may cause to increase the pressure of water at the junction and reduce the pressure penalty. So a positive value will be added to the UI value of all pumps at that time step. Inversely, if a pump head warning shows that the pump cannot deliver required head at the time step, increasing the probability of turning off pumps at that time steps may reduce energy consumed. In this case, a negative value will be added to the UI value of the pump at that time step. Therefore, if the resulting UI value of a pump at a time step is a positive number, increasing the probability of turning on the pump at the time step may result in an improved pump schedule. Conversely, if the number is negative, increasing the probability of turning off the pump at the time step might be more fruitful. Finally, if the UI value is zero, it means that there is not any definite sign that a change status in the operation of that pump at that time step will result in an improved pump schedule. As it was mentioned previously, the absolute magnitude of an UI value shows the magnitude of the probable positive effect of change of status of the pump operation on the optimization process. So larger absolute UI value indicates a high potential for improvement of pump schedule by changing the status of that particular pump.

Figure 18 Up and down arrows on the algorithm, show operations that change UI value of a pump at a time step in a way that increase or decrease the probability of turning on the pump, respectively (or increase/decrease rotational speed in the case of variable speed pump). This algorithm is encountered in each time step of the PEPSO simulation to calculate UI values of each pump at all time steps. For instance for an optimization run during a 24 hour period with one hour time intervals, this process should be repeated 24 times.

The calculated UI of each pump at each time step will directly be used during the *mutation* step. However in the *crossover* step, we need to know total UI of all pumps at a time step. The *Total Time Step Undesirability Index* (TTSUI) is simply calculated by adding up UI values of all pump at the time step. For instance for optimizing a WDS with ten pumps during a 24 hour period with one hour time intervals, we can calculate $10 \times 24 = 240$ UI values and 24 TTSUIs.

#### 2.2.2.6. Sorting

PEPSO uses the non-dominated sorting method. In this method, at first, objective values of all solutions will be compared to find the number of times that other solutions *dominated* each solution. By definition, solution A *dominates* solution B if both of these conditions are true: 1) the solution A is no worse than B in all objectives and 2) the solution A is strictly better than B in at least one objective (Deb 2001). Mathematical definition of domination for a minimization problem has been presented by equation 6 (Narzisi 2008):

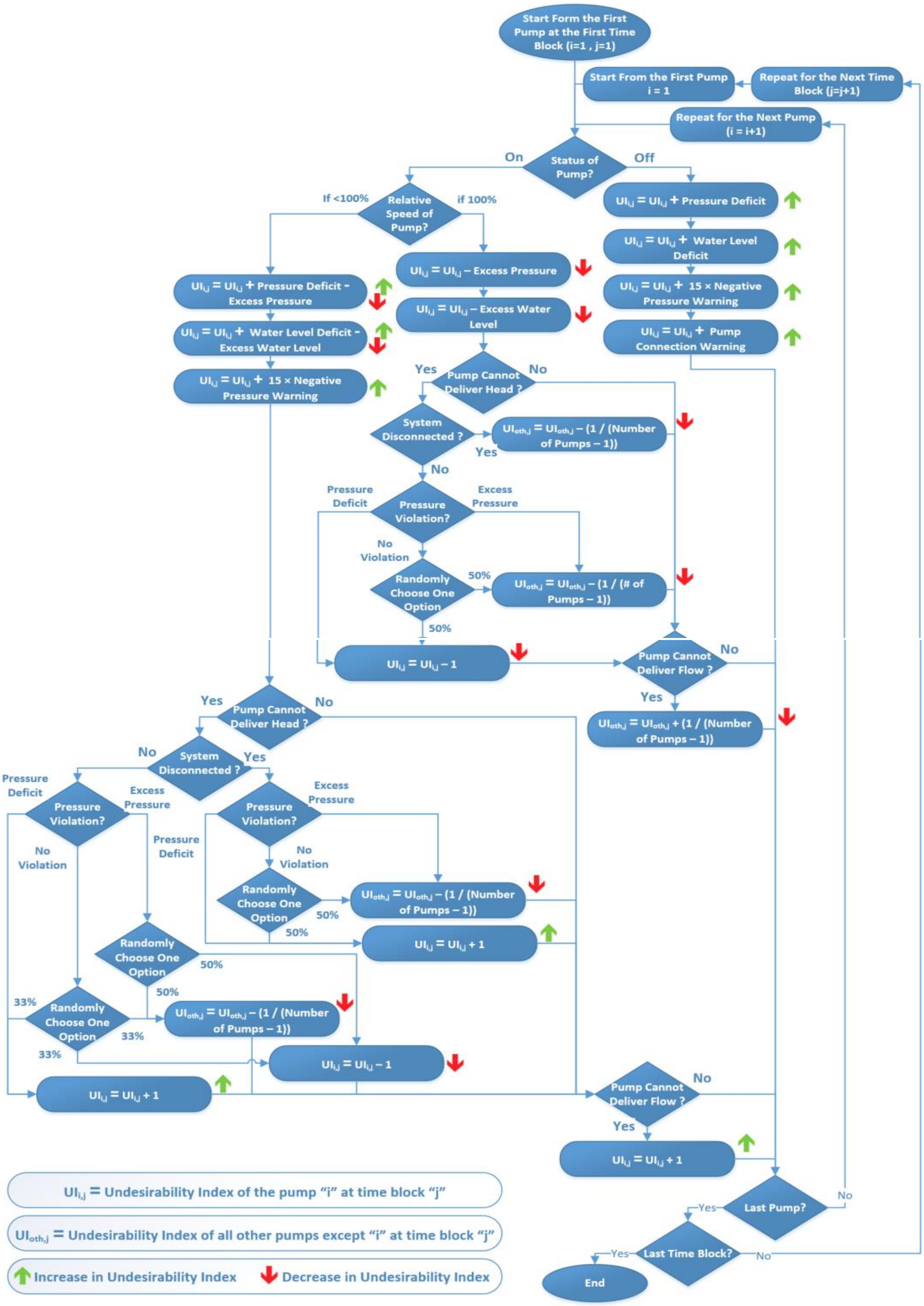



Figure 18- UI calculation algorithm of PEPSO

$$A \preccurlyeq B \;\; iff \begin{cases} f_i(A) \leq f_i(B) & \forall i \in 1, \dots, M \\ \exists j \in 1, \dots, M & f_i(A) < f_i(B) \end{cases} \qquad \text{Equation 6}$$

Where: A and B are two solutions,

$f_i(A)$ is value of $i^{th}$ objective of solution A and

$M$ is number of objectives of the minimization problem

Those solutions that have not been dominated by other solutions will be placed in the first Pareto frontier (Rank 1). Similarly, those solutions that are dominated just once will be placed in the second Pareto frontier (Rank 2), and so on. Figure 19 helps to visualize the idea of non-domination ranking for a two objective solution space. In this figure, “X1” and “X2” axis show values of two objectives of each solution that have been shown by different markers. For instance, “X1” and “X2” objective values of the solution “A” are 45 and 30, respectively. Comparing objective values of solution “A” and “B” suggest that “X1” objective of solution “B” is smaller than solution “A”. Also, the “X2” value of objective “A” is smaller than “B”. This problem is a minimization optimization problem with the utopia point of (0,0), So solution “A” is better than solution “B” in respect to “X2” objective but is worse than solution “B” in respect to “X1” objective. So none of these two solutions has an absolute advantage over the other, and neither dominates the other. Both solutions have been placed on the same Pareto frontier - as shown by the rectangular orange markers in Figure 19. However comparing solution “A” and “C” shows that Solution “A” is better than solution “C” on both objectives. So solution “C” is dominated by solution “A” and cannot be put on the same Pareto frontier as solutions “A” and “B”.

After non-domination ranking and finding the rank of each solution based on its Pareto frontier rank, the *crowding distance* of solutions that are within the same Pareto frontier will be calculated. The value of the *crowding distance* is used to sort solutions within a Pareto frontier (those that have the same rank). By convention, *crowding distance* of solutions that are located on edges of Pareto frontier is infinity. The first step in the calculation of *crowding distance* is the sorting (in ascending manner) of the solutions based on values of one objective. Then the solution with minimum objective value will be selected as the edge of the Pareto frontier, and its *crowding distance* will be infinity. The *crowding distance* of next solutions can be calculated by Equation 7 (Deb 2001).

$$CD_i(S_j) = \frac{f_i(S_{j+1}) - f_i(S_{j-1})}{f_{i_{min}} - f_{i_{max}}} \qquad \text{Equation 7}$$

Where: $CD_i(S_j)$ is crowding distance of $j^{th}$ solution in the sorted Pareto frontier based on $i^{th}$ objective

$f_i(S_{j+1})$ and $f_i(S_{j-1})$ are $i^{th}$ objective values of a solution before and a solution after the $j^{th}$ solution in the sorted Pareto frontier based on $i^{th}$ objective

$f_{i_{min}}$ and $f_{i_{max}}$ are smallest and largest values of the $i^{th}$ objective among solutions of the Pareto Frontier

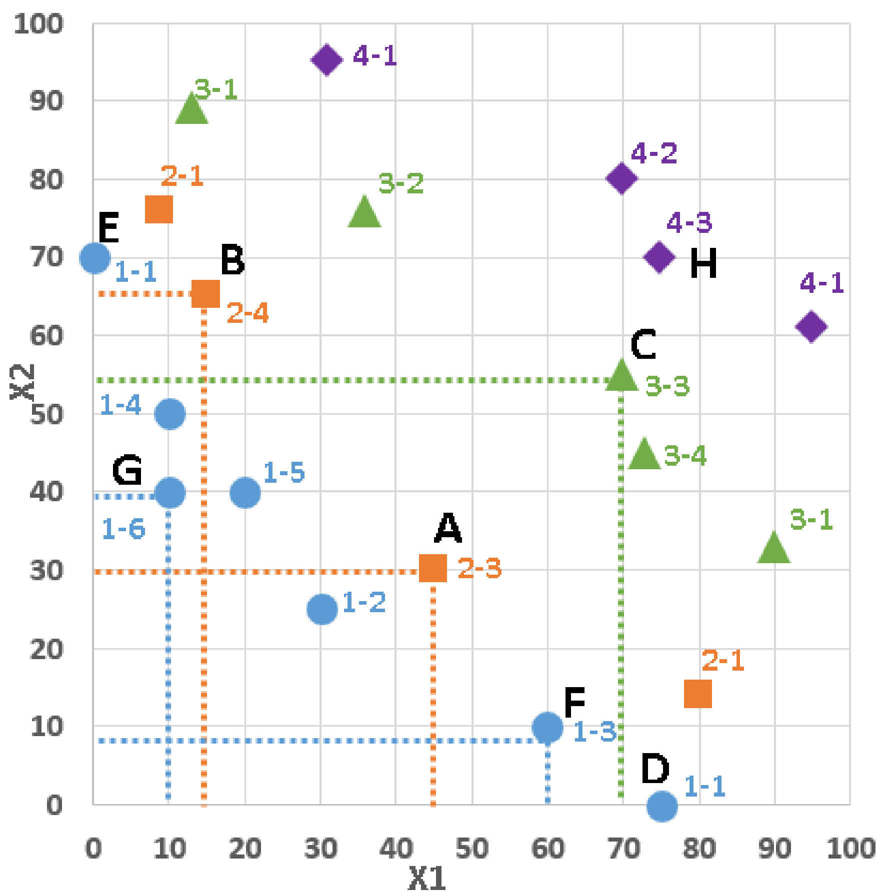



Figure 19- Non-domination ranking and *crowding distance* calculation

In the end, the same process will be repeated based on values of other objectives. The calculated *crowding distance* values of all objectives of a solution will be summed to provide the total *crowding distance* of the solution.

Figure 19 provides further insight into the crowding distance calculation. Figure 19 shows solutions that have been categorized in four Pareto frontiers. Solutions of each Pareto frontier have been shown with the same color and same marker shape. Solutions of the first Pareto frontier are shown by blue circle markers. Solutions D and E are two edges of the first Pareto frontier with minimum X1 and X2 values respectively. By definition, *crowding distance* value of these two solutions is infinity. *Crowding distance* of other solutions of the first Pareto frontier can be calculated by Equation 7:

*Crowding Distance* of Solution F = [(75-30) / (75-0)] + [(25-0) / (70-0)] = 0.957

*Crowding Distance* of Solution G = [(20-10) / (75-0)] + [(50-40) / (70-0)] = 0.276

As the *crowding distance* of solution F is larger than *crowding distance* of solution G, solution F will have a higher rank in respect to solution G among the solutions of the first Pareto frontier. After calculating the *crowding distance* values, solutions can be sorted initially based on the rank of their Pareto frontiers and subsequently based on their *crowding distance* within each Pareto frontier. By using this sorting method, all solutions of Figure 19 have been sorted and their rank are shown with two numbers that are separated by a dash (“-“). The first number shows Pareto frontier rank of each solution and the second number shows the rank of the solution among the same Pareto frontier. So Solution D and E from the edges of the first Pareto frontier have the highest rank in the whole population and solution H with the lowest *crowding distance* in the last Pareto frontier has the lowest rank. This sorting method helps the optimization algorithm to put more value on solutions that have been dominated less and are located in less crowded (less explored) regions of the solution space or on the edges of Pareto frontiers.

#### 2.2.2.7. Sampling and Elitism

PEPSO uses the roulette wheel sampling method to select solutions for *crossover*, *mutation* and to select elite solutions for the next generation (Deb 2001). It also uses the same method for selecting the part of a solution that should be changed during *crossover* or *mutation* processes. The sampling module receives an array of elements with their *proportional importance* (PI) and selects required samples *with* or *without replacement*. During elitism process when we want to select promising solutions and move them to the next generation, the above-mentioned array contains all solutions of the current generation, and their corresponding PIs are values that are calculated based on their non-dominated ranks. By this method solutions with higher non-dominated ranks have higher chance to be selected and moved to the next generation. Each solution can be moved to the next solution just once. Therefore, repeating is not allowed, and here PEPSO uses a *without replacement* roulette wheel sampling method. In another situation like *crossover* and *mutation* steps, when PEPSO wants to find an undesirable portion of a solution and change it to create an improved solution, the above-mentioned array contains different portions of the solution, and their corresponding PI is the UI of each portion of the solution. Here, a portion of the solution with a large UI should have a higher chance to be selected for replacement.

This example might help to clarify the selection process: assume there is an array of ten elements that are numbered from 1 to 10, and their PI values are equal. So we expect that all elements have the same chance of selection. However, before starting the random selection process, we need to create a cumulative PI vector. The cumulative PI of each element can be created by adding PI of the element to PI of all the previous elements in the array. Accordingly, the first value in the cumulative PI vector is one, the second number is 1+1=2 and so on to the last cumulative PI value that is 10. Now let's assume another scenario that the PI of each element is twice more than the previous element, the first value in the cumulative PI vector is one, the second number is 1 + 2 =3 and so on to the last cumulative PI value that is 1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 + 256 + 512 = 1023. For selecting an item, at first, a random number between 0 and the last value of the cumulative importance array will be generated. Index of the smallest cumulative importance value that is equal to or greater than the randomly generated value is the index of selected item. For instance, if the randomly generated value is 9, it means that in the first scenario when all items have the equal importance, the selected item is the 9$^{th}$ item (9<=9). However in the second scenario, it is the 5$^{th}$ item when the importance of each item is twice more than the previous item (9<=16). In the case of the sampling *without replacement*, effect of each selected item should be removed from the cumulative importance array, before selecting the next item. Figure 20 shows the process of sampling two sections from an array with seven elements (sections) with (left) and without (right) replacement by roulette wheel method. Based on Figure 20, the initial array has seven elements (colored sections), and cumulative PI of all elements are equal to 18. For selecting the first element, number 8 is generated randomly which leads to selecting the 3$^{rd}$ element. For selecting the second element, number 11 generated randomly that correspond to section 5$^{th}$ and 6$^{th}$ in *without replacement* (right) and *with replacement* (left) scenarios respectively.

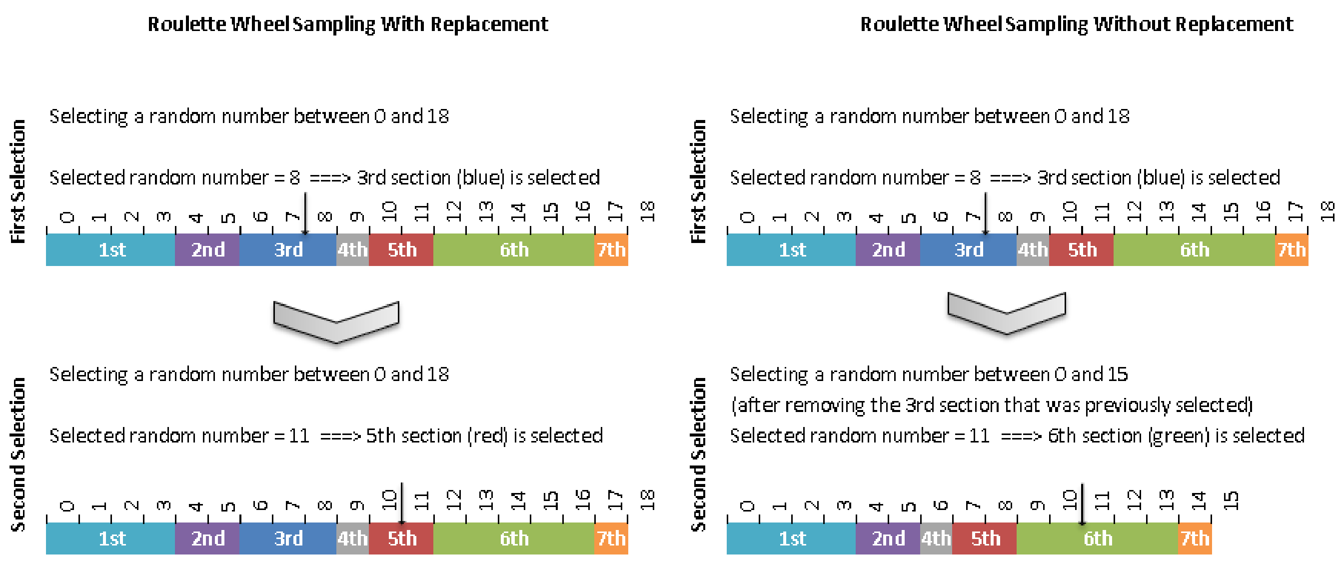



Figure 20- Sampling with (left) and without (right) replacement by roulette wheel method

It should be noted that after the first selection, total cumulative PI of the *without replacement* scenario gets 3 unit shorter than the *with replacement* scenario (as the third element (with PI length of 3) was selected, took out and not replaced in the right scenario)

Elitism process in PEPSO has two parts. After sorting solutions, a group of the solution with highest ranks will be selected directly as the first part of the elite population for the next generation. The second part of the elite population will be selected from the remaining solutions by using the roulette wheel sampling method and *without replacement*. The process of selecting the second group of elite solutions for the next generation is similar to the process of selecting solutions for *crossover* and *mutation*, which is explained in detail in the section 2.2.2.8. The size of the first group of elite solutions is user-defined. The second group of elite solutions has more variety in than the first group. Here the roulette wheel sampling method gives the solutions with low ranks in sorted population an opportunity to be selected as an elite solution. By default 20% of solutions of a population will be selected directly from the top of the sorted population and the rest will be selected by the roulette wheel method.

#### 2.2.2.8. Crossover

*Crossover* (also termed reproduction) is a GA operation which creates a new solution (child) by combining two or more selected solutions (parents). A standard *crossover* operator combines two parents by various techniques (e.g. single point, double point, and uniform) to form two children (Ting 2005). However, in PEPSO, a customized *crossover* technique has used that results in just one child. In this technique, at first, a pump schedule as the main parent will be selected. The more desirable solutions have a higher chance to be selected as the parent. Then one or multiple time blocks of the parent pump schedule that are not desirable will be selected to be replaced with potentially better time blocks of other solutions. When a time block of the parent pump schedule is selected for replacement, a better time block needed to be found to replace it. Therefore, the same time blocks of all available solutions will be ranked and by using the roulette wheel method a time block will be selected for replacement. Time blocks with a higher rank have a greater chance of selection to replace the undesirable time block of the parent pump schedule. By this method, we can expect that promising solutions will be selected and their undesirable time blocks will be replaced with better time blocks to form a more acceptable solution.

The selection of parent solutions requires the ranking of all solutions according to the non-domination rank and *crowding distance* in each Pareto frontier (as explained in section 2.2.2.6). The rank of solutions is used to calculate the PI value of each solution. Solutions with the highest rank (on the edge of the first Pareto frontier) are the most important solutions, and solutions with the lowest rank (in a crowded section of the last Pareto frontier) are the least important solutions. PI values of solutions will eventually be used to calculate the cumulative PI vector that will be used for selecting parents using the roulette wheel sampling method. However, before calculating the cumulative PI vector, PI of solutions should be adjusted based on 1) number of Negative Pressure Warnings (NPW) and 2) final tank level status of each solution.

For adjusting PI value of the solutions, at first, PI of the solutions that have NPW will be reduced by dividing it by a number that is calculated based on the number of NPW of the solution. A higher number of NPW increase the size of the denominator and reduces the importance value. By default, the function that calculates denominator value, adjusted in a way that if all solutions in a population except one has maximum number of NPW, probability of selecting the solution without NPW is 20% of probability of selecting one solution from the group of all other solutions with maximum possible number of NPW (see Equation 8). When there is no NPW associated with the solution, the minimum value of the denominator is 1 and when the solution has the maximum number of NPW the denominator value is equal to 1+(20% of the size of the population).

$$\text{NPW Probability Reducer Denominator} = 1 + (\text{Population Size} \times 0.2 \times (\text{No. of NPW of solution} / \text{Maximum No. of NPW})) \qquad \text{Equation 8}$$

After reducing PI of the solution based on the number of its NPWs, the status of final tank level of solution will be investigated. If final tank level is equal to or greater than the initial tank level, it is a desirable solution. However, if the final tank level is smaller than the initial solution, the amount of the tank level deficit will be calculated. Based on the calculated tank level deficit, PI of the solution will be reduced again. The formula and logic of calculating the tank level deficiency is similar to the calculation of the NPW Probability Reducer Denominator and appears in Equation 9.

$$\text{Tank Level Deficiency Probability Reducer Denominator} = 1 + (\text{Population Size} \times 0.2 \times (\text{Tank Level Deficiency of solution} / \text{Maximum Tank Level Deficiencies})) \qquad \text{Equation 9}$$

Accordingly, the effect of PI reduction of both NPW and tank level deficiency can be imposed on initial importance value of the solution by Equation 10.

$$\text{Adjusted Importance of the solution} = \text{Initial Importance of Solution} \,/\, (\text{NPW Probability Reducer} \times \text{Tank Level Deficiency Probability Reducer}) \qquad \text{Equation 10}$$

Larger values of the adjusted PI indicate that the solution has a higher rank in the non-dominated sorted population and has fewer NPWs and a less significant tank level deficiency. It means that in comparison to all solutions in the population, a solution with higher adjusted PI value is closer to the optimum solution and is feasible and desirable from the operation perspective. It makes the solution a good candidate to be parent and generator of the next generation of better solutions. The adjusted PI of solutions can be used to create the cumulative PI vector that will be used for selecting parents by roulette wheel technique *with replacement*. This means that a solution can be selected as a parent multiple times.

The TTSUI will be used for selecting some candidate undesirable time blocks that need to be replaced to create a better child solution. This means that, at first, the TTSUI of all time steps of each solution is calculated. Then TTSUIs are used as PI values to create the cumulative PI vector. This new cumulative PI vector will be used in the roulette wheel method *(without replacement*) to select time blocks with high TTSUI that are good candidates for replacement. Sampling with the roulette wheel method *without replacement* prevents a time block of the selected pump schedule to be selected multiple times for the replacement.

Finally, after selecting the parent and selecting those time blocks that are undesirable, it is necessary to select better time step from other solutions to replace the undesirable time blocks. For doing this, we need to compare the same time block of all solutions and rank solutions for each time step separately. PEPSO ranks time steps by using a combined factor that includes the rank of the solution in population and value of TTSUI of the same time step of each solution. The reciprocal of the rank of a solution will be added to the reciprocal of the TTSUI multiplied with a factor (by default, 15) to calculate PI of each solution for the time step (see Equation 11). By this method, the calculated PI includes the effect of both the desirability of the time step (reciprocal of TTSUI) and the rank of the solution. This means that to consider a time step of a pump schedule as a promising time step, we need to make sure that 1) the desirability of the time step is high, and 2) it comes from a high-rank solution (that means the time step can lead to a good solution when combined with other time steps of a pump schedule). It should be noted that the multiplier of 15 for the desirability part of formula puts the main emphasis on the desirability instead of the solution rank. PI of each time step of each solution will be added to the PI of the same time block of other solutions to create the cumulative PI vector of the time block. This cumulative PI vector will be used for selecting the solution that has the most promising time step by using the roulette wheel technique *with replacement*. The time step of the selected promising solution will replace the undesirable time step of a parent to form a better child.

$$\text{PI of the Time Step of the Solution} = (1 / \text{Rank of the solution}) + 15 \times (1 / \text{TTSUI}) \qquad \text{Equation 11}$$

By using this customized *crossover* technique, each parent generates one child that is mainly created from the one parent but may have some time blocks from other solutions. The focus of this *crossover* technique is on improving the solution condition by changing some time blocks of the pump schedule (columns), and it will not affect a row (the whole operation plan of a single pump) or a cell of pump schedule individually.

The number of parents involved in each iteration and number of time steps that need to be replaced can be defined by users. The first value that users define is the *crossover percentage*, which defines the percentage of the solution in the population which should be selected as parents. The second *crossover* parameter that users define is the *crossover rate*, which represents the percentage of the number of time steps of a solution that should be replaced with the similar time steps of other solutions. Both of these parameters can vary between 0 and 100%. By default, they are both set to 50%. This means that during each optimization iteration, by default, 50% of solutions will be selected as parents and 50% of time blocks of each selected parent will be replaced with promising time block of other solutions. It should be noted that if users want to input this numbers via the user interface they can use percentage values. However, inside the PEPSO, these percentages will be changed to a number between 0 to 1, and if users want to change them by editing the project file manually, they should convert percentages to a number between 0 to 1.

#### 2.2.2.9. Mutation

*Mutation* is a GA operator that generates a new solution by changing (mutating) some parts of a selected solution. Similar to the *crossover* operator, the *mutation* operator of PEPSO uses the UI map to select that portion of a pump schedule that would benefit from alteration. The process of selecting solutions for *mutation* is the same as the process of selecting parents for the *crossover* process.

After selecting the solution that should be mutated, absolute UI values of all cells of the selected pump schedule are used as PI values. This means that a cell of a pump schedule with high absolute UI value (high PI value) is a good candidate for *mutation*. PI values will be added to form the cumulative PI vector. Then the cumulative PI vector will be used for sampling some cells with the roulette wheel technique *without replacement*. Cells that are selected for the *mutation* will have their value reversed (on to off; off to on) if the cell represents a constant speed pump. The value in mutated cells associated with a variable speed pump will be modified based on the UI value of the cell. This means that the *mutation* will change an “off” pump to “on”, and the relative speed will be selected randomly from a distribution of numbers between the minimum RRS and 1 (full speed). Likewise, if the pump is “on”, its relative speed will be increased randomly if the UI values are positive. If the pump is “on” and its UI value is negative, there is 50% chance to turn off the pump and 50% chance to reduce its RRS randomly. In any case, RRS should always be between minimum RRS and 1.

The number of solutions that should be mutated can be defined by the user as a *mutation percentage*, representing the percent of the population that should be mutated. They also can define the *mutation rate* parameter, which represents the percentage of the cells of a selected pump schedule which should be mutated. Both of these parameters can be a number between 0 and 100% and by default are 5% and 10%. It means that during each optimization iteration, by default, 5% of the solutions will be selected for *mutation* and 10% of cells of each selected pump schedule will be mutated. Like the *crossover* parameters, users can adjust *mutation* parameters by inputting percentage values (between 0% and 100%) via the user interface but inside the PEPSO, these numbers will be converted to values between 0 and 1.

#### 2.2.2.10. Stopping Criteria

At the end of each iteration, PEPSO checks the stopping criteria to determine if the iterative solution process should continue or if the current solution should be accepted and the post-optimization process begin. PEPSO includes five stopping criteria. Users can define either: 1) the maximum time of optimization, 2) the maximum number of iterations, 3) the maximum number of solution evaluations, 4) a goal for each optimization objective and run the process until reaching those goals, or 5) the maximum number of stagnant iterations. For item 3), the number of solution evaluations can be calculated as:

$$\text{No. of Solution Evaluations} = \text{Population Size} \times (1 + \text{No. of Iterations} \times (\text{Crossover Percentage} + \text{Mutation Percentage})) \qquad \text{Equation 12}$$

For item 5), the term “stagnant” relates to the change in the value of the objectives of the solution. If the value of objectives of the best solution does not change more than a defined minimum value during an iteration, the iteration will be considered as a stagnant iteration. If multiple consecutive stagnant iterations occur, it means that the optimization process reached a local and potentially global optimum solution.

It is possible to select one or a combination of the five stopping criteria. If more than one is selected, the optimization process will be stopped when the first criterion is satisfied.

#### 2.2.2.11. Best Solution Finder

The best solution finder module of PEPSO is designed to select a single pump schedule as the best solution among a population of solutions. For selecting the best solution, three characteristics are considered. First of all, the best solution should be selected from solutions of the first Pareto frontier (those solutions that have not been dominated by any other solution). After that, the best solution should have the minimum combined objective value and an acceptable *inadmissibility* value.

The combined objective value is a linear combination of values of three objectives, each of these objectives having been multiplied by the user-defined weighting factor for that objective. It is important to note that the combined objective value is used solely to choose the best solution from among a population of solutions. The objective weighting factors are not used to convert the multi-objective optimization problem to a single objective problem. Equation 13 shows the simple formula that is used for calculating the combined objective value.

$$\text{Combined Objective Value} = \text{Electricity Cost} \times \text{Weighting Factor 1} + \text{Pollution Emission} \times \text{Weighting Factor 2} + \text{Total Penalty} \times \text{Weighting Factor 3} \qquad \text{Equation 13}$$

By definition, the value of the *inadmissibility* of a solution indicates how well a solution satisfies the minimum requirements for an acceptable and practical solution. The calculation of *inadmissibility* is similar to the calculation of penalties. The *inadmissibility* will be calculated using constraints of pump operations, water levels in tanks, water pressure at strategic junctions and water velocity in strategic pipes. For water level in tanks, water pressure at strategic junctions and water velocity in strategic pipes, users can define hard constraint boundaries that are wider and stricter than the soft, desirable ranges that have been used for calculating penalties. If the water level in tanks, the pressure at strategic junctions or water velocity in strategic pipes exceeds these hard constraints, the solution cannot be considered as a fully acceptable solution. For each time block, when a violation of these hard constraints occurs, a unit value will be added to the tank level, junction pressure or velocity *inadmissibility* values of the solution. Note that the *inadmissibility* values of the strategic junction and pipes will be multiplied by the *constraint importance multiplier*.

For pump operation, users can also define a hard constraint for a maximum number of switches in a day, minimum start intervals or maximum length of pumping cycles. Similarly, if a pump operation of a solution violates one of these limits a unit value will be added to the pump operation *inadmissibility* of the solution.

Before adding up *inadmissibility* values of pump operation, the water level in tanks and water pressure and velocity, the effect of all of them should be normalized to ensure that each of the *inadmissibility* values has the same scaled effect on the best solution selection process. *Inadmissibility* of pumps will be divided by the number of pumps times three (for three types of pump operation constraint that can be defined in PEPSO). *Inadmissibility* of tanks will be divided by the number of tanks. *Inadmissibility* of water pressure and velocity will be divided by total pressure and velocity *constraint importance multipliers*, respectively. Total pressure and velocity *constraint importance multipliers* can be calculated by adding *constraint importance multipliers* of strategic junctions and pipes, respectively. After implementing these normalization operations, each *inadmissibility* value will have a maximum value of 1.0. Therefore, the total *Inadmissibility* of pump operations, tank levels, water pressure at strategic junctions and water velocity in strategic pipes of a solution will sum to a value of 4.0 in the worst case.

Acceptable *inadmissibility* will be calculated based on the *inadmissibility* of the first Pareto frontier. Acceptable *inadmissibility* is the minimum *inadmissibility* among solution of the first Pareto frontier plus the *inadmissibility* tolerance value. By default, the *inadmissibility* tolerance is 10% of the difference between the minimum and maximum *inadmissibility* values of the solutions of the first Pareto frontier. This formulation allows PEPSO to select a solution with minimum *inadmissibility*, or select among solutions that may have a slightly larger value of *inadmissibility* (equal to *inadmissibility* tolerance) but low combined objective value. For instance, without considering the *inadmissibility* tolerance, a solution with high energy consumption and an *inadmissibility* value of zero might prevent another solution in the first Pareto frontier with lower energy consumption and a tank level violation at just one optimization time block from selection as the best solution. Although the latter solution has a violation from hard tank level constraint that creates an *inadmissibility* value slightly above zero, its electricity cost might be considerably lower than the former solution. In such a case, even though its *inadmissibility* value is not zero, it may be a good candidate for the best solution. Equation 14 provides the formula for calculating the acceptable amount of *inadmissibility* of a solution. As it was mentioned, the default *inadmissibility* tolerance percentage is 10%.

$$\text{Acceptable Amount of Inadmissibility among Solutions of the First Pareto Frontier} = \text{Minimum Inadmissibility} + \text{Inadmissibility Tolerance Percentage} \times (\text{Maximum Inadmissibility} - \text{Minimum Inadmissibility}) \qquad \text{Equation 14}$$

### 2.2.3. Hydraulic Solver

At the core of all the modeling, is the simulation of the hydraulic aspects of the WDS in order to determine the power demand and energy consumption of pumps, tank levels, water pressure at strategic junctions, water velocity in strategic pipes and finally warning messages of pump and system under a suggested pump schedule to evaluate the pump schedule and find ways to make that optimized. A WDS hydraulic simulator with extended period simulation (EPS) ability can provide all of this information. PEPSO uses two modeling approaches for this purpose. In the first approach, EPANET V2.0.12 toolkit (Rossman 1999) is used as a high-fidelity modeling tool. This toolkit lets us use the EPANET model of WDS, change its pump schedule and model it to get high accuracy results. The second approach is using a metamodel of WDS to model the hydraulic system faster than EPANET toolkit. The metamodeling technique that is implemented in PEPSO adopts an artificial neural network (ANN) to input pump schedule and other parameters into a trained ANN and receive the required result that normally was provided by EPANET toolkit. We expect increased the computational efficiency of hydraulic modeling by using a metamodel instead of a high fidelity model. However, it may reduce the accuracy of the solution. So using ANN instead of EPANET might increase the speed of optimization but may decrease its accuracy. Each of these approaches is described in the following sections.

#### 2.2.3.1. EPANET Toolkit

As introduced in Section 2.2.2.1, the EPANET file is initialized for optimization purpose during the pre-optimization phase. After initialization, each pump schedule of the initial population will be converted to a series of pump control commands that are readable for EPANET and will be added to the initialized EPANET file. This EPANET file will be used for hydraulic simulation with the toolkit.

EPANET toolkit solves the hydraulic network time step by time step. If between two time steps, the state of the system changes in a way that affects the hydraulic results, EPANET will solve the hydraulic equations another time in between the two time steps. For instance, assume a hydraulic simulation uses an hourly time step, and it starts from hour 00:00. The toolkit solves hydraulic equations for time 00:00 at first. However, if there is a rule or control at 00:30 that changes the status of a pump or valve or even if a tank gets full or empty at 00:30, the toolkit solves the hydraulic equation at 00:30 also. It then continues to solve the equations at 01:00, 02:00, and so on. Usually, report intervals are defined by the reporting time step parameter of the EPANET file. However, we might see some additional intermediate reports that correspond to a change of state of the system between two hydraulic time steps that creates an additional intermediate time step as was explained above. During the PEPSO optimization process, we are mostly interested in hydraulic results at hydraulic time steps. However, there are components of the intermediate results that may be important to us. For instance, PEPSO stores the *peak power demand* at each time step for use in calculating the power demand cost. If *peak* *power demand* at an intermediate report is higher than the *peak power demand* at the previous hydraulic time step report, *peak power demand* of the intermediate report will be stored as the *peak power demand* of the current optimization time step. Energy consumption during an optimization time step also is calculated as weighted average of energy consumption during all sub time intervals between two optimization time steps (weighted based on the length of each sub-time interval).

EPANET toolkit reports power demand of each pump during the reporting time step. PEPSO retrieves these numbers and uses them to calculate the *peak power demand* and energy consumption of each meter at each optimization time step. *Peak power demand* of each meter can be calculated by adding power demand of all pumps that are connected to the meter at each reporting time step and selecting the largest power demand value during the intended optimization time step. It should be noted that if the reporting time step is smaller than the *peak power demand* calculation period (based on the electricity tariff) the average *peak power demand* of all reporting time steps during that *peak power demand* calculation period will be used. Multiplying power demand of all pumps that are connected to a meter by the reporting time step results in the energy consumption at the meter during the reporting time step. Adding these energy consumptions over the period of an optimization time step provides the energy consumption at that meter during that optimization time step.

In addition to energy consumption and *peak power demand* of meters, PEPSO determines pump efficiencies during each optimization time step. PEPSO receives flow rate, head and power demand of pumps from the EPANET toolkit and uses these values to determine pump efficiencies.

Before starting the hydraulic simulation, PEPSO reads and stores the initial tank levels. Final tank level of each optimization time step will be obtained as the result of the hydraulic simulation. The water pressure at strategic junctions and water velocity in strategic pipes are also obtained as a result of the hydraulic simulation and stored by PEPSO. Finally, PEPSO reads the warning section of the EPANET toolkit hydraulic simulation report and determines if a pump has a Pump Head Warning (PHW) or a Pump Flow Warning (PFW) during an optimization time step. The presence or absence of these warnings is used in the calculation of UI values. Another pump warning that PEPSO may retrieve from the hydraulic simulation report is the Pump Disconnection Warning (PDW). In addition to the pump related warnings, for each optimization time step, PEPSO determines if the system has Negative Pressure Warning (NPW) or not.

After hydraulic simulation with the EPANET toolkit, all the above-mentioned information is stored in the data structure of each solution. It will be used later for evaluating the solution and finding promising ways to use *crossover* and *mutation* to improve the solution.

#### 2.2.3.2. Training Set Generator

Before using an ANN as a metamodel of WDS, it must be trained. For training an ANN, a training set that is comprised of a set of input values and their corresponding output values is needed. In the case of the WDS modeling, inputs are initial levels of tanks, the status of pumps, the speed of pumps (for variable speed pumps) and water demands. Required outputs of a model are final tank levels, peak power demand and energy consumption of pumps, water pressure at strategic junctions, water velocity in strategic pipes, PHW, PFW, and PCW of each pump and NPW of the system. As an approach to reduce the number of input values, the water demand multiplier of each demand pattern can be considered as an input, rather than providing water demands at all junctions. This number can be multiplied by base demand at each junction to calculate demand of the junction. In this case, instead of inputting multiple values as the demand of multiple junctions, a single value is provided for each demand pattern. For instance, if all junctions of a WDS have the same water demand pattern, all demand inputs of an ANN can be replaced with a single demand multiplier.

The training set creator module of PEPSO uses the EPANET input file of the WDS to create an ANN training set. Each ANN training point has two parts: 1) input part and 2) output part. A group of training points can form a training set which will be used for training an ANN. The training set creator module of PEPSO randomly changes initial tank levels, status and speed of pumps and demand multipliers and stores them as the input part of an ANN training point. Randomly generated input values will be used to initialize the EPANET model. After simulating the initialized EPANET model with the toolkit for a hydraulic time step, final results will be saved as output values of the same ANN training point. Repeating the process for several thousand times will form an ANN training set with thousands of training points.

The most important part of creating a training set is generating a balanced distribution of random input values. For instance, if status and speed of pumps change randomly, in most cases, about half of pumps are on, and half of the pumps are off. In this case, the probability of generating a random input condition that 20% or 80% of the pumps are on is low. However, in a real WDS there are often times when, for example, 20% or 80% of pumps are operating to satisfy the flow and pressure demands. Therefore, it is more practical to create a training set that is designed such that the number of training points with 20% or 80% pumps on (for example) is equal to the number of training points with 50% pumps on. These two preferred and problematic distributions of the number of training points in an ANN training set with respect to the percentage of pumps which are “on” have been shown in graphs of Figure 21. For creating this graph, 10000 random pump schedule are created for a WDS with ten pumps. The horizontal axis shows the percentage of pumps that are “on” in the randomly generated pump schedule. The blue (high dot density) bars show the distribution of pump schedule when they are created by a complete random and problematic algorithm. The orange (low dot density) bars illustrate the distribution of pump schedules that generated randomly but by using a controlled algorithm which results in a more diverse group of randomly generated pump scheduled with respect to the number of pumps that are “on”.

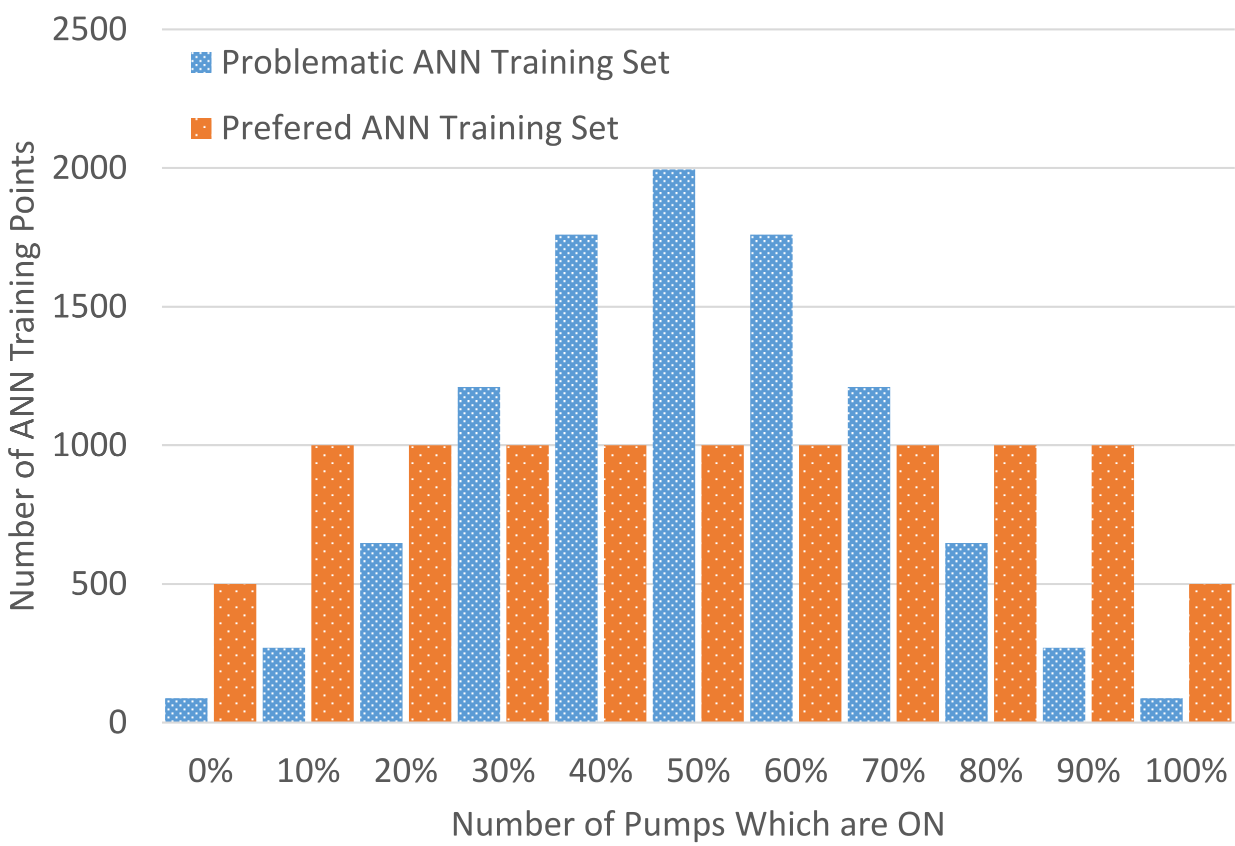



Figure 21- The preferred and problematic distribution of number of training points in an ANN training set with respect to the percentage of pumps which are ON

#### 2.2.3.3. ANN Trainer

The ANN trainer is a module of PEPSO, which uses the prepared training set to train an ANN for modeling the WDS hydraulically. PEPSO uses Fast Artificial Neural Network (FANN) libraries of code to create and train ANNs. FANN is a widely used free and open source library that was initially developed in C language in 2003 (Nissen 2003). A .Net wrapper is used that let PEPSO call FANN functions directly from VB.NET environment. ANN trainer module of PEPSO has two parts. The first part creates an ANN structure and the second part trains it. By using FANN library, PEPSO can create a *standard*, *shortcut* or *sparse* structure for the ANN. In a *standard* structure, each layer has connections to the next layer, while in a *shortcut* structure, a neuron can be connected to neurons of all the later layers. A *sparse* structure allows neural networks that are not fully connected. Activation (transfer) functions of hidden and output layers of ANN can be selected from a list of functions that are introduced in Table 5.

Table 5- Available activation (transfer) functions of hidden and output layers of ANN

| Name | Description |
|---|---|
| Sigmoid | Special case of logistic function with range of 0 to 1 |
| Sigmoid Symmetric | Hyperbolic tangent function with range of -1 to 1 |
| Sigmoid Stepwise | Stepwise linear approximation of sigmoid function |
| Linear | Linear function |
| Linear Piece | Bounded linear function |
| Sin Symmetric | Periodic sine function |
| Cos Symmetric | Periodic cosine function |
| Gaussian | Gaussian curve function |
| Gaussian Symmetric | Symmetric type of Gaussian function |
| Elliot | Fast sigmoid-like function defined by David Elliott |
| Elliot Symmetric | Fast sigmoid symmetric-like function defined by David Elliott |

The FANN library provides two training stop functions: *Mean Squared Error* (MSE) and *Bit*. MSE is a common type of stop function of ANN training process for function fitting. The *Bit* stop function can be used for training of the binary classification ANNs. FANN also can calculate training error using two linear and hyperbolic tangent functions.

Tangent hyperbolic function aggressively selects outputs that differ considerably from target values.

By using the FANN library, PEPSO can train ANNs with *standard* and *cascade* methods. In the *standard method*, the number and size of ANN layers must be defined by the user. In the *cascade method*, the trainer automatically adds layers to the ANN structure one by one to reach to an optimum structure. ANN trainer module is also able to train the network with three different algorithms. The *incremental algorithm* is a standard backpropagation method where weights are updated after each training. This means the weights will be updated many times during a single epoch. The *batch algorithm* is similar to the *incremental algorithm*, but all weights will be updated at once during an epoch (at the end of calculating MSE of the entire training set). It is also possible to use the *Rprop* and *Quickprop algorithm* for training. Although these advanced batch training algorithms can be more efficient than the standard *incremental* and *batch algorithm*, they have more parameters that need to be adjusted.

The FANN library enables PEPSO to use these many different options for training the ANN. However, most of them are not familiar options for WDS operators, and designers and PEPSO do not rely on the user to select the ANN training options. Instead, if a user selects the ANN metamodel instead of EPANET hydraulic model, all the related options will be selected automatically by PEPSO.

By default, PEPSO uses the *standard* structure with one hidden layer for ANN training of WDSs. Using the sigmoid symmetric and linear activation functions for hidden and output layers are suggested in the case of training ANN for function fitting. For training ANNs for classification problem, the sigmoid and linear activation functions work better (Kriesel 2007). Training ANN for warning message simulation that provides binary output (true and false: for existence or absence of warning message at a time step) is a classification problem, so the Sigmoid and linear functions are the default activation function of ANN trainer of PEPSO. However, training ANN for calculating final tank level, power demands of pumps, junction pressures, and velocities are function fitting type of problems. Accordingly, the sigmoid symmetric and linear activation functions have been used as default activation function of ANN trainer of PEPSO. For this ANN, the MSE functions are used as default stopping function of ANN trainer. The *Bit* function is used as the default stopping function of the ANN trainer for warning messages. By default, the *batch algorithm* with the back propagation training method is used for training all ANNs.

It is important to know that when PEPSO trains an ANN, it trains a metamodel that receives inputs as an initial condition of the system at the start of an optimization time step and provides outputs that are the hydraulic result of the system at the end of the optimization time step. So for each time step metamodel should be used to provide outputs, and this process should be repeated to model the WDS during the whole optimization period. For instance, for an optimization run during a 24 hour period with one hour time intervals, ANN should be used 24 times.

#### 2.2.3.4. ANN User

After training a metamodel, the ANN user module of PEPSO uses it to replace the EPANET model. Trained ANN works like the EPANET toolkit and reports results for each optimization time step. Tank level output of ANN for the previous time step is the tank level input of the current time step. Tank level inputs of the Initial time step are the initial tank levels of the system. This consecutive usage of ANN may cause some cumulative error for tank level at the end of simulation duration.

### 2.2.4. Output Reporter

PEPSO provides different types of outputs. It includes formatted text outputs (*.txt), graphs (*.fig, *.jpg) and optimized EPANET input file (*.inp). Final EPANET input file includes the optimized pump schedule and can be used directly in EPANET software to consider the effect of the optimized pump schedule on a different part of the WDS. All of these different types of outputs are explained in the following sections.

#### 2.2.4.1. Text Output Creator

This module of PEPSO receives a population and reports almost everything about the solutions of the population in the form of a formatted text file. Although users can select the level of details that they want to have in the text report, by default, it includes all details that have been listed in Table 6. Most of the information in the text report are formatted in a tabular form with tab delimiters that make it easy to read information directly from the text report or to copy and paste it into an Excel file or even read it with other software for further process or archiving.

Table 6- Sections of the text report of PEPSO

| Section name | Frequency of report | Content |
|---|---|---|
| Optimization project summary | One at the beginning of the report | Main project information (name, location, files) |
| | | Electricity meter data (pump list, tariff, and pollution emission scenarios) |
| | | Constraint (pumps, tanks, strategic junction and pipes) |
| | | Optimization options (objectives, stopping criteria, algorithm) |
| | | Initial pump schedules |
| | | Reporting options (text and graphics) |
| Iteration Report | Every iteration | Population summary schedule (objective values of all solutions) |
| | | UI summary of population (UI of all solutions at each time step) |
| Final population report | One at the end of the report | Optimization trend summary (average of populations and best solution) |
| | | Final population summary schedule (objective values of all solutions) |
| | | Final UI summary of population (UI of all solutions at each time step) |
| | | Detailed results of final solutions (pump schedule, UI table, pump flow, head and connection warning, pump operation statistics report, pump penalty, electricity cost, power demand table, tank level table, tank penalty table, strategic junctions pressure table, strategic pipes velocity table, strategic junction and pipe penalty tables and negative pressure warning table) |

### 2.2.4.2. Plotter

In addition to the detailed text report, PEPSO can provide results in graphical format. PEPSO reports the selected pump schedule in a graphical format as shown in Figure 22. Pump schedule graph indicates working cycles of the pump by colorful horizontal bars. The horizontal axis of this graph shows the optimization period. The vertical axis shows the name of the pumps. For instance, Figure 22 shows pump schedule of 13 pumps of a WDS during a 24 hour period of optimization. Each row of the graph shows operation plan of a pump. For constant speed pumps, the duration of time that the pump is on is indicated by a colored line. For variable speed pumps, the RRS of the pumps are noted on the colorful bars (see pump PMP-9 and PMP-544 of Figure 22).

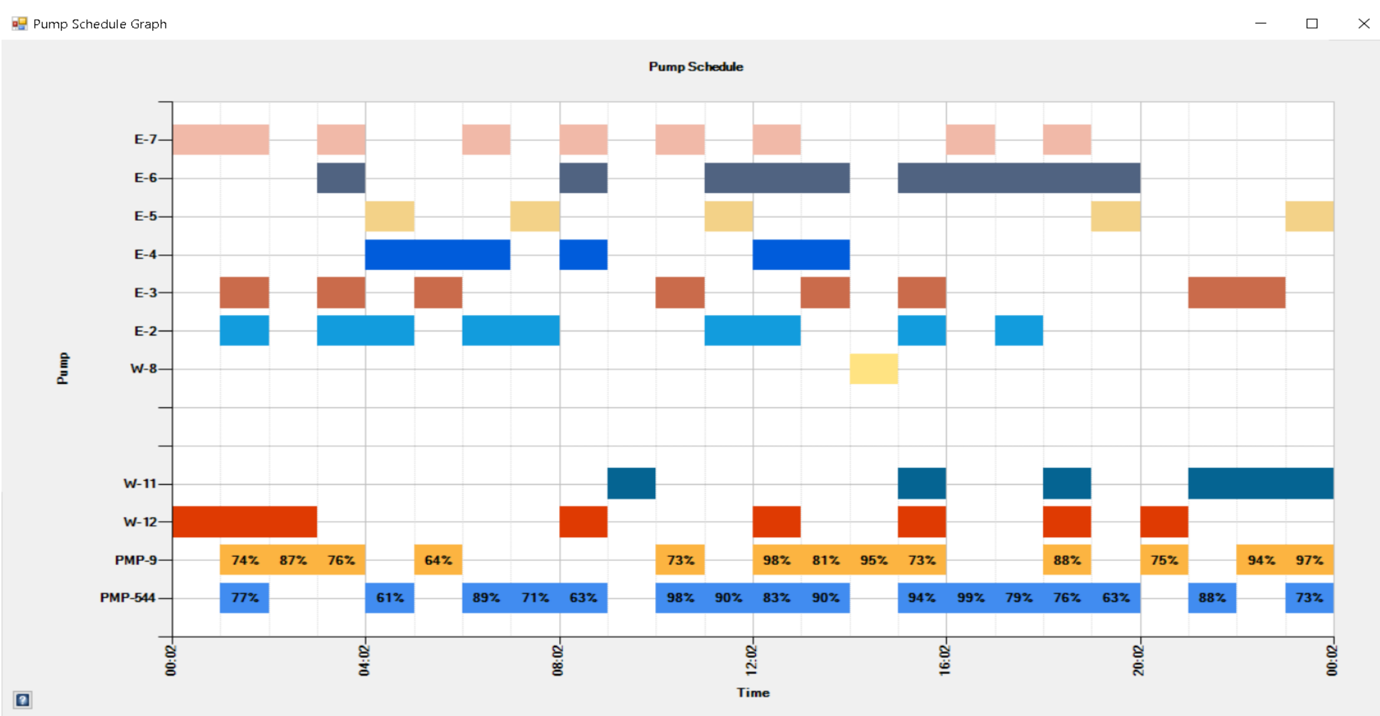



Figure 22- Graphical representation of the best pump schedule

PEPSO can display optimization trend graphs for the best solution and average of the population (see Figure 23). Each graph shows the trend of minimization of different objectives. For objectives like total penalty or electricity cost that are calculated from different values, values of the components can also be plotted. For instance, users can select the display of optimization trend of energy consumption cost and power demand cost component of the electricity cost. They also can see trends of pump operation penalties, penalty of water level in tanks, penalty of water pressure at strategic junctions and penalty of water velocity in strategic pipes that ultimately form the total penalty. Trends of objective values of the best solution of each population are plotted on the left side, and trends of the average of objective values of solutions of a population are plotted on the right side (see Figure 23).

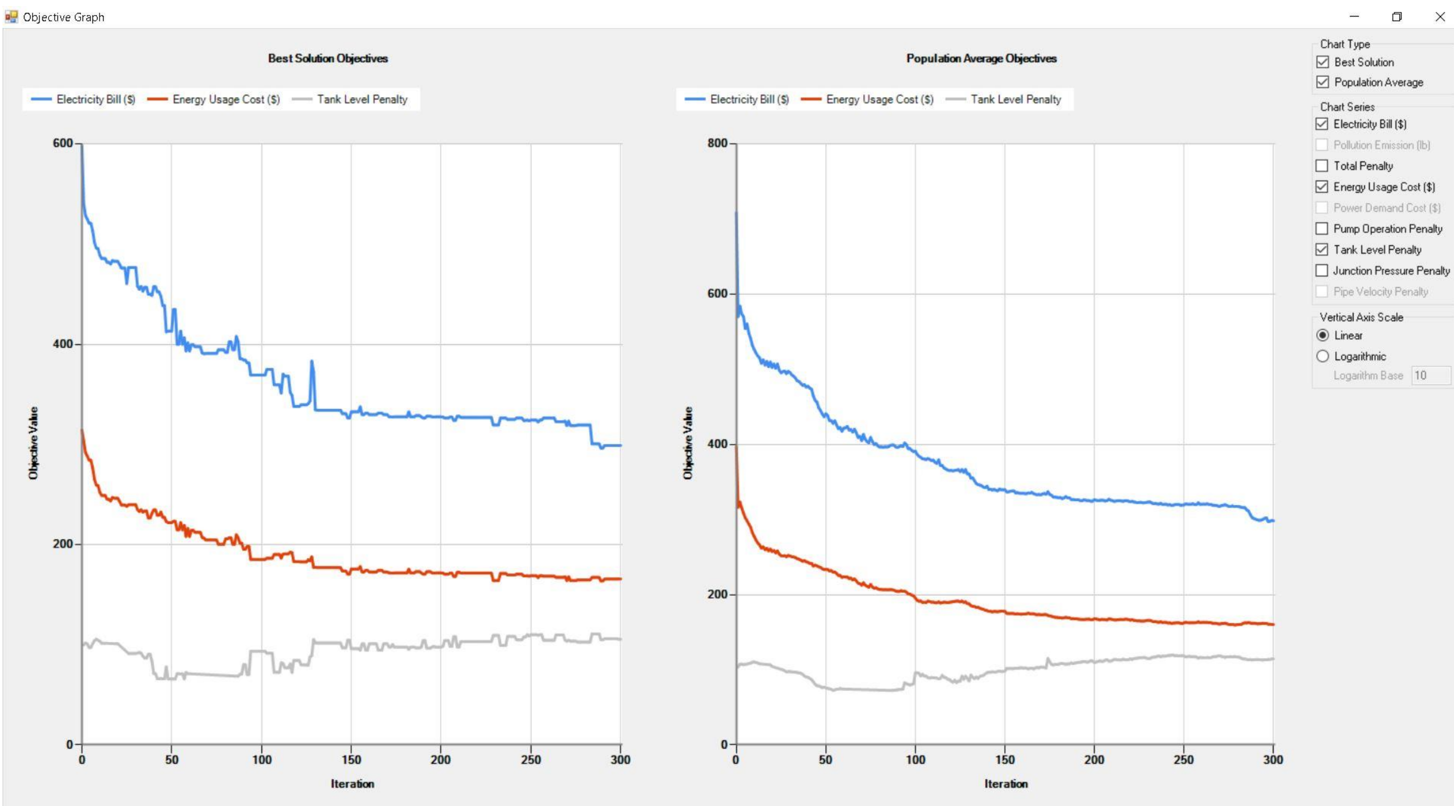



Figure 23- Optimization trend of objectives of the best solution (left) and average of population (right)

The horizontal axis of each graph shows iterations of optimization and the vertical axis shows the objective values. Users can select a linear or logarithmic scale for the vertical axis. This feature might be useful for investigating results of the WDS which initially has a considerable amount of penalty but after successive generations have much smaller penalty values.

Finally, PEPSO can be used to display each population in the solution space using a three-dimensional Pareto frontier graph, as shown in Figure 24. The 3D plot is generated by an implemented MATLAB library inside the plotter module of PEPSO. This plot is customizable and has a user-friendly interface that allows users to rotate, zoom and pan. Via the reporting options form of PEPSO, users can define the objective value that is displayed on each axis. It also is possible to change scale of each axis to logarithmic scale. A logarithmic scale works well in displaying penalty values that are calculated using a power penalty function. As shown in Figure 24, a group of solutions (dots) that are in the same Pareto frontier are separated from other solutions by color coding. The legend of this plot defines the color coding (colored gradient bar on the right side of plot).

## 2.3. Test Cases and Optimization Tests Setup

The test plan and scenarios that are designed to evaluate efficiency and accuracy of PEPSO are explained in part 2.3.1. Following that, the characteristics of WDSs that are used as test cases are described.

### 2.3.1. Testing Plan and Scenarios

In this study, PEPSO tested with two approaches. At first, different functionalities of PEPSO are tested and the result of the optimization process in different scenarios compared with a base scenario. In the second approach, PEPSO results compared with some other available methods including the famous Darwin Scheduler component of WaterGEMS software.

As it was mentioned, in the first approach a base case scenario was used as a reference point of comparison. In this scenario, it was aimed to reduce electricity cost and the total penalty of a WDS. The electricity cost objective is total power demand cost and energy consumption cost based on a time variant tariff. The total penalty objective composed of penalties that are associated with the water level violation in tanks and water pressure violation at strategic junctions.

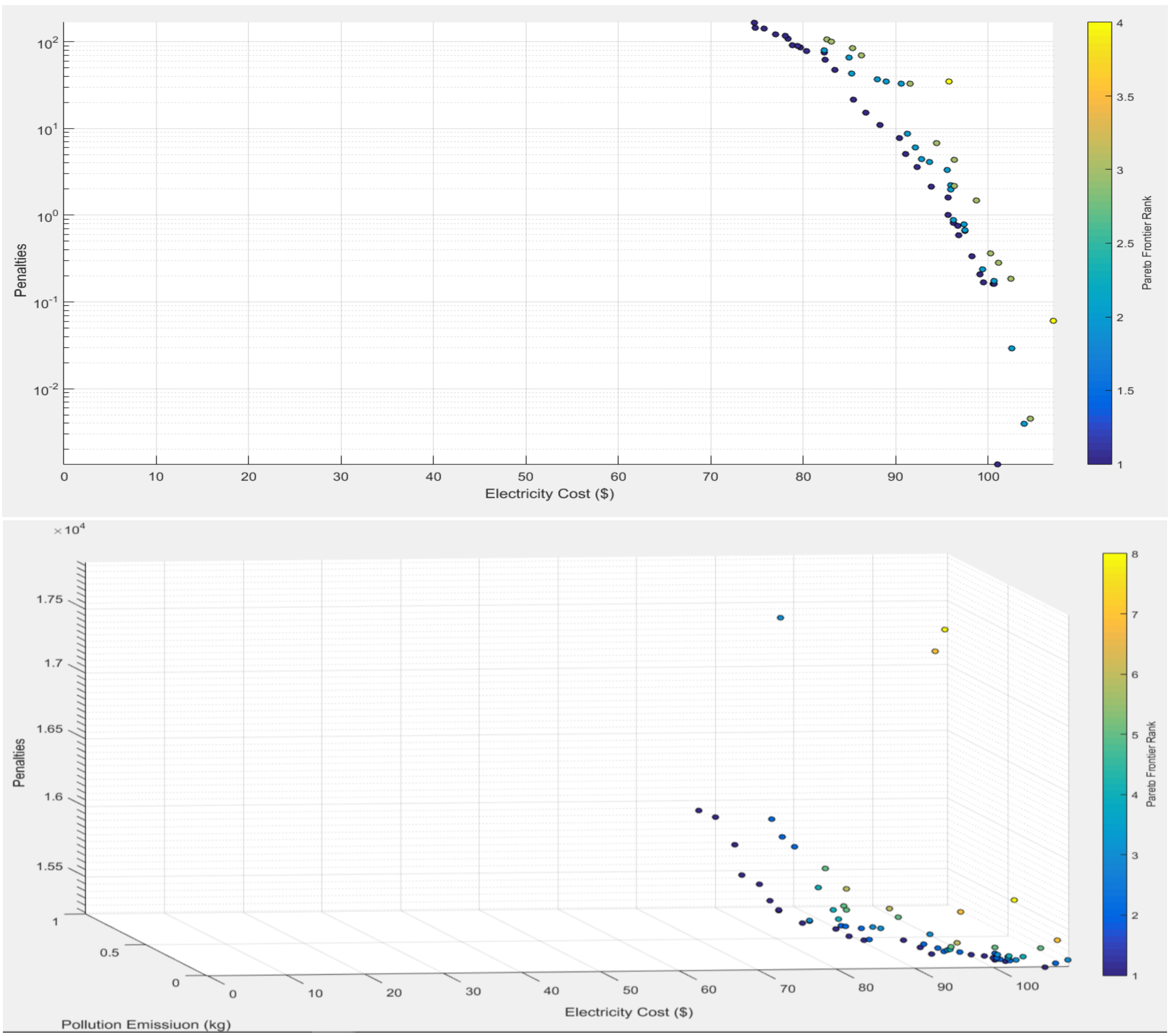



Figure 24- 2D (top) and 3D (bottom) views of Pareto frontier plot of PEPSO

The EPANET hydraulic solver is used for optimizing the base scenario. Seven more scenarios are defined to evaluate different functionalities of PEPSO. These scenarios are described in Table 7. First row is related to the base scenario (Sc1) and in all other rows, one cell is shaded that show one component of the base scenario which is changed to create a new scenario. In the second Scenario (Sc2), in addition to electricity cost and total penalty, pollution emission is optimized. This scenario is using all features of PEPSO to reduce both electricity cost and pollution emission of the system while satisfying required pressure and flow requirements.

Table 7- Scenarios that are used for testing different functionalities of PEPSO

| Scenario | Hydraulic model solver | UI calculation | Energy consumption cost / power demand cost | Electricity cost / pollution emission | Pump operation / water level in tank / pressure at junction penalties |
|---|---|---|---|---|---|
| Sc1- Base (EPANET solver, with electricity cost & total penalties optimization & without pollution emission optimization) | EPANET | Yes | Time-dependent / Yes | Yes / No | Yes / Yes / Yes |
| Sc2- All objectives (electricity cost, pollution emission, total penalty optimization) | EPANET | Yes | Time-dependent / Yes | Yes / Yes | Yes / Yes / Yes |
| Sc3- Just total penalty optimization (without electricity cost & pollution emission optimization) | EPANET | Yes | Time-dependent / Yes | No / No | Yes / Yes / Yes |
| Sc4- With pollution emission & total penalty optimization & without electricity cost optimization | EPANET | Yes | Time-dependent / Yes | No / Yes | Yes / Yes / Yes |
| Sc5- Without using UI calculation | EPANET | No | Time-dependent / Yes | Yes / No | Yes / Yes / Yes |
| Sc6- Just pressure penalties (without tank penalties) | EPANET | Yes | Time-dependent / Yes | Yes / No | Yes / No / Yes |
| Sc7- Energy usage optimization instead of electricity cost optimization | EPANET | Yes | Constant / No | Yes / No | Yes / Yes / Yes |
| Sc8- ANN solver instead of EPANET solver | ANN | Yes | Time-dependent / Yes | Yes / No | Yes / Yes / Yes |

The total penalty is the only objective that is optimized in the third scenario (Sc3). In this scenario pollution emission and electricity cost are calculated but not optimized. In some extent, this scenario shows the common operational condition of WDSs that reducing electricity cost and pollution emission is not the first priority of operators, and they focus on satisfying required pressure and flow of the system. The emphasis of the fourth scenario (Sc4) is on reducing pollution emission.

In the fifth scenario (Sc5), UI calculation module of PEPSO is deactivated to evaluate the effect of this unique feature of PEPSO on optimization efficiency and accuracy.

Accordingly, in this scenario electricity cost is calculated but not optimized and just pollution emission and total penalty are optimized. In the sixth scenario (Sc6) the effect of penalties of water level violation in tanks on optimization process is investigated. In contrast with the base scenario, in this scenario tank levels are not controlled and penalized. In the seventh scenario (Sc7) amount of energy usage (kWh) is optimized instead of electricity cost ($). In this scenario energy consumption charge does not change at different time and peak power demand doesn't increase total electricity cost. So the only factor that changes electricity cost of the system is the total amount of energy consumption (kWh) which should be optimized. Finally, in the last scenario (Sc8), instead of EPANET solver, ANN, model and solver are used to investigating the effect of using ANN model on increasing speed of optimization process.

Each of these scenarios repeated five times and descriptive statistics measures of the results are used to compare scenarios and report the findings. At first, the average of the results of all runs of a scenario is calculated and reported. In addition, to average values, unbiased estimation of the standard deviations, standard errors of the mean and relative standard errors of the mean are calculated and used to show the accuracy of the results; Osborn 2006). By definition *standard error of mean* (SEM) shows how the calculated mean of a sample is likely to differ from the real mean of the population (Vogt and Johnson 2011). Here, the unbiased estimation of the standard deviation of a sample is used in the Equation 15 to calculate the SEM. Then the relative SEM is calculated and reported as a percentage by dividing the calculated SEM by mean value.

$$SEM = \frac{s}{\sqrt{n}} \qquad \text{Equation 15}$$

Where,

$SEM$ is standard error of mean,

$s$ is the sample-based estimation of the standard deviation of the population

$n$ is size of the sample

In addition to the SEM calculation, optimized pump schedules of similar optimization runs of a scenario were compared to check accuracy and consistency of PEPSO results. In this comparison, the status of each pump at each time block of a pump schedule is compared with the status of the same pump at the same time block of other pump schedules to see if repeating the same optimization scenario is generating similar results or not. After comparing pairs of pump schedules cell by cell, average index of similarity is reported for each scenario. This index can be a percentage between 0% to 100%. A 100% value means that repeating the optimization process of one scenario always resulted in the same optimized pump schedule and 0% mean that result of different repetition looks completely random and dissimilar.

Comparing the result of PEPSO optimization with another available optimization is the second approach that is taken for testing PEPSO. Besides PEPSO, *Darwin Scheduler* (DS) component of WaterGEMS software used as the other pump schedule optimization tool for this test. The same Monroe WDS model that was used for the previous set of test is used for this test too. Optimization scenario of this test is similar to the base scenario of previous test and optimization scenario that is used for comparing the older version PEPSO with DS and Markov Decision Process (MDP) in previous studies (Alighalehbabakhani, Abkenar et al. 2014). DS cannot use LEEM report or other sources of data to calculate time-dependent pollution emission of the system. Therefore, pollution emission is not included in the scenario of this test. Minimum and maximum physical limits of water elevation in tanks considered as hard constraints and the soft constraint on desired water level in the tank are not used in this test. The water pressure constraint at strategic junction and pump operation constraints of the base scenario of the previous tests are used for this test too. The fast, messy Genetic Algorithm of DS is used as the optimizer algorithm that its parameters are listed in Table 8. It is important to note that from different stopping criteria the maximum trial number is the first one that will be met in this tests. This number is equal to the number of solution evaluation of the PEPSO. So by this way, we make sure that both tools have the same amount of solution evaluation chance to find the optimum solution. Most of other parameters that are listed in Table 8 are default values of DS. The same optimization algorithm options and reporting options that were used for the base scenario of the previous test is used for this test too. DS algorithm uses a random seed value to randomize initial condition of optimization process a different random value is used for each optimization run of DS to enable it to start the optimization process from different areas of the solution space (similar to PEPSO)

After optimizing the WDS with both PEPSO and DS, results are compared and discussed to evaluate the accuracy, speed, and usability of these two tools. DS is a well know commercial tool in the market for optimizing operation of pumps in WDS but it is not able to optimize time and location dependent pollution emission of WDS. So the optimization scenario that is defined for this test is aiming to reduce the electricity cost while the water level in the tank and water pressure at the strategic junction are within the defined ranges. The results of optimization with PEPSO (V2.0) and DS (V8i, series 6) are investigated and required time for optimization, total electricity cost, the amount of water level violation in tanks and water pressure violation at strategic junctions are compared.

Table 8- Optimization parameters that are used for DS optimization runs

| Parameter | Value |
|---|---|
| Objective | Minimizing energy cost |
| Optimization algorithm | Fast messy genetic algorithm |
| Population size | 100 |
| Elite Population size | 10 |
| Number of crossover points | 5 |
| Probability of crossover | 95% |
| Probability of mutation | 1.5% |
| Probability of creeping mutation | 0.1% |
| Probability of creeping down | 65% |
| Probability of cut | 1% |
| Probability of splice | 90% |
| Probability of elite mate | 0.5% |
| Probability of tournament winner | 95% |
| Maximum generation | 100 |
| Maximum eras | 10 |
| Maximum trials | 16600 |
| Maximum non-improvement generations | 200 |
| Pressure penalty factor | 1 |
| Velocity penalty factor | 1 |
| Pump starts penalty factor | 10 |
| Tank final level penalty factor | 10 |
| Tank high/low-level penalty factor | 1 |
| Minimum relative speed change of variable speed pumps | 1% |

For doing all the above-mentioned tests a computer system with these specifications is used: Lenovo ThinkPad W520 with Intel Core i7-2820QM 2.3GHz, 8MB cache CPU, 8GB DDR3 RAM, 7200 RPM SATA HDD and NVIDIA Quadro 2000M w/2GB DDR3 GPU

This computer system is selected for conducting the test because it is a common type of computer system that can be found in engineering offices for designing or operating WDS. So result that is obtained by this computer can be obtained in a practical situation in an ordinary design or operation office in water industry section. In the result chapter, *CPU time* and real time of different optimization runs that are conducted with this computer are reported. *CPU time* is the amount of time that CPU spent on a processing instructions of a section of code of PEPSO and calculated by multiplying real time of completing the process by average CPU usage percentage at that period.

### 2.3.2. Test Cases

Two WDS models are used in this research for evaluating PEPSO. The first one is WDS of the city of Monroe, Michigan. This WDS consist of over 450 (km) of distribution lines which range in size from 50 to 910 (mm). There is 11 constant speed pump in the main pump station that is connected to a reservoir as the only source of water of the system. Also, two variable speed booster pump are installed in the second pump station. Nominal power of these pumps range from 36 to 220 (kW) Figure 25 displays a model schematic of WDS of Monroe.

Ground level has a mild slope from North West toward South East. The minimum and maximum elevation of the demand points of the system are 174.5 and 201.8 (m) respectively. Three elevated tanks are located in different spots of the WDS and their total water storage capacity is 3974 ($m^3$) (11% of the daily water demand). For filling the elevated tanks, in addition to the required dynamic head of the system, pumps need to provide enough pressure to overcome 60 (m) of static head. WDS of the city of Monroe serves about 8000 customers, and its water demand is 36500 ($m^3$/day). The minimum and maximum hourly demand multipliers of the system are 0.67 and 1.19 respectively.

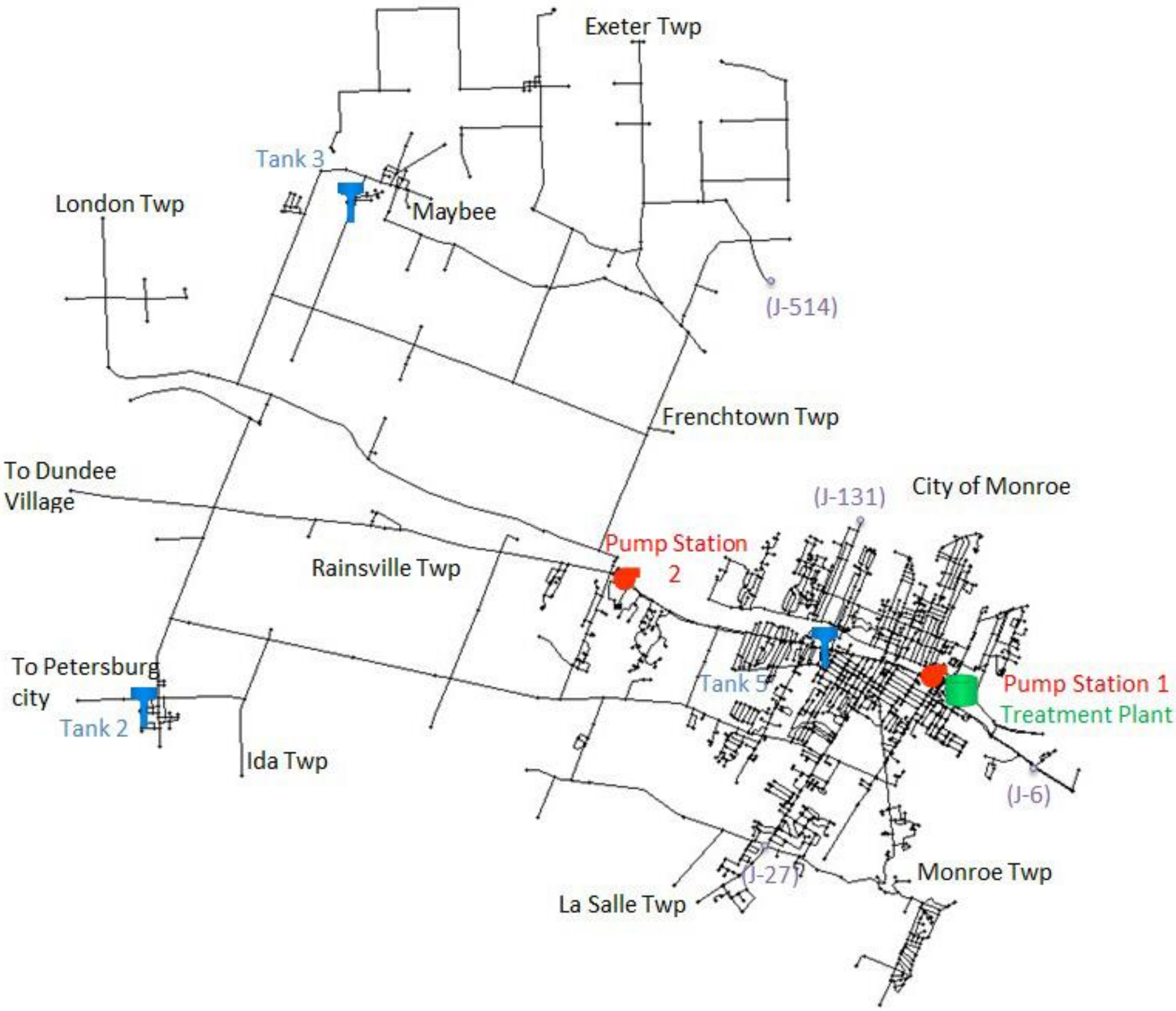



Figure 25- Model schematic WDS of Monroe

EPANET hydraulic model of WDS of Monroe that is used in this study has 1531 junctions, 1945 pipes, 11 constant speed pumps, two variable speed pumps, one reservoir, three tanks, and one 24 hour water demand pattern with one hour time step. The hydraulic simulation period of the model is also 24 hour with a one-hour time step EPANET model of WDS of Monroe was used for both groups of tests. It is used for evaluating different functionalities of PEPSO by testing eight different scenarios. It also is optimized by both PEPSO and DS for comparing these two tools. Table 9 and

Table 10 presents a constraint on the water level in tanks and water pressure at strategic junctions that was used in all optimization scenarios. As it is shown in

Table 10 *constraint importance multipliers* of all four strategic junctions are one that indicates to the same importance level for water pressure violation at all strategic junctions

Table 9- Constraints on water level in tanks of the Monroe WDS

| Tank ID | Elevation (m) | Water Capacity ($m^3$) | Minimum Allowed Water Level (m) | Minimum Desired Water Level (m) | Maximum Desired Water Level (m) | Maximum Allowed Water Level (m) |
|---|---|---|---|---|---|---|
| T-2 | 217.09 | 965 | 0.15 | 1.56 | 8.12 | 9.53 |
| T-3 | 225.78 | 956 | 0.15 | 1.41 | 7.28 | 8.53 |
| T-5 | 235.31 | 2053 | 0.30 | 1.78 | 8.66 | 10.13 |

Table 10- Constraints on water pressure at strategic junctions of the Monroe WDS

| Strategic Junctions ID | Minimum Allowed Pressure (psi) | Minimum Desired Pressure (psi) | Maximum Desired Pressure (psi) | Maximum Allowed Pressure (psi) | Constraint Importance Multiplier |
|---|---|---|---|---|---|
| J-6 | 0 | 42 | 52 | 284 | 1 |
| J-27 | 0 | 31 | 45 | 284 | 1 |
| J-131 | 0 | 28 | 42 | 284 | 1 |
| J-514 | 0 | 42 | 55 | 284 | 1 |

The minimum relative rotational speed of all variable speed pumps is 60%. The maximum allowed a number of pump switches in a day is 24 and the minimum duration of time between pump shut-down and start-up is 15 minutes. The maximum allowed a continuous period of operation for the pump is 24 hours.

Electricity tariff includes the *energy consumption charge* and *power demand charge*. The *energy consumption charge* for on-peak hours (11:00 to 18:59) is 0.04408 ($/kWh) and for off-peak hours (19:00 to 10:59) is 0.04108 ($/kWh). For Scenario Sc7 that *energy consumption charge* is constant throughout the day, off-peak rate is used for the whole 24 hour period. The *power demand charge* is 14.34 ($/kW) that should be multiplied by the 30 minutes peak power demand during 30 days period to calculate the power demand cost. So considering the similar peak power demand for all days of a month, daily power demand charge is 0.48 ($/kW). Each one of the main and booster pump stations has an electricity meter and based on their location; they receive a $CO_2$ emission factor report from LEEM server that is presented in Table 11. This emission factor data are used as an offline source of emission data for all test scenarios. Except special cases that definition of a test scenario required to change the optimization option, it was tried to keep optimization options of all the test scenarios the same. Different parameters that are used as optimization options of the WDS of Monroe are listed in Table 11.

The skeletonized version of the Richmond WDS is used as the second test case in this study (van Zyl 2001). This WDS has over 22.69 (km) of distribution lines which range in size from 76 to 300 (mm). There is seven constant speed pump in six pump stations. The main pump station has two pumps that are connected to a reservoir as the only source of water. Each one of other five booster pump stations has only one pump. Nominal power of these pumps ranges from 3 to 60 (kW). Like Monroe city, ground level of Richmond has a slope from North West toward South East. The minimum and maximum elevation of the demand points are 60 and 242 (m) respectively. Figure 26 displays a model schematic of this WDS.

There are 6 water tanks in the system with total water storage capacity of 2598 ($m^3$) (66% of the daily demand). For filling tanks and answering water demands, in addition to the required pressure at demand point and dynamic head loss, pumps need to provide enough pressure to overcome 199 (m) of static head. The water demand of system is 3921 ($m^3$/day). Richmond WDS has only one demand pattern.

Table 11- Emission factor values that are used for all optimization scenarios of WDS of Monroe

| Time | $CO_2$ Emission Factor (kg/MWh) |
|---|---|
| 00:00 | 767.771 |
| 01:00 | 738.324 |
| 02:00 | 702.904 |
| 03:00 | 702.904 |
| 04:00 | 702.904 |
| 05:00 | 767.771 |
| 06:00 | 781.469 |
| 07:00 | 808.212 |
| 08:00 | 764.333 |
| 09:00 | 719.768 |
| 10:00 | 719.768 |
| 11:00 | 695.334 |
| 12:00 | 662.793 |
| 13:00 | 630.703 |
| 14:00 | 630.531 |
| 15:00 | 628.591 |
| 16:00 | 628.882 |
| 17:00 | 666.549 |
| 18:00 | 693.607 |
| 19:00 | 665.274 |
| 20:00 | 730.766 |
| 21:00 | 790.628 |
| 22:00 | 808.212 |
| 23:00 | 780.477 |

Table 12- Optimization options of test scenarios

| Parameter | Value |
|---|---|
| Optimization Duration (hr) | 24 |
| Optimization Time Step (min) | 60 |
| Maximum Number of Iterations | 300 |
| Maximum Number of Solution Evaluations | 16600 |
| Maximum Optimization Time (min) | 500 |
| Minimum Optimization Rate | 1% During 100 Iterations |
| Electricity Cost Goal ($) | 0 |
| Pollution Emission Goal (kg) | 0 |
| Total Penalty Goal | 0 |
| Population Size | 100 |
| Percentage of Elite Solution | 20% |
| Crossover Percentage | 50% |
| Crossover Rate | 50% |
| Mutation Percentage | 5% |

| Mutation Rate | 10% |
|---|---|

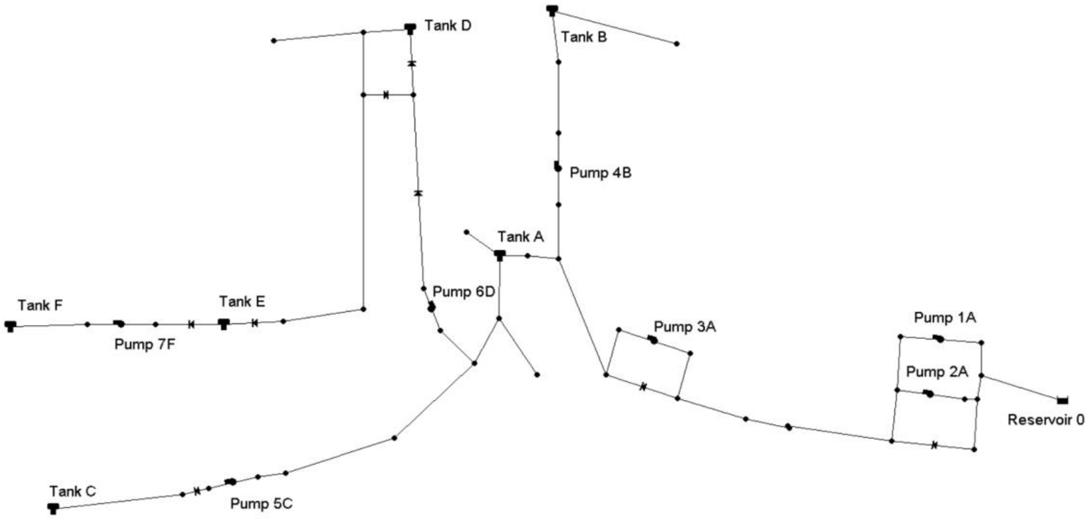



Figure 26- Model schematic WDS of Monroe

The skeletonized version of hydraulic model of the Richmond WDS has 41 junctions, 44 pipes, seven constant speed pumps, one reservoir, six tanks, and one 24 hour water demand pattern. The minimum and maximum hourly demand multipliers of the system are 0.39 and 1.53 respectively. The hydraulic simulation period of the model is also 24 hour with one hour time step. This EPANET model was used for evaluating different functionalities of PEPSO and has not been used for comparing PEPSO with DS. Table 13 and Table 14 presents constraints on the water level in tanks and water pressure at strategic junctions that was used in all optimization scenarios. It is shown in Table 14 *constraint importance multipliers* of all 10 strategic junctions are one that indicates to the same importance level for water pressure violation at all strategic junctions

Table 13- Constraints on water level in tanks of the Monroe WDS

| Tank ID | Elevation (m) | Water Capacity (m3) | Min. Allowed Water Level (m) | Min. Desired Water Level (m) | Max. Desired Water Level (m) | Max. Allowed Water Level (m) |
|---|---|---|---|---|---|---|
| A | 184.13 | 68.42 | 0.00 | 0.30 | 1.70 | 2.00 |
| B | 216.00 | 1461.69 | 0.00 | 0.50 | 2.86 | 3.37 |
| C | 258.90 | 230.75 | 0.00 | 0.32 | 1.79 | 2.11 |
| D | 241.18 | 679.87 | 0.00 | 0.55 | 3.10 | 3.65 |
| E | 203.01 | 135.21 | 0.00 | 0.44 | 2.29 | 2.69 |

| | | | | | | |
|---|---|---|---|---|---|---|
| F | 235.71 | 22.29 | 0.00 | 0.33 | 1.86 | 2.19 |

Table 14- Constraints on water pressure at strategic junctions of the Monroe WDS

| Strategic Junctions ID | Min. Allowed Pressure (psi) | Min. Desired Pressure (psi) | Max. Desired Pressure (psi) | Max. Allowed Pressure (psi) | Constraint Importance Multiplier |
|---|---|---|---|---|---|
| 42 | 0 | 20 | 140 | 200 | 1 |
| 1302 | 0 | 0 | 100 | 200 | 1 |
| 10 | 0 | 0 | 100 | 200 | 1 |
| 312 | 0 | 0 | 100 | 200 | 1 |
| 325 | 0 | 0 | 100 | 200 | 1 |
| 701 | 0 | 0 | 100 | 200 | 1 |
| 745 | 0 | 20 | 100 | 200 | 1 |
| 249 | 0 | 20 | 100 | 200 | 1 |
| 753 | 0 | 20 | 100 | 200 | 1 |
| 637 | 0 | 20 | 140 | 200 | 1 |

Similar to pump operation constraints of the WDS of Monroe, the maximum allowed number of pump switches in a day is 24 and the minimum duration of time between pump shut-down, and start-up is 15 minutes. The maximum allowed a continuous period of operation for the pump is 24 hours.

Each pump station of the Richmond WDS has a unique electricity tariff that just include the *energy consumption charge* (there is no *power demand charge* in Richmond WDS). On-peak hours of all tariffs start from 07:00 and end by 24:00. The *energy consumption charge* of all pumps is shown in Table 15. For Scenario Sc7 that *energy consumption charge* is constant throughout the day, off-peak rate is used for the whole 24 hour period. The same $CO_2$ emission factors that are presented in Table 12 and used for WDS of Monroe are used for Richmond WDS too. Except the maximum number of iterations and mutation percentage, all other optimization options of the WDS of Monroe that is listed in Table 11 are used for Richmond WDS optimization scenarios. Mutation percentage of Richmond WDS scenarios is 10%. Accordingly to keep the maximum number of solution evaluations of Richmond scenarios similar to scenarios of Monroe

(16600 solution evaluation), 275 iterations used as the maximum number of iteration of the Richmond scenarios.

Table 15- Energy consumption charge of pumps of the Richmond WDS

| Pump ID | On-Peak Rate ($/kWh) | Off-Peak Rate ($/kWh) |
|---|---|---|
| 1A | 0.0679 | 0.0241 |
| 2A | 0.0679 | 0.0241 |
| 3B | 0.0754 | 0.0241 |
| 4C | 0.1234 | 0.0246 |
| 5D | 0.0987 | 0.0246 |
| 6E | 0.1122 | 0.0246 |
| 7F | 0.1194 | 0.0244 |

Finally, it should be noted that in these tests emission factors just change in time, and we did not include any special variation for emission factors. However, as it was mentioned previously, emission factors that are reported by LEEM may vary due to change in location of energy consumption. We assumed that the area that is covered by both Monroe and Richmond WDS were not wide enough to change emission factor values based on the location of energy consumption. However, one can use PEPSO to optimize a WDS that its pump stations are far from each other. In this case, PEPSO can take advantage of the change in emission factors at different locations and find better solution by shifting location of energy consumption from one pump station to another one.

# CHAPTER 3 RESULTS AND DISCUSSIONS

## 3.1. Optimization Results

It was described in Section 2.3.1 that eight scenarios were used to evaluate different functionalities of PEPSO. Section 3.1.1 presents the result of these scenarios that have been tested on both Monroe and Richmond WDSs. Although PEPSO reports the result of the best solution and average result of all solutions of the final Pareto frontier, this section only presents results of the best solution of each optimization simulation. Section 0 shows obtained results from the comparison tests of PEPSO and DS. The result of PEPSO that is used in this section is also the best solution of each optimization run.

### 3.1.1. Results of PEPSO Functionality Evaluation Tests

Table 16 is used for reporting the result of all individual tests of all scenarios that have been conducted on Monroe WDS model. The name of each test is formed from two parts. The first part shows the name of the scenario and the second part shows the identical code of each test. These two parts are separated by a dash. Input data and optimization options and criteria of all different test of a scenario were the same. This table reports optimization results including, total electricity cost and its components, $CO_2$ emission, total penalty and its components.

In general CPU time of each optimization run of the Monroe WDS with the Lenovo ThinkPad W520 workstation (see Section 2.3.1 for specifications) are 1245±34 seconds (real time 02:14:44±00:03:43). This time for optimizing the skeletonized version of Richmond WDS is 287±13 seconds (real time 00:35:38±00:01:36).

Table 17 shows results of functionality evaluation tests on Richmond WDS.

Table 16- Results of PEPSO functionality evaluation tests on Monroe WDS

| Test Name | Electricity Cost ($) | Energy Consumption Cost ($) | Power Demand Cost ($) | $CO_2$ Emission (kg) | Total Penalty | Water Level Penalty at Tank | Pressure Penalty at Junction |
|---|---|---|---|---|---|---|---|
| Sc1-1 | 377.1 | 209.7 | 167.4 | 3534.1 | 26.5 | 26.1 | 0.4 |
| Sc1-2 | 359.1 | 216.4 | 142.7 | 3663.9 | 29.2 | 29.2 | 0.0 |
| Sc1-3 | 379.2 | 213.9 | 165.3 | 3617.6 | 25.5 | 25.3 | 0.2 |
| Sc1-4 | 368.7 | 215.6 | 153.1 | 3625.3 | 27.8 | 27.7 | 0.1 |
| Sc1-5 | 345.9 | 203.2 | 142.6 | 3474.8 | 195.9 | 33.0 | 162.9 |
| Sc2-1 | 368.1 | 204.2 | 163.9 | 3429.6 | 25.3 | 24.7 | 0.7 |
| Sc2-2 | 360.5 | 209.1 | 151.4 | 3553.1 | 27.5 | 25.3 | 2.2 |
| Sc2-3 | 372.6 | 215.8 | 156.8 | 3632.4 | 25.9 | 25.9 | 0.1 |
| Sc2-4 | 344.9 | 211.7 | 133.2 | 3589.9 | 29.6 | 29.6 | 0.0 |
| Sc2-5 | 363.9 | 213.6 | 150.3 | 3597.5 | 29.5 | 29.5 | 0.0 |
| Sc3-1 | 370.3 | 218.2 | 152.2 | 3688.4 | 26.1 | 26.0 | 0.1 |
| Sc3-2 | 362.6 | 213.0 | 149.6 | 3621.1 | 26.9 | 26.9 | 0.0 |
| Sc3-3 | 367.2 | 214.1 | 153.1 | 3632.9 | 25.3 | 24.1 | 1.1 |
| Sc3-4 | 360.3 | 214.4 | 146.0 | 3630.9 | 23.6 | 23.5 | 0.1 |
| Sc3-5 | 363.8 | 216.2 | 147.6 | 3692.1 | 23.1 | 23.1 | 0.0 |
| Sc4-1 | 375.7 | 214.1 | 161.6 | 3625.2 | 24.6 | 24.5 | 0.1 |
| Sc4-2 | 346.3 | 206.7 | 139.7 | 3501.5 | 29.5 | 29.0 | 0.5 |
| Sc4-3 | 387.1 | 217.8 | 169.4 | 3678.3 | 25.0 | 25.0 | 0.0 |
| Sc4-4 | 360.0 | 211.1 | 148.9 | 3568.4 | 27.4 | 27.4 | 0.0 |
| Sc4-5 | 367.0 | 214.9 | 152.1 | 3645.7 | 27.5 | 27.1 | 0.3 |
| Sc5-1 | 317.9 | 192.9 | 124.9 | 3263.6 | 31.0 | 29.1 | 1.9 |
| Sc5-2 | 333.4 | 198.0 | 135.3 | 3381.2 | 88.1 | 36.2 | 51.9 |
| Sc5-3 | 339.5 | 205.6 | 133.9 | 3481.9 | 29.5 | 28.6 | 1.0 |
| Sc5-4 | 336.1 | 200.5 | 135.6 | 3396.0 | 49.5 | 45.6 | 3.9 |
| Sc5-5 | 336.1 | 211.0 | 125.1 | 3589.9 | 32.0 | 32.0 | 0.0 |
| Sc6-1 | 261.4 | 142.9 | 118.5 | 2408.9 | 1227.1 | 150.6 | 1076.5 |
| Sc6-2 | 253.6 | 140.2 | 113.4 | 2371.8 | 1767.0 | 155.8 | 1611.2 |
| Sc6-3 | 273.4 | 148.8 | 124.6 | 2515.0 | 615.7 | 149.1 | 466.7 |
| Sc6-4 | 282.5 | 155.8 | 126.7 | 2637.7 | 717.5 | 148.4 | 569.1 |
| Sc6-5 | 293.2 | 157.6 | 135.6 | 2653.6 | 570.3 | 143.0 | 427.3 |
| Sc7-1 | 205.5 | 205.5 | 0.0 | 3538.0 | 28.0 | 28.0 | 0.0 |
| Sc7-2 | 203.0 | 203.0 | 0.0 | 3519.7 | 22.5 | 22.4 | 0.2 |
| Sc7-3 | 217.3 | 217.3 | 0.0 | 3778.7 | 23.8 | 23.7 | 0.0 |
| Sc7-4 | 206.7 | 206.7 | 0.0 | 3600.4 | 27.5 | 26.6 | 0.9 |
| Sc7-5 | 198.2 | 198.2 | 0.0 | 3421.2 | 35.6 | 31.7 | 3.9 |

Table 17- Results of PEPSO functionality evaluation tests on Richmond WDS

| Test Name | Electricity Cost ($) | Energy Consumption Cost ($) | Power Demand Cost ($) | $CO_2$ Emission (kg) | Total Penalty | Water Level Penalty at Tank | Pressure Penalty at Junction |
|---|---|---|---|---|---|---|---|
| Sc1-1 | 92.9 | 92.9 | 0.0 | 1098.9 | 6.6 | 6.6 | 0.0 |
| Sc1-2 | 91.2 | 91.2 | 0.0 | 1067.7 | 6.6 | 6.6 | 0.0 |
| Sc1-3 | 92.2 | 92.2 | 0.0 | 1085.6 | 6.6 | 6.6 | 0.0 |
| Sc1-4 | 92.3 | 92.3 | 0.0 | 1079.0 | 7.1 | 7.1 | 0.0 |
| Sc1-5 | 92.0 | 92.0 | 0.0 | 1097.3 | 6.4 | 6.4 | 0.0 |
| Sc2-1 | 81.2 | 81.2 | 0.0 | 941.9 | 95.8 | 5.4 | 90.3 |
| Sc2-2 | 74.9 | 74.9 | 0.0 | 879.7 | 127.8 | 5.6 | 122.3 |
| Sc2-3 | 65.4 | 65.4 | 0.0 | 795.3 | 216.2 | 5.8 | 210.4 |
| Sc2-4 | 91.2 | 91.2 | 0.0 | 1099.3 | 7.3 | 7.3 | 0.0 |
| Sc2-5 | 66.3 | 66.3 | 0.0 | 809.2 | 215.9 | 5.5 | 210.4 |
| Sc3-1 | 95.1 | 95.1 | 0.0 | 1111.7 | 6.1 | 6.1 | 0.0 |
| Sc3-2 | 94.2 | 94.2 | 0.0 | 1102.3 | 6.3 | 6.3 | 0.0 |
| Sc3-3 | 92.5 | 92.5 | 0.0 | 1093.7 | 6.4 | 6.4 | 0.0 |
| Sc3-4 | 92.1 | 92.1 | 0.0 | 1092.7 | 6.5 | 6.5 | 0.0 |
| Sc3-5 | 95.1 | 95.1 | 0.0 | 1098.0 | 6.1 | 6.1 | 0.0 |
| Sc4-1 | 50.6 | 50.6 | 0.0 | 544.8 | 434.4 | 5.4 | 429.0 |
| Sc4-2 | 57.9 | 57.9 | 0.0 | 697.1 | 340.0 | 5.5 | 334.5 |
| Sc4-3 | 62.7 | 62.7 | 0.0 | 737.2 | 275.8 | 5.6 | 270.1 |
| Sc4-4 | 51.0 | 51.0 | 0.0 | 678.5 | 367.2 | 5.8 | 361.4 |
| Sc4-5 | 63.2 | 63.2 | 0.0 | 731.2 | 276.9 | 5.4 | 271.4 |
| Sc5-1 | 88.0 | 88.0 | 0.0 | 1079.4 | 7.1 | 7.1 | 0.0 |
| Sc5-2 | 91.0 | 91.0 | 0.0 | 1091.3 | 6.6 | 6.6 | 0.0 |
| Sc5-3 | 91.3 | 91.3 | 0.0 | 1084.7 | 6.8 | 6.8 | 0.0 |
| Sc5-4 | 92.0 | 92.0 | 0.0 | 1122.9 | 6.5 | 6.5 | 0.0 |
| Sc5-5 | 93.1 | 93.1 | 0.0 | 1082.8 | 6.2 | 6.2 | 0.0 |
| Sc6-1 | 87.2 | 87.2 | 0.0 | 1086.8 | 8.8 | 8.8 | 0.0 |
| Sc6-2 | 86.5 | 86.5 | 0.0 | 1049.8 | 8.0 | 8.0 | 0.0 |
| Sc6-3 | 86.6 | 86.6 | 0.0 | 1055.0 | 8.5 | 8.5 | 0.0 |
| Sc6-4 | 86.3 | 86.3 | 0.0 | 1047.4 | 8.5 | 8.5 | 0.0 |
| Sc6-5 | 87.6 | 87.6 | 0.0 | 1071.9 | 11.1 | 11.1 | 0.0 |
| Sc7-1 | 119.9 | 119.9 | 0.0 | 1080.2 | 6.1 | 6.1 | 0.0 |
| Sc7-2 | 119.0 | 119.0 | 0.0 | 1082.3 | 6.2 | 6.1 | 0.1 |
| Sc7-3 | 123.3 | 123.3 | 0.0 | 1111.6 | 6.1 | 6.1 | 0.0 |
| Sc7-4 | 112.6 | 112.6 | 0.0 | 1030.5 | 6.6 | 6.5 | 0.2 |
| Sc7-5 | 111.8 | 111.8 | 0.0 | 1025.1 | 8.6 | 8.0 | 0.6 |

In Section2.3.1 it was mentioned that in the last test scenario (Sc8), EPANET hydraulic solver replaced with the ANN trainer module of PEPSO to use the ANN-based metamodel instead of high fidelity EPANET hydraulic model during the optimization process. Results of this test were not satisfactory. The tests on both Monroe and Richmond WDSs showed that although PEPSO was able to train an ANN and use that instead of a high fidelity hydraulic model, the accuracy of trained ANN was not enough for optimization purpose. Accordingly using the trained ANN as hydraulic solver did not help the optimization algorithm to get closer to the global optimum point. In addition to the fact that the trained ANN was not accurate enough, the ANN training process was a time-consuming and complicated process. PEPSO II uses multiple ANN for modeling WDS. Each ANN trained based on a set of inputs that includes initial water level in tanks, status, and speed of pumps and the water demand multiplier. Each ANN returns just one output (e.g. final level of a tank or power demand of a pump). Although theoretically it is possible to train an ANN that can provide multiple outputs we used an ANN for each required output to increase the accuracy of the results and simplify the training process. Therefore, number of required trained ANN for each WDS is calculated with Equation 16:

$$\mathrm{No.\,of\,required\,ANNs} = \mathrm{n} + \mathrm{No.\,of\,pumps} + \mathrm{No.\,of\,strategic\,junctions} + \mathrm{No.\,of\,strategic\,pipes} + (\mathrm{No.\,of\,pumps} \times \mathrm{m}) \qquad \text{Equation 16}$$

Where,

$\mathrm{n}$ is equal to 1 (for negative pressure warning)

$\mathrm{m}$ is equal to 6 (3 for pump flow, head and power demand, 3 for flow, head and connection warning of pumps)

Based on Equation 16, 86 and 59 ANNs need to be trained for Monroe and Richmond WDSs respectively. For training each ANN of Monroe WDS, a training set with 17 inputs and one output and 10000 training points created. Each training point was a set of inputs before a time block and one output after one time step. For training each ANN of Skeletonized Richmond WDS, a training set with 14 inputs and one output and 10000 training points, created. With the Lenovo ThinkPad W520 workstation (see Section 2.3.1 for more specification) CPU time of creating a training set with 10000 training point for Monroe and skeletonized version of Richmond WDSs are 50 and 18 (their real time are 623 and 223 seconds) respectively. CPU time of training all required ANN of these two WDS with the above-mentioned computer are 10.2 and 2.2 minutes (their real time are 81.5 and 17.5 minutes) respectively.

Although training of ANNs needed a considerable amount of time, after training them for one optimization run, the same ANNs could be used for other optimization runs. The biggest problem about training an ANN is adjusting its training parameters. Different variables need to be adjusted before training an ANN, and all of them affects the quality of results of the trained ANN. We realized that there was not any defined way to select the proper set of ANN training parameter for a specific WDS. So various combinations of possible values of parameters were used for training ANN and the best set of parameters that result in more accurate ANN selected for each specific WDS.

For the ANNs that were providing final water level in tanks, pressure at junctions and flow, head and power demand of pumps MSE stop function was used. However for the ANNs that were determining if pumps have flow, head and connection warning or if the system has negative pressure warning, Bit stop function was used. The result of the first group of ANNs were real numbers but the result of the second group of ANN was binary values. The linear error function was used for training all ANNS. Some ANN were trained better with sigmoid symmetric activation function for both hidden and output layers. However, other were trained better with sigmoid symmetric activation function for hidden layer and linear activation function for the output layer. All ANNs were designed with shortcut structure and trained by the standard method and incremental algorithm (see section 2.2.3.3 for more information). Other ANN training option that was used for training all ANN are listed in Table 18.

Table 18- Parameter of training ANN with FANN library

| Parameter | Value |
|---|---|
| Maximum epochs | 1000 |
| Minimum weight | -0.5 |
| Maximum weight | 0.5 |
| Hidden layer activation steepness | 0.05 |
| Output layer activation steepness | 0.05 |
| Learning rate | 0.01 |
| Maximum acceptable error | 0.001 |
| Input to hidden layer ratio | 2 |

After training ANN and using them instead of EPANET solver, it was observed that the optimization process gets 6.96 and 2.01 times faster for Monroe and Richmond WDSs respectively. However, it faced a problem when optimization progressed, and PEPSO started to search for the optimum solution in undiscovered regions of the solution space. As initially trained ANNs were not trained for providing accurate hydraulic results for newly discovered areas of the solution space, they could not provide accurate results and it prevented PEPSO to get closer to the optimum solution. Although the results of optimization tests with ANN metamodel was not satisfactory, this results made it possible to create a list of suggestion that can be investigated in further studies to get closer to using ANN metamodel and speeding up the optimization process while accuracy is not sacrificed.

### 3.1.2. Results of PEPSO and DS Comparison Tests

The results of all PEPSO and DS comparison tests are displayed in Table 19. Before comparing this results, the best pump schedule that is reported by PEPSO at the end of optimization process and the final result of DS used to create two scenarios in WaterGEMS software. By this method, the same software was used for hydraulic simulation of the WDS based on all proposed optimum pump schedules and the same method of reporting result was used for obtaining required values for comparisons. Therefore, all results that are presented in Table 19 are outputs of WaterGEMS after running the hydraulic model by optimized pump schedule of PEPSO and DS. Optimization options of DS let us define a random seed to randomize initial point of the optimization process. If we use the same random seed for all optimization runs, we will always get the same results. Therefore, in these tests different random seed were used. Like PEPSO, it gives DS a possibility of starting optimization from different areas of the solution space.

Table 19- Results of PEPSO and DS comparison tests

| Test Name | Energy Consumption (kWh) | On-peak Energy Consumption (kWh) | Off-peak Energy Consumption (kWh) | Peak Power Demand (kW) | Power Demand Cost ($) | Pressure Penalty at Junction |
|---|---|---|---|---|---|---|
| PEPSO 1 | 6810.3 | 2609.8 | 4200.5 | 418.5 | 199.8 | 0.00 |
| PEPSO 2 | 6662.5 | 2266.9 | 4395.6 | 349.1 | 166.6 | 0.00 |
| PEPSO 3 | 6475.5 | 2374.1 | 4101.4 | 385.4 | 184.0 | 0.04 |
| PEPSO 4 | 6144.4 | 2222.7 | 3921.7 | 400.4 | 191.1 | 0.00 |
| PEPSO 5 | 6309.2 | 2464.4 | 3844.8 | 410.7 | 196.0 | 0.07 |
| DS 1 | 7049.9 | 2359.7 | 4690.2 | 500.5 | 238.9 | 196.8 |
| DS 2 | 7049.9 | 2359.7 | 4690.2 | 500.5 | 238.9 | 196.8 |
| DS 3 | 7049.9 | 2359.7 | 4690.2 | 500.5 | 238.9 | 196.8 |
| DS 4 | 7702.0 | 3035.1 | 4666.9 | 599.3 | 286.1 | 203.0 |
| DS 5 | 7797.9 | 2868.7 | 4929.2 | 731.8 | 349.3 | 229.2 |

## 3.2. Analysis and Discussions

### 3.2.1. Analyzing PEPSO Functionality Evaluation Tests Results On Monroe WDS

The effect of change of stored water in tanks during operation period on total electricity usage of a system is an important factor that needs to be considered before comparing results. Assuming that energy consumption (kWh) of two proposed pump schedules are the same, but in the first one final volume of stored water in tanks is higher than the initial volume of stored water and in the other one the initial and final volume of stored water are equal. In this scenario, the first pump schedule is better that the second one in respect to the *net energy consumption*. Net energy consumption is total energy consumption of the system considering the stored or drained energy of the system due to change in volume of stored water in elevated tanks. When final level of water in a tank is higher than the initial level, it shows water accumulation in the elevated tank that can be comprehended as energy accumulation in the system. Vice versa, draining water from an elevated tank is equal to draining energy from the system. Accordingly, to be fair while comparing energy consumption of two optimization scenarios, we should take into account the amount of energy accumulation or draining of tanks. For calculating accumulated or drained energy, at first, we need to calculate the change in volume of stored water in elevated tanks. Then the average *Energy Intensity* (EI) of the system should be calculated. EI is the average amount of energy needed to transport water from source to demand points per unit of volume of water (kWh/m$^3$). Multiplying the volume change of stored water (m$^3$) by EI (kWh/m$^3$) gives the amount of accumulated energy in the system or drained energy from the system. Note that the negative volume change (less volume of the final stored water in respect to the initial volume) results in negative energy values that show energy draining. Subtracting the calculated energy change from the total energy consumption of the system will result in the *net energy consumption* of the system. Therefore, draining tanks during an operation cycle, increases the *net energy consumption* of the system. The value of energy change due to change in volume of stored water can be multiplied by *average energy consumption charge* to calculate energy consumption cost change of the system. Subtracting this cost change from the total energy consumption cost provides the *net energy consumption cost*. The *average energy consumption charge* is a weighted average of on-peak and off-peak energy consumption charges based on the length of on-peak and off-peak periods of an electricity tariff. The similar method can be used for calculating change in pollution emission due to change in volume of stored water and then use it to calculate the net pollution emission of the system. It should be noted that as it was explained in section 2.2.2.4 , the net pollution emission does not show the total pollution emission of the real system. However, it can be used to calculate the change in total pollution emission of the system due to different scenarios.

The second to fifth columns of Table 20 (from left) present raw electricity related reports of PEPSO for different scenarios. The ninth column shows the percentage of change of the total stored water in elevated tanks of the system. The last four columns show *net energy consumptions (kWh)*, net energy consumption costs ($), net electricity costs ($) and net $CO_2$ emission of the system. Instead of using raw electricity related results of PEPSO, values of these last three columns are used for comparing results of different scenarios.

Table 20- Electricity consumption results of PEPSO functionality evaluation tests on Monroe WDS

| Test Name | Off-Peak Energy Consumption (kWh) | On-Peak Energy Consumption (kWh) | Peak Power Demand (kW) | Total Unadjusted Energy Consumption (kWh) | Stored Water Volume Change (%) | Net Energy Consumption (kWh) | Net Energy Consumption Cost ($) | Net Electricity Cost ($) | Net $CO_2$ Emission (kg) (Based on marginal Emission factors) |
|---|---|---|---|---|---|---|---|---|---|
| Sc1-1 | 3048.0 | 1915.5 | 350.3 | 4963.5 | -6% | 4994.8 | 211.0 | 378.4 | 3556.4 |
| Sc1-2 | 3371.9 | 1767.8 | 298.49 | 5139.7 | 4% | 5116.1 | 215.4 | 358.1 | 3647.1 |
| Sc1-3 | 3285.1 | 1791.8 | 345.75 | 5076.9 | 0% | 5074.9 | 213.8 | 379.1 | 3616.2 |
| Sc1-4 | 3180.2 | 1927.2 | 320 | 5107.4 | -6% | 5143.5 | 217.1 | 370.2 | 3651.0 |
| Sc1-5 | 3231.6 | 1599.0 | 298.39 | 4830.6 | -8% | 4873.3 | 205.0 | 347.7 | 3505.5 |
| Sc2-1 | 2805.6 | 2017.9 | 342.84 | 4823.5 | -5% | 4851.1 | 205.4 | 369.3 | 3449.2 |
| Sc2-2 | 3162.4 | 1797.2 | 316.17 | 4959.6 | 8% | 4914.1 | 207.2 | 358.6 | 3520.5 |
| Sc2-3 | 3173.5 | 1937.4 | 327.99 | 5110.9 | 12% | 5042.4 | 212.9 | 369.7 | 3583.8 |
| Sc2-4 | 3255.3 | 1767.9 | 277.85 | 5023.2 | 9% | 4971.6 | 209.5 | 342.7 | 3553.0 |
| Sc2-5 | 3017.1 | 2034.3 | 314.39 | 5051.4 | 6% | 5016.6 | 212.1 | 362.4 | 3572.8 |
| Sc3-1 | 3410.5 | 1771.2 | 317.85 | 5181.7 | 15% | 5097.8 | 214.6 | 366.8 | 3628.7 |
| Sc3-2 | 3187.7 | 1860.9 | 312.91 | 5048.6 | 11% | 4990.2 | 210.5 | 360.1 | 3579.2 |
| Sc3-3 | 3281.7 | 1798.2 | 320.36 | 5080.0 | 9% | 5029.0 | 211.9 | 365.1 | 3596.4 |
| Sc3-4 | 3206.3 | 1875.5 | 305.36 | 5081.8 | -1% | 5088.9 | 214.7 | 360.6 | 3636.0 |
| Sc3-5 | 3397.5 | 1739.2 | 308.72 | 5136.7 | 6% | 5104.2 | 214.9 | 362.4 | 3668.7 |
| Sc4-1 | 3200.5 | 1874.0 | 337.12 | 5074.5 | 6% | 5042.3 | 212.7 | 374.3 | 3602.2 |
| Sc4-2 | 3091.5 | 1807.5 | 291.98 | 4899.0 | -11% | 4958.3 | 209.2 | 348.8 | 3543.9 |
| Sc4-3 | 3242.5 | 1918.1 | 353.42 | 5160.6 | 3% | 5144.2 | 217.1 | 386.4 | 3666.6 |
| Sc4-4 | 3112.0 | 1888.4 | 311.09 | 5000.4 | 13% | 4931.6 | 208.2 | 357.1 | 3519.3 |
| Sc4-5 | 3259.1 | 1838.6 | 318.23 | 5097.7 | 14% | 5021.5 | 211.7 | 363.8 | 3591.2 |
| Sc5-1 | 2937.8 | 1639.4 | 261.38 | 4577.2 | -26% | 4709.7 | 198.5 | 323.5 | 3358.0 |
| Sc5-2 | 3225.4 | 1486.1 | 283.13 | 4711.5 | -16% | 4795.0 | 201.5 | 336.9 | 3441.2 |
| Sc5-3 | 3106.8 | 1768.7 | 280.17 | 4875.5 | -13% | 4944.0 | 208.5 | 342.4 | 3530.8 |
| Sc5-4 | 3060.5 | 1696.1 | 282.52 | 4756.6 | -9% | 4805.5 | 202.6 | 338.2 | 3430.9 |
| Sc5-5 | 3281.9 | 1727.6 | 261.74 | 5009.5 | -7% | 5045.7 | 212.5 | 337.6 | 3615.8 |
| Sc6-1 | 2090.1 | 1294.5 | 247.8 | 3384.6 | -35% | 3521.4 | 148.7 | 267.2 | 2506.3 |
| Sc6-2 | 2094.4 | 1228.1 | 237.32 | 3322.4 | -39% | 3469.1 | 146.4 | 259.8 | 2476.5 |
| Sc6-3 | 2177.3 | 1346.7 | 260.61 | 3524.0 | -37% | 3672.6 | 155.1 | 279.7 | 2621.0 |
| Sc6-4 | 2335.8 | 1357.1 | 264.7 | 3692.8 | -35% | 3840.9 | 162.0 | 288.7 | 2743.5 |
| Sc6-5 | 2326.2 | 1407.8 | 280.03 | 3734.0 | -31% | 3865.0 | 163.1 | 298.7 | 2746.7 |
| Sc7-1 | 3093.9 | 1909.0 | 345.09 | 5002.9 | -10% | 5055.8 | 207.7 | 207.7 | 3575.4 |
| Sc7-2 | 3169.5 | 1771.3 | 379.8 | 4940.8 | -1% | 4947.4 | 203.2 | 203.2 | 3524.4 |
| Sc7-3 | 3280.4 | 2009.0 | 373.61 | 5289.3 | 7% | 5247.6 | 215.6 | 215.6 | 3748.9 |
| Sc7-4 | 3125.2 | 1906.1 | 317.82 | 5031.3 | 6% | 5000.9 | 205.4 | 205.4 | 3578.6 |
| Sc7-5 | 2901.0 | 1922.8 | 353.68 | 4823.8 | -14% | 4901.1 | 201.3 | 201.3 | 3476.1 |

The process of calculating average and SEM of results of all similar tests of a scenario was described in Section 2.3.1. The average net electricity cost values of all scenarios are displayed in Figure 27. The SEM values are showed as error bars on top of each column. Figure 28 and Figure 29 also demonstrate the average and SEM of net $CO_2$ emission and total penalty values of all scenarios.

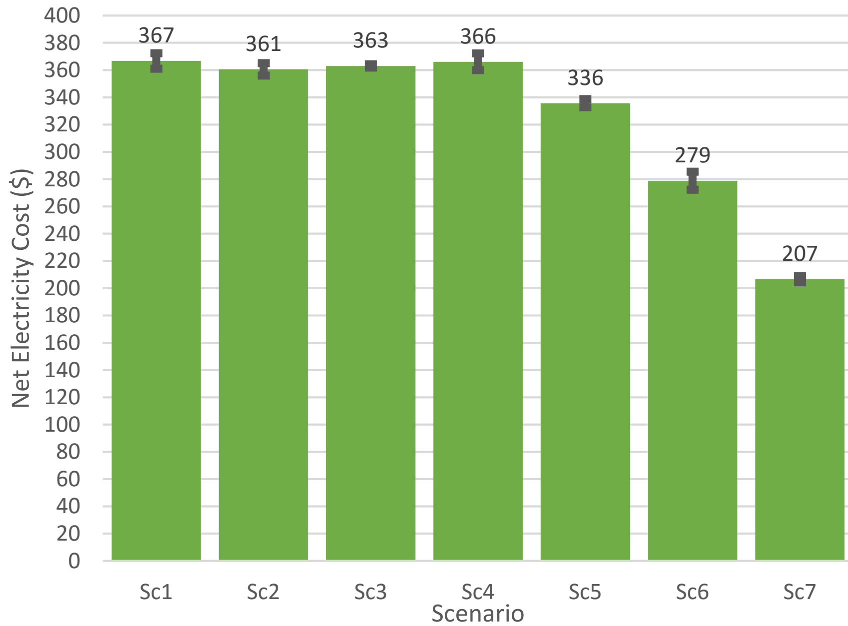



Figure 27- Net electricity cost of PEPSO functionality evaluation tests on Monroe WDS

At the first look, it seems that SEMs of net electricity cost (Figure 27) and $CO_2$ emission (Figure 28) data are relatively low, but SEMs of total penalties (Figure 29) are higher than other objectives. However, it should be noted that water level and pressure violations are raised to the power of 1.5 to calculate penalties which mean, a slight change in violation may result in a considerable change in the penalty so it is expected to see higher variation and larger SEM for a penalty with respect to two other objectives.

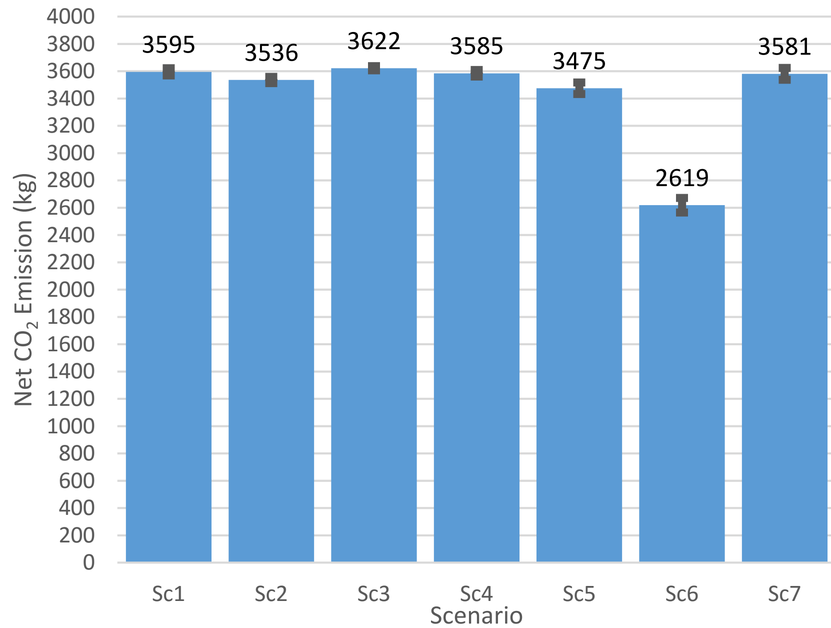



Figure 28- Net $CO_2$ emission of PEPSO functionality evaluation tests on Monroe WDS (calculated based on marginal emission factors)

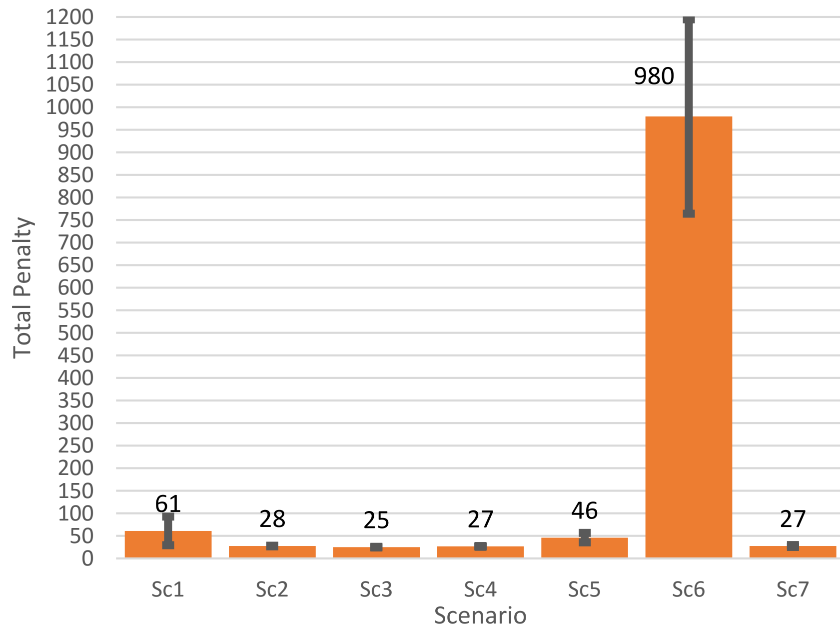



Figure 29- Total penalty of PEPSO functionality evaluation tests on Monroe WDS

Based on results of the base scenario (Sc1), average daily electricity cost of pump stations of Monroe WDS is \$367±\$6. About 58% of this cost is related to energy consumption, and the remaining 42% is peak power demand cost. From the total net

5041±46 (kWh) energy consumption of the system, about 36% is consumed during on-peak hours. Electricity cost of scenarios Sc2, Sc3 and Sc4, are very close to results of Sc1 optimization runs. Which indicates, in this test case, if WDS get optimized based on one objective (e.g. penalty) all other objectives will also be optimized in some extent. As reducing energy consumption in most cases will cause reduction of $CO_2$ emission this direct relation between electricity cost objective and $CO_2$ emission objective is predictable. But reducing penalty and reducing energy usage are not always aligned. Taking a closer look at detailed results of this test case showed that most of the penalties of Sc1 tests are related to high tank level violation. Therefore, it is understandable that in this specific case, reducing energy usage can reduce the total penalty. Accordingly optimizing based on all three objectives (Sc2) results in almost the similar to Sc1 optimum solution, with a similar amount of electricity cost.

Similar to electricity cost results, $CO_2$ emission results of optimized solutions of Sc1, Sc2, Sc3 and Sc4 scenarios are relatively close. However $CO_2$ emission of scenario Sc2 and Sc4 that consider pollution emission as an optimization objective is slightly lower than Sc1 and Sc3 scenarios. It is interesting that when both electricity cost and $CO_2$ emission optimized at the same time (Sc2), highest reduction in $CO_2$ emission obtained (48 to 123 (kg/day) less than pollution emission of the system when is not optimized based on these two objectives). Assuming that the average emission reduction for the system is possible throughout a year, we can see that this small difference in emission can lead to 31.2 ton reduction of $CO_2$ emission of Monroe WDS per year.

The amount of $CO_2$ emission is a function of both energy consumption (kWh) and time variant emission factor (kg/kWh). Therefore optimizing both electricity cost and $CO_2$

emission at the same time (Sc2) can have amplified effect on reducing $CO_2$ emission by reducing the total energy consumption and shifting it to the times with lower emission factors. Accordingly, the test results showed that optimizing the system based on both energy consumption and $CO_2$ emission (Sc2) can reduce the daily $CO_2$ emission by 1.4±1.3% in respect to $CO_2$ emission of Sc4 scenario that is optimized based on $CO_2$ emission (not electricity cost). This result indicates that, although theoretically optimizing based on only $CO_2$ emission should show us a solution with the minimum weight of emitted $CO_2$, but in practice, optimizing based on multiple objectives that amplify the effect of each other may result in finding better solutions in limited duration of optimization.

In general, total penalty values of most scenarios (except Sc4) are low. However when Monroe WDS optimized just based on penalties (Sc3), the total penalty value is 10±7% lower than the total penalty of Sc2 scenario which optimizes all three objectives.

Comparing results of the optimization without using UI calculation (Sc5) with the base scenario (Sc1) showed that the electricity cost of Sc5 scenario is 8.5±2.3% lower than the base scenario (Sc1). At first, this result suggests that using UI calculation reduced the effectiveness of PEPSO and results in solutions with higher energy consumption. However, more investigation revealed that during the whole operation period, stored volume of water in tanks and pressure of strategic junctions in Sc5 solutions are in average 5.6% and 1.9% lower than Sc1 results. In addition, solutions of Sc1 scenario in average have less than 2 warnings about pumps that cannot deliver head, but Sc5 results in average have about 4 and 1 warnings about pumps that cannot deliver head and flow respectively. The final volume of stored water in tanks for the Sc5 scenario is 10.9±5.3% lower than final volume of stored water in the Sc1 scenario.

Figure 30 displays pattern of water level in tanks (top) and water pressure at strategic junctions (bottom) of typical results of Sc1 (left) and Sc5 (right) scenarios. It can be seen than Sc3 solutions tends to drain tanks more than Sc1 solutions and water pressure at strategic junctions in Sc5 solutions are slightly lower than the pressure of Sc1 solutions. The UI module of PEPSO calculates UI values to help PEPSO to find better solutions that are more practical and cause less warning message during simulation with EPANET. Effect of this module on optimization of Monroe WDS is preventing to drain tanks and keeping water pressure at the strategic junction in the acceptable range but not very close to the minimum limits. This effect might be more desirable for operators that want to stay on the safe side and prevent to operate the system in an extreme way that reduces electricity cost but is sensitive to potential changes in demands.

All of these show that although electricity cost of Sc5 result is lower than Sc1 but results of Sc1 scenario are more safe and practical and can better satisfy operation needs of the system. So in this test case, calculating UI helped PEPSO to find more practical optimized solutions. The required CPU time of optimizing Sc1 scenario of the Monroe WDS with the Lenovo ThinkPad W520 workstation (see Section 2.3.1 for more specification) was 1152±97 seconds (real time 02:04:41±00:10:32). While turning off the UI calculator module (Sc5) reduce this CPU time to 1357±69 seconds (real time 02:26:53±00:07:26). So optimizing with UI can in average increase the optimization time of Monroe WDS by 8.9% but it increase the quality of the final solution.

It was explained in the section 2.3.1 that water level constraint in the tank was removed from the Sc6 optimization runs. Results of this optimization run showed that giving PEPSO the possibility of operating pumps without tank level constraints, reduces electricity cost and $CO_2$ emission of the system by 24±2.4% and 27.2±2.0%. Despite the fact that removing water level constraints reduces electricity cost and $CO_2$ emission, it considerably increased water level violation of tanks and water pressure violation at strategic junctions. In most of the scenarios, PEPSO was successful to find near-optimum solutions with relatively low amount of water level and pressure violations.

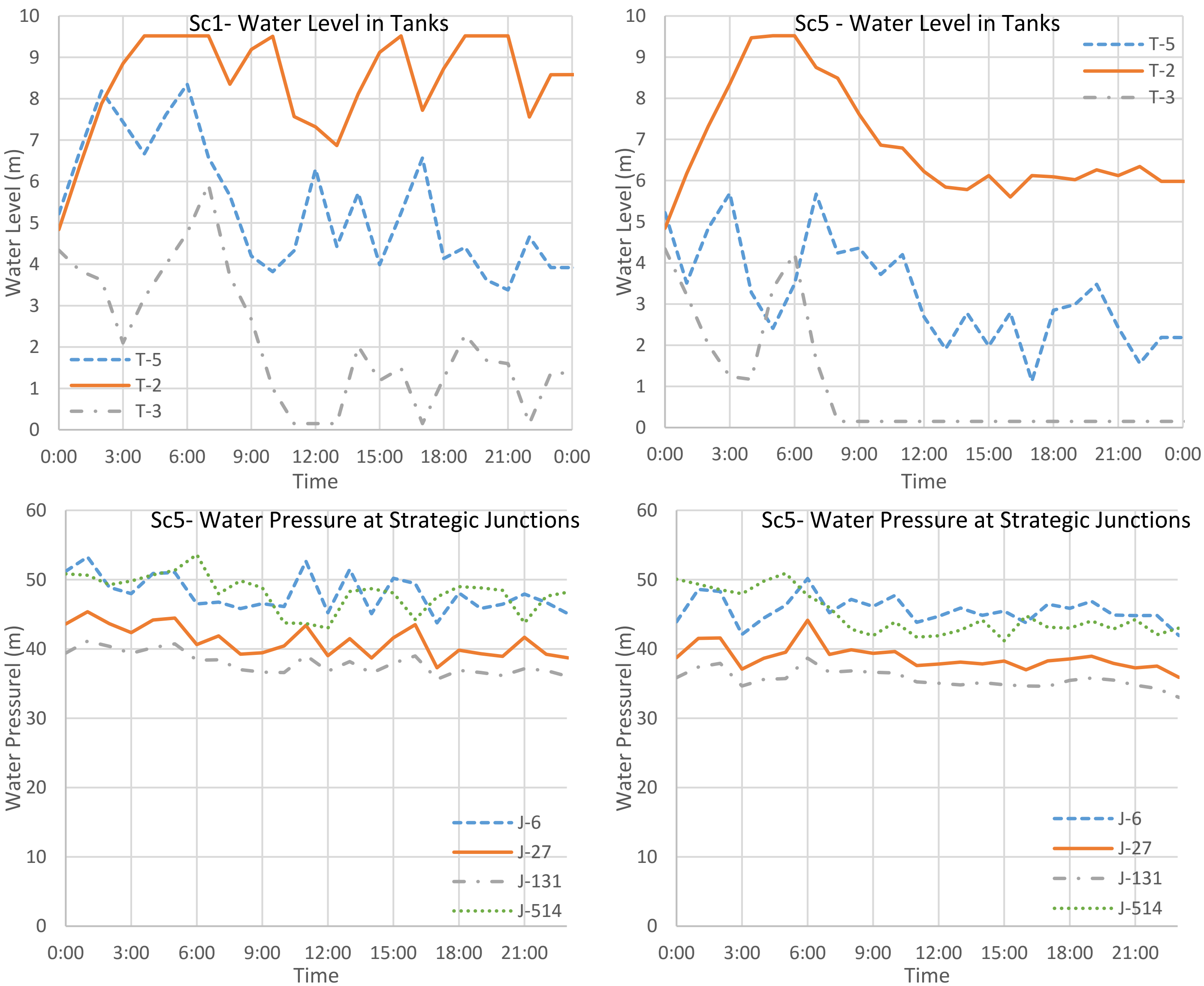



Figure 30- Pattern of water level in tanks (top) and water pressure at strategic junctions (bottom) of typical results of Sc1 (left) and Sc5 (right) optimization scenarios of Monroe WDS

However, in Sc6 scenario pressure at junctions has considerable fluctuation that caused considerable low and high-pressure penalties. The tank level penalty of Sc6 scenario is more than four times of tank level penalty of Sc1 scenario Comparing time patterns of water level in tanks and water pressure at junctions of Sc6 (Figure 31), and Sc1 (Figure 30) can clearly show this difference.

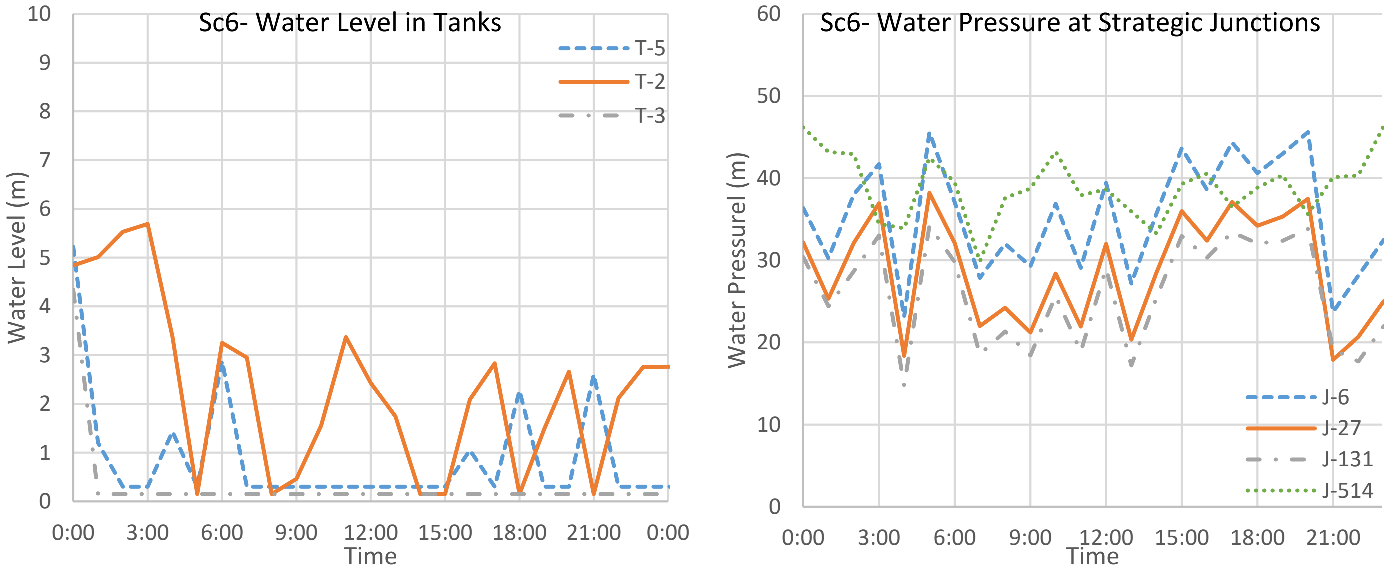



Figure 31- Pattern of water level in tanks (left) and water pressure at strategic junctions (right) of typical results of Sc6 optimization scenario of Monroe WDS

Although water level limits and constraint is removed from scenario Sc6 but it should be noted that even in that scenario, PEPSO tries to keep the final level of water in tank equal or above the initial water level. However test results showed that trying to keep the final tank level balanced is not enough for preventing draining tanks during the whole operation period, and it might result in pressure fluctuation and pressure deficit as it is shown in Figure 31. Operating Monroe WDS based on optimized results of the base scenario (Sc1) decrease the final volume of store water in tanks 3.1±2.2%. But operating the same system with results of Sc6 optimization runs decrease the final volume by 35.4±1.2%. It shows that constraining water levels of tanks prevented more than 32%

reduction in the volume of stored water in tanks. Stored water loss of Sc6 scenario is more drastic than all other scenarios and it can be directly related to the tank level control.

In all the optimization scenarios that are discussed up to this point, a time of use electricity tariff was used for calculating the electricity consumption cost. However for Sc7 optimization runs, a constant energy consumption cost is used for the whole operation period, and peak power demand charge was removed from the electricity tariff. Results of this test on Monroe WDS showed that having a flat rate electricity tariff, in average, can lead to 9.7% increase in peak power demand (kW) while the total consumed energy (kWh) is almost unchanged. Although the total energy consumption in both Sc1 and Sc7 scenarios are almost unchanged, 2.1% of the total energy consumption in Sc7 scenario shifted from off-peak hours to on-peak hours. These results confirms that power demand charge and time of use electricity tariffs will force PEPSO to find optimized solution with more energy consumption during off-peak times and with shaved peak power demand. Repeating the same scenario with flat rate energy tariff but including the peak power demand charge led to an optimized solution with the peak power demand equal to the base scenario (Sc1) but 1.7% more energy consumption during on-peak hours. Even in this case, that power demand charge was not removed from the electricity tariff, 1.7% energy consumption shifting from off-peak hours to on-peak hours is observed. Therefore, it can be confirm that most of the energy consumption shift in the Sc7 scenario were because of flat electricity charge not removing the power demand charge. The time of use energy tariff can be considered as a limiting factor that forces PEPSO to find solutions which consume more energy during off-peak hours.

### 3.2.2. Analyzing PEPSO Functionality Evaluation Tests Results On Richmond WDS

The same process that was done on results of Monroe test is done on the result of Richmond test to calculate Net energy consumption and $CO_2$ emission. The calculated net values are reported in Table 21.

Based on the optimum result of base scenario (Sc1), the daily net electricity cost of the Richmond skeletonized WDS is \$111.4±\$2.5. In average 68% of energy is consumed during on-peak hours. The total penalty of the optimum solution is almost negligible (6.6±0.1) and in most case are related to high water level violation of tank E. Water level in this tank almost always stays above the desired tank level and slightly below the maximum allowed level. Although this might show a sign of some excess energy consumption in the system, but from an operational point of view, it will not cause serious concerns like those cases that tanks are empty.

Figure 32 displays net electricity cost of the optimum solutions of different scenarios of Richmond WDS optimization test. The error bars on top of column show SEM of each column. Similarly, Figure 33 and Figure 34 presents net $CO_2$ emission and total penalty values of these tests.

From these graphs it can be grasped that SEMs of net electricity cost (Figure 32), $CO_2$ emission (Figure 33) and total penalty (Figure 34) of Richmond test in most scenarios except Sc2 and Sc4 are relatively low. Both Sc2 and Sc4 scenarios consider $CO_2$ emission reduction as one of their optimization objectives. For explaining these relatively high SEM values, we need to take a closer look at all solutions in the final Pareto frontier of these optimization runs.

Table 21- Electricity consumption results of PEPSO functionality evaluation tests on Richmond WDS

| Test Name | Off-Peak Energy Consumption (kWh) | On-Peak Energy Consumption (kWh) | Peak Power Demand (kW) | Total Unadjusted Energy Consumption (kWh) | Stored Water Volume Change (%) | Net Energy Consumption (kWh) | Net Energy Consumption Cost ($) | Net Electricity Cost ($) | Net $CO_2$ Emission (kg) (Based on marginal Emission factors) |
|---|---|---|---|---|---|---|---|---|---|
| Sc1-1 | 502.5 | 1032.2 | 135.7 | 1534.7 | -21% | 1852.2 | 112.1 | 112.1 | 1326.3 |
| Sc1-2 | 468.1 | 1024.8 | 136.2 | 1492.9 | -15% | 1709.6 | 104.5 | 104.5 | 1222.7 |
| Sc1-3 | 496.0 | 1026.1 | 135.8 | 1522.1 | -23% | 1890.5 | 114.5 | 114.5 | 1348.4 |
| Sc1-4 | 479.8 | 1031.5 | 135.5 | 1511.3 | -16% | 1749.9 | 106.8 | 106.8 | 1249.3 |
| Sc1-5 | 513.9 | 1018.4 | 136.0 | 1532.3 | -27% | 1983.7 | 119.1 | 119.1 | 1420.5 |
| Sc2-1 | 441.7 | 886.9 | 136.1 | 1328.6 | -20% | 1586.7 | 97.0 | 97.0 | 1124.8 |
| Sc2-2 | 440.2 | 813.3 | 136.4 | 1253.5 | -24% | 1559.3 | 93.2 | 93.2 | 1094.3 |
| Sc2-3 | 453.0 | 668.4 | 136.5 | 1121.4 | -33% | 1547.6 | 90.3 | 90.3 | 1097.6 |
| Sc2-4 | 524.0 | 1012.5 | 136.3 | 1536.4 | -19% | 1832.0 | 108.7 | 108.7 | 1310.8 |
| Sc2-5 | 468.3 | 670.7 | 136.6 | 1139.0 | -37% | 1646.8 | 95.8 | 95.8 | 1170.1 |
| Sc3-1 | 503.0 | 1050.4 | 136.2 | 1553.4 | -24% | 1935.3 | 118.5 | 118.5 | 1385.0 |
| Sc3-2 | 497.6 | 1045.0 | 135.9 | 1542.5 | -19% | 1832.7 | 111.9 | 111.9 | 1309.7 |
| Sc3-3 | 505.2 | 1026.4 | 135.5 | 1531.6 | -21% | 1856.5 | 112.1 | 112.1 | 1325.8 |
| Sc3-4 | 503.6 | 1026.4 | 135.7 | 1529.9 | -21% | 1845.8 | 111.2 | 111.2 | 1318.3 |
| Sc3-5 | 476.4 | 1057.9 | 135.8 | 1534.3 | -19% | 1821.3 | 112.9 | 112.9 | 1303.3 |
| Sc4-1 | 288.6 | 495.0 | 137.5 | 783.6 | -53% | 1398.6 | 90.3 | 90.3 | 972.3 |
| Sc4-2 | 414.0 | 572.9 | 137.7 | 986.9 | -53% | 1753.9 | 102.9 | 102.9 | 1238.8 |
| Sc4-3 | 397.7 | 637.8 | 136.9 | 1035.5 | -38% | 1522.1 | 92.1 | 92.1 | 1083.6 |
| Sc4-4 | 479.0 | 460.5 | 169.1 | 939.5 | -45% | 1498.9 | 81.3 | 81.3 | 1082.5 |
| Sc4-5 | 405.1 | 639.7 | 136.7 | 1044.9 | -37% | 1516.4 | 91.8 | 91.8 | 1061.3 |
| Sc5-1 | 523.4 | 984.4 | 135.9 | 1507.7 | -23% | 1871.7 | 109.3 | 109.3 | 1340.0 |
| Sc5-2 | 509.8 | 1013.2 | 135.7 | 1522.9 | -22% | 1854.6 | 110.9 | 110.9 | 1329.0 |
| Sc5-3 | 490.4 | 1020.7 | 135.4 | 1511.1 | -22% | 1840.2 | 111.2 | 111.2 | 1320.8 |
| Sc5-4 | 555.1 | 1013.5 | 169.4 | 1568.5 | -26% | 1997.9 | 117.2 | 117.2 | 1430.3 |
| Sc5-5 | 479.1 | 1038.5 | 136.0 | 1517.6 | -19% | 1809.1 | 111.0 | 111.0 | 1290.8 |
| Sc6-1 | 540.3 | 972.5 | 136.1 | 1512.8 | -28% | 1968.6 | 113.5 | 113.5 | 1414.3 |
| Sc6-2 | 492.0 | 976.6 | 135.0 | 1468.6 | -19% | 1740.6 | 102.5 | 102.5 | 1244.2 |
| Sc6-3 | 490.6 | 977.6 | 135.9 | 1468.2 | -18% | 1727.7 | 101.9 | 101.9 | 1241.5 |
| Sc6-4 | 491.4 | 974.0 | 136.1 | 1465.4 | -20% | 1765.2 | 103.9 | 103.9 | 1261.7 |
| Sc6-5 | 515.4 | 980.4 | 136.0 | 1495.7 | -23% | 1848.8 | 108.2 | 108.2 | 1325.0 |
| Sc7-1 | 455.5 | 1057.3 | 135.8 | 1512.8 | -15% | 1727.8 | 136.9 | 136.9 | 1233.8 |
| Sc7-2 | 467.9 | 1049.6 | 168.2 | 1517.5 | -7% | 1610.8 | 126.3 | 126.3 | 1148.8 |
| Sc7-3 | 481.2 | 1070.3 | 169.0 | 1551.5 | -17% | 1812.0 | 144.0 | 144.0 | 1298.2 |
| Sc7-4 | 396.6 | 1050.3 | 133.9 | 1446.9 | -11% | 1590.0 | 123.7 | 123.7 | 1132.4 |
| Sc7-5 | 422.9 | 1017.2 | 135.6 | 1440.0 | -10% | 1566.9 | 121.6 | 121.6 | 1115.4 |

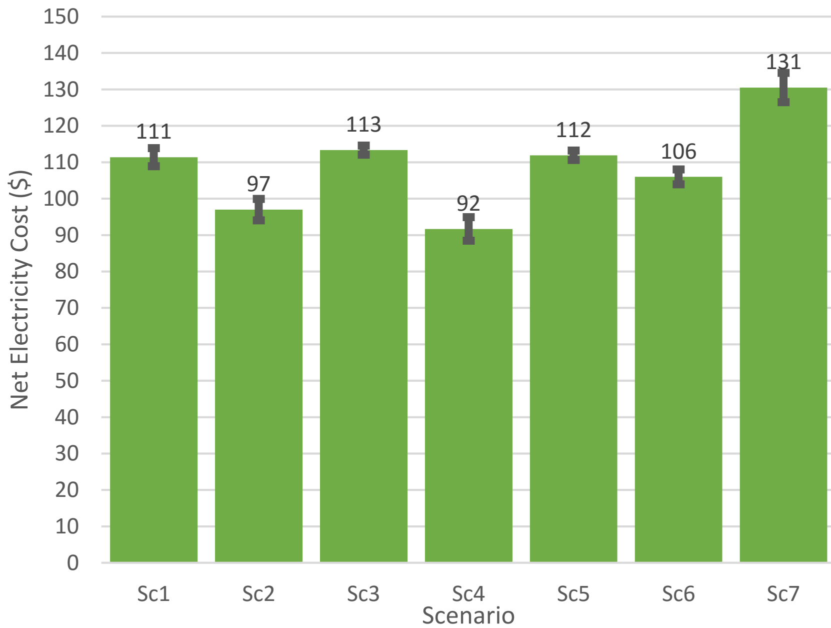



Figure 32- Net electricity cost of PEPSO functionality evaluation tests of Richmond WDS

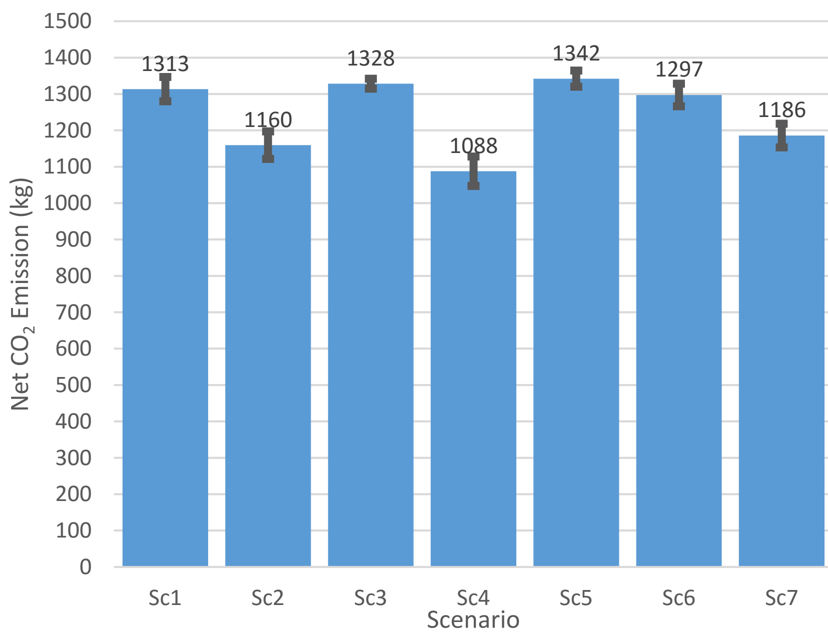



Figure 33- Net $CO_2$ emission of PEPSO functionality evaluation tests of Richmond WDS (calculated based on marginal emission factors)

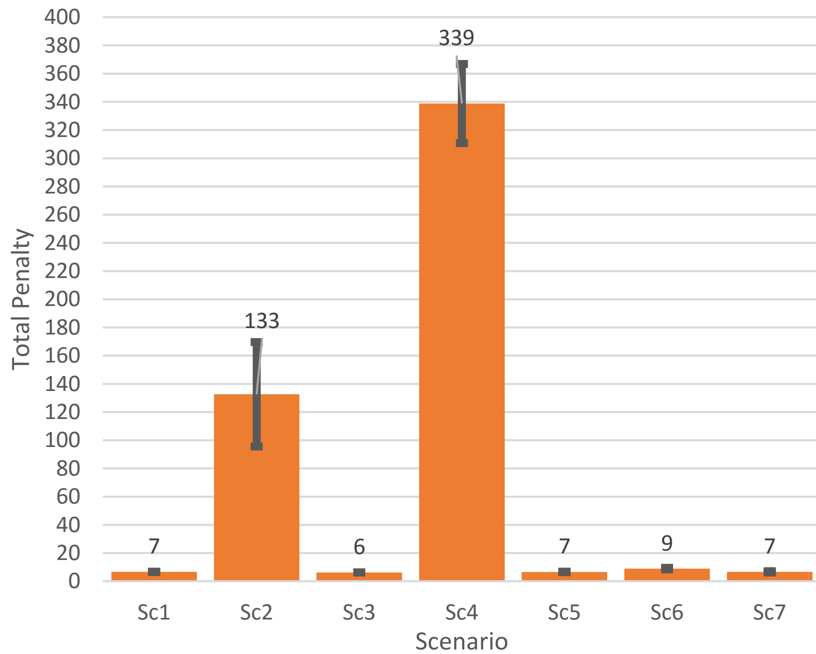



Figure 34- Total penalty of PEPSO functionality evaluation tests of Richmond WDS

In general solutions in the final Pareto frontier of these optimization runs can be categorized into two groups. The first group of solutions has higher energy consumptions and accordingly higher $CO_2$ emission. Consuming higher amount of energy increases the water pressure in the WDS and make water pressure at all strategic junction closer to the upper bound of the acceptable pressure range. It also increases the water level in tanks and push them up closer to the high allowed level of water in tanks. Solutions in the second group have lower energy consumption and accordingly their tanks will be drained during the operation period. In this solution, at the end of the operation period, the water level of some tanks drops below the desired level. This also causes low water pressure at some junctions (especially junctions 42 and 1302). In most cases, the second group of the solution has large penalties that are caused by low water level and pressure violations. In this cases, PEPSO selects the best solution from the other group of the solutions with high energy consumption but low penalty value and report it as the optimum solution. But in Sc2 and Sc4 scenarios, considering the $CO_2$ emission as one objective, and including it in the process of selecting the best solution, changes the final result. In this scenarios low $CO_2$ emission of the second group of solutions encourages PEPSO to select and report solution from this group as the optimum solution. This will happen more frequently in case of Sc4 scenario which electricity cost is not an objective and happen less frequently when all three objectives are evaluated (Sc2). This is explaining the high variation in the result of Sc2 and Sc4 scenarios that are showed as high SEMs. The average decrease in volume of stored water in Sc2 and Sc4 scenarios are 27% and 45% respectively. While the average decrease in volume of store water in the tank of other scenarios are 12% to 22%.

The above explanation can clarify the main reason of seeing relatively low energy consumption cost, $CO_2$ emission, and considerably high total penalty values in Sc2 and Sc4 scenarios with respect to the results of other scenarios. The net electricity cost of Sc2 and Sc4 are 30.7±5.5% and 17.7±4.7% lower than Sc1 respectively. Similarly, net $CO_2$ emission of Sc2 and Sc4 are 29.8±5.7% and 17.2±5.2% less than Sc1 respectively. As the results of Monroe tests, here it can be observed that optimizing based on both electricity cost and $CO_2$ emission has an amplifying effect on reduction of both objectives. The total penalty of Sc2 scenario is 133±37 while the total penalty of Sc4 scenario is 339±28. Higher penalty value and lower variation of results of Sc4 scenario can be explained by the fact that solution with low energy consumption and high penalty value are more frequent in the Sc4 scenario. Therefore, in an average solution of Sc4 are more uniform and with higher penalty values.

In the fifth scenario (Sc3) Richmond WDS optimized just based on the total penalty. In this case, net electricity cost and $CO_2$ emission are in average 1.7% and 1.1% higher than the base scenario (Sc1) respectively. This increase in energy consumption in Sc3 scenario let PEPSO find a solution with 5.7% lower penalty in respect to Sc1.

The hydraulic model of the skeletonized version of Richmond WDS was simpler that Monroe WDS. So, in this case, optimizing with or without UI calculation did not considerably changed the results. Results of both Sc1 and Sc5 scenarios are close in respect to total penalty, electricity cost and number of warnings. It seems that UI calculation helped a little bit to find solutions with slightly lower (2.2%±1.6%) $CO_2$ emission. But it should be considered that calculating UI is additional computation load on the optimization process.

Optimizing pump operation of Richmond WDS without water level constraints for tanks (Sc6), in average reduces the net electricity cost and $CO_2$ emission by 4.8% and 1.2% respectively. However, this increases the total penalty by 35.1%. The water pressure penalty at a junction in both Sc6 and Sc1 scenarios was zero, so the above-mentioned increase in total penalty was only related to increasing in water level penalty of tanks.

It was discussed previously that the electricity tariffs of Richmond WDS do not have power demand charge, so the whole electricity cost in this system is related to the time-dependent energy consumption charge. However in the Sc7 scenario, constant off-peak energy consumption charge was applied to all hours of the day. This flat rate electricity tariff in average reduced 3% of total energy consumption (kWh) from off-peak hours and added half of that to the on-peak hours. By this change, the remaining 1.5% of energy is saved. Previously, due to using a time of use electricity tariff, PEPSO needed to shift energy usage to reduce electricity cost of the system. This shift of energy usage caused some head losses during filling and draining tanks. By using the flat rate electricity tariff, energy was consumed at the time that it was needed which reduced 1.5% of the total energy consumption due to eliminating unnecessary head loss. It is interesting to see that flat rate tariff gives PEPSO more flexibility to find a solution that satisfies required volume of stored water at the end of the operation. The solutions of the base scenario (Sc1) drained 21.6±2.1% of stored volume of water in tanks, but Sc7 solution just drained 12.0±1.7% of this volume.

### 3.2.3. Analyzing PEPSO and DS Comparison Test Results

In Section 3.2.1 it was discussed that to have a fair comparison between results of two optimization test, reported energy consumption should be adjusted to take into account the effect of changes in the volume of stored water during the operation period. Results of PEPSO and DS runs also adjusted by this method and presented in Table 22.

Results of PEPSO and DS comparison tests that are presented in both Table 19 and Table 22 show that solutions that PEPSO and DS provided for the same problem are different in various aspects. For instance, electricity consumption and peak power demand of PEPSO solution are lower than DS while PEPSO solutions tend to drain tanks, but DS solutions tend to fill tanks.

The patterns of water level in tanks of Figure 35 indicates that DS tends to increase water levels in tanks at the end of operation period to meet the initial water level and even go beyond that. This can increase accumulated power demand at pump station due to turning on multiple pumps at the same time which increases peak power demand of the system for the whole electricity billing period and can considerably increase electricity cost of the system. However, PEPSO tries to keep water levels balanced during the optimization period and prevent power demand accumulation. Looking at typical peak power demand pattern of DS and PEPSO solutions (Figure 36) helps to see clearly this effect.

Table 22- Adjusted electricity consumption results of PEPSO and DS comparison tests

| Test Name | Stored Volume Change | Net Energy Consumption (kWh) | Net Energy Consumption Cost ($) | Power Demand Cost ($) | Net Electricity Cost ($) |
|---|---|---|---|---|---|
| PEPSO 1 | 1% | 6818.4 | 286.4 | 199.8 | 486.14 |
| PEPSO 2 | -3% | 6643.0 | 281.1 | 166.6 | 447.69 |
| PEPSO 3 | -12% | 6406.9 | 276.2 | 184.0 | 460.16 |
| PEPSO 4 | -15% | 6060.9 | 262.8 | 191.1 | 453.91 |
| PEPSO 5 | -6% | 6272.4 | 267.6 | 196.0 | 463.60 |
| DS 1 | 7% | 7091.3 | 296.0 | 238.9 | 534.9 |
| DS 2 | 7% | 7091.3 | 296.0 | 238.9 | 534.9 |
| DS 3 | 7% | 7091.3 | 296.0 | 238.9 | 534.9 |
| DS 4 | 45% | 7976.6 | 286.2 | 286.1 | 572.2 |
| DS 5 | 22% | 7941.5 | 323.6 | 349.3 | 672.9 |

Average, SEM and relative SEM of results of PEPSO and DS comparison test are calculated and shown in Table 23. It can be seen that SEM of PEPSO results is less than DS in almost all cases, which refers to more consistency in PEPSO results. The net energy consumption, peak power demand and electricity cost of all PEPSO runs are lower than DS results. One good point about the DS optimum pump schedules is that the volume of stored water in tanks at the end of the operation period of solutions of DS is always higher than the initially stored volume of water. For the PEPSO test in some cases the volume of stored water increased slightly but in most cases we can see 12% to 38% decrease in volume of stored water during a day.

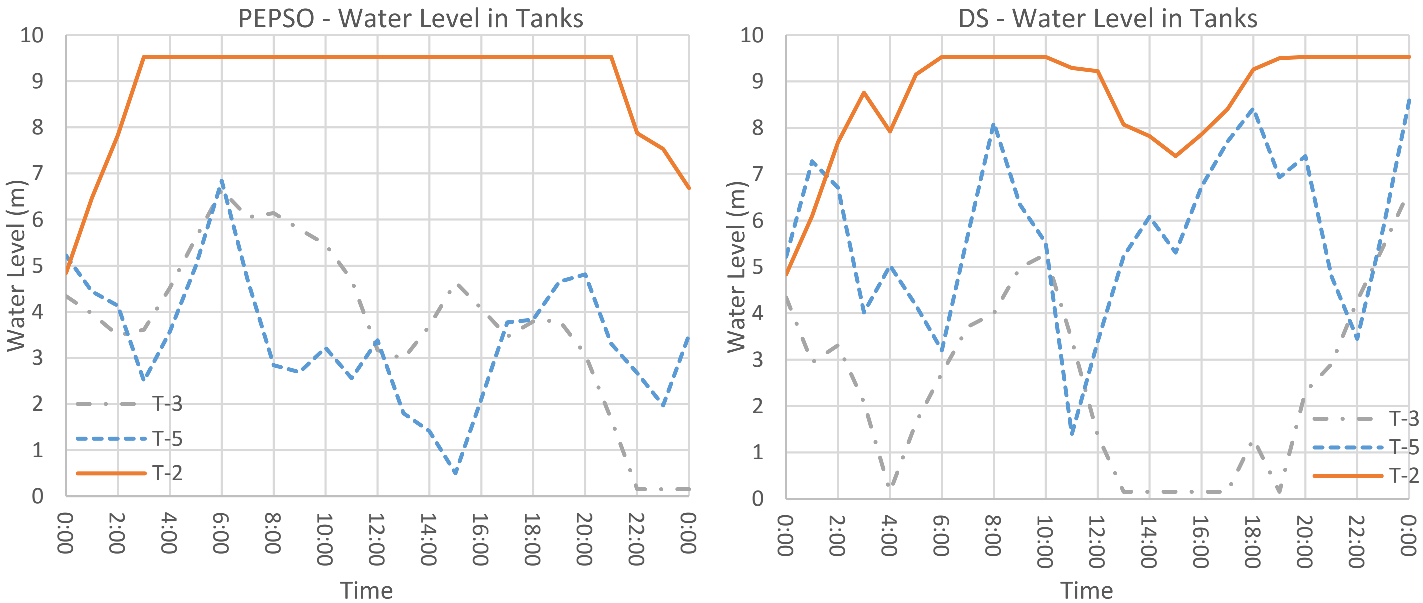



Figure 35- Pattern of water level in tanks of typical results of PEPSO (left) and DS (right) optimization runs of Monroe WDS

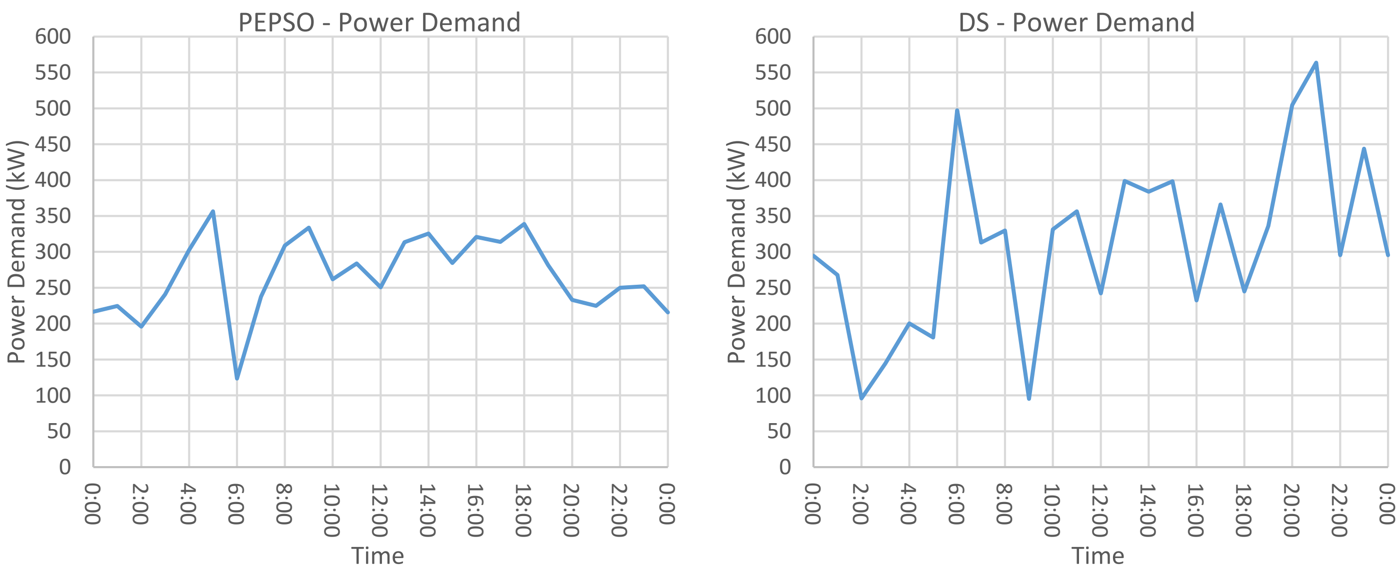



Figure 36- Pattern of power demand of typical results of PEPSO (left) and DS (right) optimization runs of Monroe WDS

Taking into account the SEM values and change in volume of stored water, the net energy consumption of the system based on DS solution is 9.0±3.8% higher than solutions of PEPSO. The peak power demand of optimum pump schedules of DS is 44.2±15.1% higher than optimum pump schedules of PEPSO. Accordingly, the net daily electricity cost of the Monroe WDS based on DS results is 23.3±7.1% higher than results of PEPSO.

Table 23- Statistical analysis of results of PEPSO and DS comparison test

| Parameter | | Net Energy Use (kWh) | Net Energy Use Cost | On-peak Energy Use (kWh) | Off-peak Energy Use (kWh) | Peak Power Demand (kW) | Power Demand Cost | Stored Volume Change | Pressure Penalty at Junction | Net Electricity Cost |
|---|---|---|---|---|---|---|---|---|---|---|
| PEPSO | Mean | 6640.3 | $274.8 | 2387.6 | 4092.8 | 392.8 | $187.5 | -0.1 | 0.02 | $462.3 |
| | SEM | 125.6 | $4.1 | 65.5 | 92.6 | 11.5 | $5.5 | 0.03 | 0.01 | $6.2 |
| | Relative SEM | 2.0% | 1.5% | 2.7% | 2.3% | 2.9% | 2.9% | -39.1% | 60.9% | 1.3% |
| DS | Mean | 7438.4 | $299.5 | 2596.6 | 4733.3 | 566.5 | $270.4 | 0.2 | 204.5 | $569.9 |
| | SEM | 199.9 | $5.9 | 138.6 | 46.2 | 42.8 | $20.4 | 0.1 | 5.9 | $25.1 |
| | Relative SEM | 2.7% | 2.0% | 5.3% | 1.0% | 7.6% | 7.6% | 40.1% | 2.9% | 4.4% |

The left pair of columns in Figure 37 displays daily electricity cost of Monroe WDS after optimizing by PEPSO and DS. The small error bar on top of each column shows its SEM. Electricity cost of the system combined from the power demand cost and energy consumption cost. So these components are shown by the two pairs of columns on the right side of Figure 37.it is clear that most of the difference between net electricity cost of PEPSO and DS solution are related to the power demand cost. So important effect of peak demand shaving in the operation of WDS and reducing electricity cost of systems in obvious here.

The left pair of columns of Figure 38 displays the net energy consumption of both solutions of PEPOS and DS. As Figure 37, error bars on this chart show SEM values. Both on-peak and off-peak component of energy consumption are shown separately by two pairs of columns on the right side of Figure 38. This bar chart indicates that on-peak energy consumption of the system in PEPSO solution is slightly less than DS solution.

However off-peak energy consumption of the system based on optimum pump schedule of PEPSO is 15.7±3.7% lower than DS.

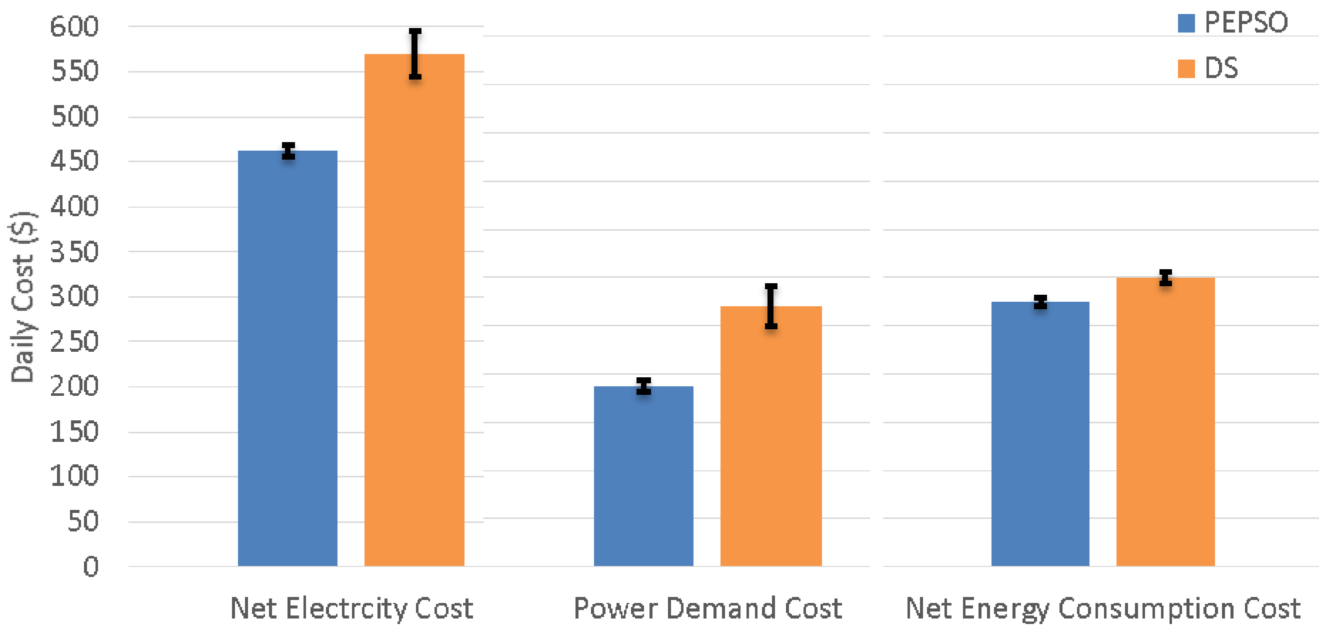



Figure 37- Electricity, power demand and energy costs of solutions of the PEPSO and DS comparison tests on Monroe WDS

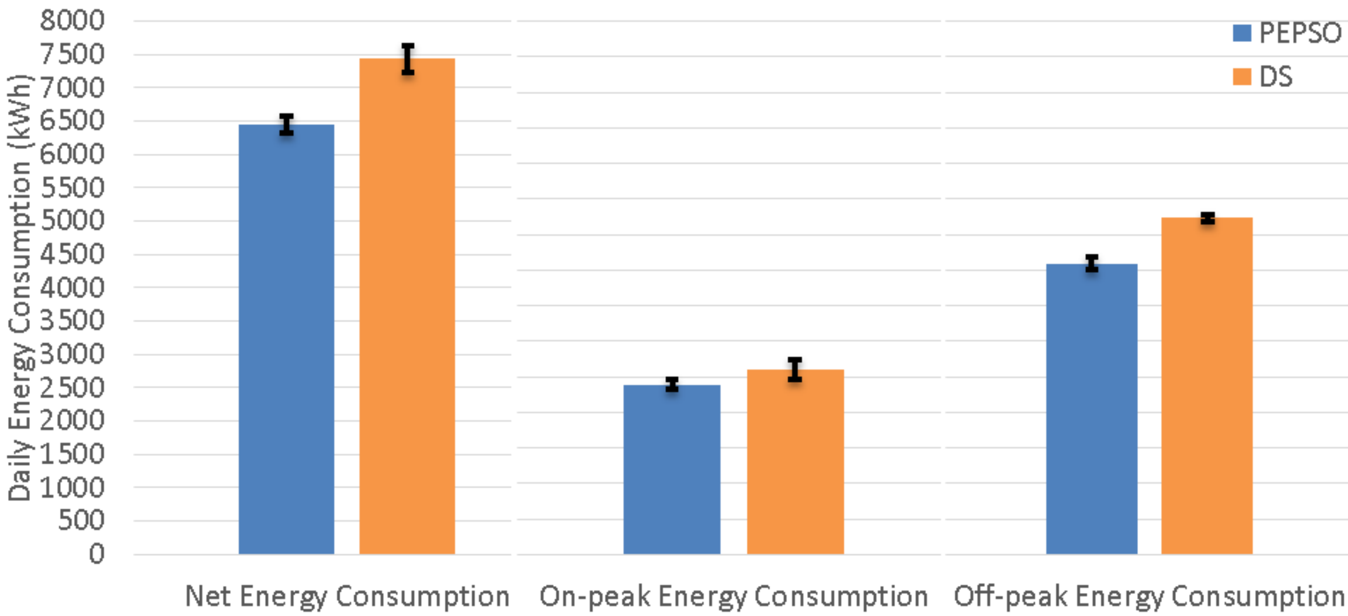



Figure 38- Net, on-peak and off-peak energy consumption of solution of the PEPSO and DS comparison test on Monroe WDS

Figure 39 displays a typical optimized pump schedule of Monroe WDS with PEPSO (top) and DS (bottom). Numbers on a cell of the pump schedules show the relative rotational speed of two variable speed pumps in percentage. We know that some elevated demand points of the Monroe WDS need the help of the booster pump station to receive water. So at each time, at least one of the two variable speed pumps in the booster pump station should work. Figure 39 shows that PEPSO addressed this need better that DS. At all times, there is, at least, one of two variable speed pump is ON and just in 5 hours both pumps are working.

Optimized pump schedule of PEPSO

| Pump \ Hour | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E-2 | | | | | | | | | | | | | | | | | | | | | | | | | |
| E-3 | | | | | | | | | | | | | | | | | | | | | | | | | |
| E-4 | | | | | | | | | | | | | | | | | | | | | | | | | |
| E-5 | | | | | | | | | | | | | | | | | | | | | | | | | |
| E-6 | | | | | | | | | | | | | | | | | | | | | | | | | |
| E-7 | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-8 | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-9 | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-10 | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-11 | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-12 | | | | | | | | | | | | | | | | | | | | | | | | | |
| PMP-9 | 97 | 81 | 80 | 89 | | 94 | 74 | | 92 | | 82 | | 96 | 86 | 92 | | | | 93 | | | | 82 | | 97 |
| PMP-544 | | 69 | 84 | | 96 | | | 99 | | 95 | | 71 | 77 | 87 | 92 | 86 | 80 | 99 | | 80 | 83 | 65 | | 64 | |

Optimized pump schedule of DS

| Pump \ Hour | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E-2 | | | | | | | | | | | | | | | | | | | | | | | | | |
| E-3 | | | | | | | | | | | | | | | | | | | | | | | | | |
| E-4 | | | | | | | | | | | | | | | | | | | | | | | | | |
| E-5 | | | | | | | | | | | | | | | | | | | | | | | | | |
| E-6 | | | | | | | | | | | | | | | | | | | | | | | | | |
| E-7 | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-8 | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-9 | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-10 | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-11 | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-12 | | | | | | | | | | | | | | | | | | | | | | | | | |
| PMP-9 | | 99 | 77 | | 77 | 75 | 91 | | 93 | 82 | | | 67 | 83 | | 96 | 74 | 70 | 67 | 78 | 92 | 97 | 87 | 76 | |
| PMP-544 | 65 | 75 | | | 70 | 74 | 70 | 96 | 93 | 85 | 74 | 72 | | | 76 | | 67 | 91 | 73 | 80 | 82 | 99 | 98 | 93 | 65 |

Figure 39- Typical optimized pump schedule of Monroe WDS, PEPSO (top) and DS (bottom)

Initially number of pump switches in both PEPSO and DS was limited to 24. This can be considered as practically no constraint on pump switches. As in 24 hours period with one-hour time step, it is possible to have 12 pump starts. Despite the fact that number pump switches were not constrained, in an average number of pump switches for PEPSO and DS are about 5 and 4 times per day respectively.

It shows that even without having a constraint on pump switches it is rare to have an optimized pump schedule with a considerably high number of pump starts. Pumps PMP-544 and PMP-9 are in the same pump station and has the same characteristic curves. So when one of them is on we can simply turn the other one and turn of the first pump and see the same flow and head out of pump station. Pumps E-3, E-4, and E-5 and pumps W-10, W-11 and W-12 also can be grouped in the same way. Accordingly, we can edit optimized pump schedules of Figure 39 and make them simpler.

Optimized pump schedule of PEPSO

| Hour / Pump | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E-2 | | ■ | | ■ | ■ | ■ | ■ | ■ | | | ■ | ■ | ■ | ■ | ■ | ■ | | ■ | | ■ | | | | | |
| E-3 | | ■ | ■ | | ■ | ■ | | ■ | ■ | ■ | | ■ | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | |
| E-4 | | | | | ■ | ■ | | | | ■ | | ■ | | | | | | ■ | | ■ | | | ■ | | |
| E-5 | | | | | | | | | | | | ■ | | | | | | | | | | | | | |
| E-6 | ■ | | | ■ | | ■ | | | | | | | ■ | | | | | | | | | | | ■ | ■ |
| E-7 | | ■ | | | ■ | | | | | ■ | ■ | | | | ■ | ■ | | ■ | | | | ■ | | | |
| W-8 | | | | | | | | | | | | | | ■ | | | ■ | | ■ | | | | | | |
| W-9 | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-10 | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-11 | | | | | | | | | ■ | | ■ | | | | | | | | | | ■ | | | | |
| W-12 | | | ■ | | | | ■ | ■ | ■ | ■ | | | | | ■ | ■ | | | | ■ | | ■ | ■ | | |
| PMP-9 | 97 | 81 | 80 | 89 | 96 | 94 | 74 | 99 | 92 | 95 | 82 | 71 | 96 | 86 | 92 | 86 | 80 | 99 | 93 | 80 | 83 | 65 | 82 | 64 | 97 |
| PMP-544 | | 69 | 84 | | | | | | | | | | 77 | 87 | 92 | | | | | | | | | | |

Optimized pump schedule of DS

| Hour / Pump | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E-2 | | | ■ | ■ | | | | | ■ | | | | | | | | | ■ | | ■ | | | ■ | ■ | |
| E-3 | | ■ | | | ■ | | | | ■ | | | ■ | | | | | | | | ■ | ■ | | | | |
| E-4 | | | | ■ | | ■ | | | | ■ | ■ | ■ | | ■ | | ■ | ■ | ■ | ■ | | ■ | ■ | ■ | | |
| E-5 | | | | | | | | | | ■ | | ■ | | | | | ■ | | | | | | | | |
| E-6 | | | | | | | | | | | | | | | | | | | | | | | | | |
| E-7 | | ■ | | | | | | | | | | | | | | | | | | ■ | | ■ | ■ | ■ | |
| W-8 | ■ | | | | | | ■ | ■ | | | | | ■ | ■ | ■ | ■ | ■ | ■ | | | | | ■ | ■ | ■ |
| W-9 | | | | | | | | ■ | | | | | | | | | | | | | | | | | |
| W-10 | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-11 | | | | | ■ | | ■ | | ■ | ■ | | ■ | ■ | | | | | | ■ | | | | | ■ | |
| W-12 | ■ | | | | | ■ | | | | | | | | | | ■ | | | | ■ | ■ | | | | ■ |
| PMP-9 | 65 | 99 | 77 | | 77 | 75 | 91 | 96 | 93 | 82 | 74 | 72 | 67 | 83 | 76 | 96 | 74 | 70 | 67 | 78 | 92 | 97 | 87 | 76 | 65 |
| PMP-544 | | 75 | | | 70 | 74 | 70 | | 93 | 85 | | | | | | | 67 | 91 | 73 | 80 | 82 | 99 | 98 | 93 | |

Figure 40- Polished optimized pump schedule of Monroe WDS, PEPSO (top) and DS (bottom)

In each group, we do not turn on the second or third pump unless the first pump is on. Polished pump schedules of Figure 39 are displayed in Figure 40. Here we can see that the average number of pump switches of results of PEPSO and DS dropped to less than 4.

A *solution evaluation* is hydraulically simulating a WDS based on a proposed pump schedule and evaluating its results. Required CPU time for one *solution evaluation* of the Monroe WDS by a Lenovo ThinkPad W520 workstation (see section 2.3.1 for its specifications) is recorded for both PEPSO and DS optimizations. In average, CPU time of each solution evaluation by PEPSO and DS are 0.052 and 0.100 seconds (their real time are 0.416 and 0.796 seconds) respectively. This result shows that speed of PEPSO in solution evaluation in average is about two times more than DS. Also, the number of iteration that each of them needs to find an acceptable near optimum solution can change total required the time of optimization.

Optimization objectives trend graph of PEPSO that is displayed in Figure 41 (top) is a useful component of this tool. It lets the user see the optimization trend and decide what the optimum number of iteration is for reaching to an acceptable result in a limited amount of time. The horizontal axis shows a number of iteration and *solution evaluation*; the left vertical axis shows net electricity cost ($) and the right vertical axis shows total penalties. It can be seen that PEPSO rapidly reduces values of both objectives (especially the total penalty). Almost after 200 iterations, it can report an optimized solution comparable with DS final solution after 1000 iterations. It shows that it is possible to run PEPSO with a considerably lower number of iterations and in a shorter period of time get a near optimum solution that is practically acceptable. This optimum number of iterations changes based on the complexity of the problem and different options for optimization. DS do not report the result of intermediate iteration in a tabular or graphical format. Therefore, it is hard for DS users to decide about the optimum number of iteration for optimizing their WDSs.

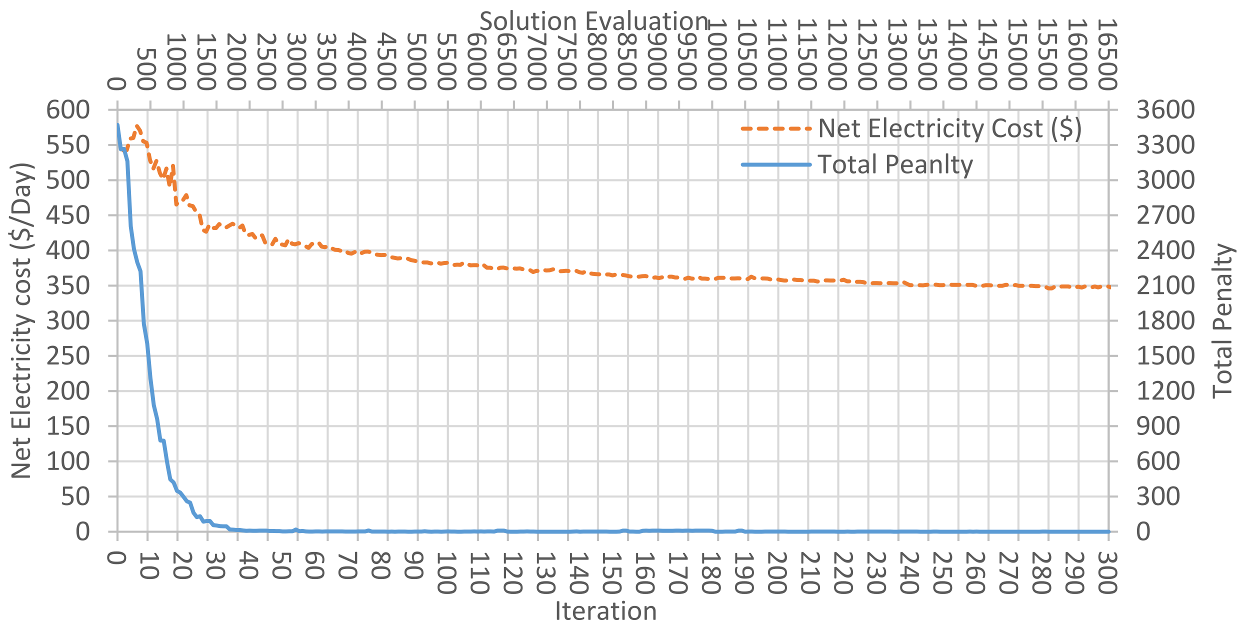



Figure 41- A typical optimization trend of optimization run of Monroe WDS with PEPSO

# CHAPTER 4 CONCLUSION AND FUTURE STUDIES

Based on the test result that explained and analyzed in chapter 3 and based on experience that gained while developing PEPSO findings of this study are concluded in the following section, and some suggestion for further researches in this area are provided.

## 4.1. Conclusion

One of the main goals of this study was the development of software that can effectively optimize pump operation of WDSs to reduce the associated electricity cost and pollution emission. This tool should be able to provide a user-friendly environment and give the user an ability to optimize medium and large size WDS under different scenarios. The second version of PEPSO, which is introduced in section 2.2 has a graphical user interface which is designed by considering different factors including clarity, concision, familiarity, responsiveness, efficiency, consistency, aesthetic and forgiveness (for more details see Section 2.2.1). All the seven forms and multiple tabs of PEPSO designed based on a set of logic that let the user input data efficiently and accurately. Default options of PEPSO enable less technical users to run a simple optimization simulation without dealing with adjusting numerous options. However, users can edit any part of input and define various options to create a customized optimization run.

Via, the electricity form users, can define detailed electricity tariff for each pump including time of use energy consumption charge ($/kWh) and power demand charge ($/kW). Pollution emission form lets the user select desired pollution or combination of pollutions for optimization. PEPSO can connect to the LEEM server or use offline reports to get emission factors (kg/kWh) that are required for pollution emission optimization.

Different tabs of the constraint form let users define different types of hard and soft constraints on pumps, tanks, junction and pipe of WDS. Various optimization option including different stopping criteria, exploration and exploitation rates, initial conditions, etc. can be defined via the optimization option form. The user also can select any combination of objectives for optimization. Finally, the Report form provides a wide range of reports that user can see and save in the format of text (tabular data) and/or 2D and 3D graphics (charts and plots). All of these features can be accessed through the graphical user interface or can be defined and edited directly on the PEPSO project file by using a simple text editor.

The modular structure of PEPSO and unique functions and procedure that is defined for this tool are even more interesting than the user-friendly interface. The modular structure of PEPSO, which organize its 18 thousands lines of code, can make it more useful and powerful tool for researchers, WDS designers, and operators. Modular structure also makes it easier to add and remove components of PEPSO and make it more editable and upgradable in future. PEPSO has 17 main modules including but not limited to, EPANET hydraulic solver, ANN training set creator and trainer, ANN hydraulic solver, objective calculator, UI calculator, NSGA II, best solution finder, text reporter and plotter. PEPSO uses a customized version of the non-dominated sort genetic algorithm II to find at first Pareto frontier of solutions and then select the best one as the optimum pump schedule of the WDS. PEPSO can use both EPANET toolkit for hydraulically stimulating the WDS or train and use an ANN instead of EPANET model. UI and inadmissibility calculator modules are a unique part of this tool that enables it to find promising ways of combining and changing solution to improve them and get closer to the global optimum solution ( see Sections 2.2.2.5 and 2.2.2.11). Using UI to guide mutation, crossover and elitism steps of the GA to generate feasible solutions and find the optimum solution faster is an approach that can be used in other optimization effort beyond the WDS operation optimization.

PEPSO tested with different scenarios and compared with DS, which is one of the most famous commercial tools in this field. These tests have been done on the detailed hydraulic model of Monroe WDS and skeletonized version of Richmond WDS model. Results of test with eight optimization scenarios on Monroe WDS showed that:

- PEPSO was able to optimize the detailed model of Monroe WDS effectively with 13 pumps in about 2 hours with a computer system that can be found in typical WDS design or operation center (see Section 2.3.1 for more details).
- Optimizing based on electricity cost and CO2 emission can reduce CO2 emission of the system by 1.3 to 3.4%.
- Optimizing based on these two objectives at the same time is more effective that optimizing based on only the CO2 emission. An optimized system based on all objectives generate 1.4±1.3% less CO2 emission in comparison with the same system that is solely optimized based on the pollution emission.
- In general, total penalties of an optimized solution of all scenarios of Monroe WDS were low. However, Optimizing based on just penalty (Sc3 scenario) reduced the total penalty by 10±7% with respect to the scenario of optimizing all objectives (Sc2).

- Calculating UI values helped PEPSO to find more practical optimized solutions with fewer EPANET warnings and less tank draining. However, UI calculation in average increased required time of optimization by 8.9%.
- Scenario Sc6 is optimized Monroe WDS without tank level constraints. Water level penalty of tanks of Sc6 scenario is more than 4 times of the water level penalties of the base scenario (Sc1). Although removing the tank level constraints reduced about one-fourth of the energy consumption cost and CO2 emission but it was concluded that it considerably increases water level and water pressure penalties and led to impractical and unacceptable solutions.
- From the Sc6 tests, it was also concluded that trying to balance tank level at the end of operation period is not solely enough for preventing draining of tanks and water level constraints on tanks helps to balance the final water level in tanks effectively.
- Time of use electricity tariff forces PEPSO to shift 1.7% of energy consumption from on-peak hours to off-peak hours to reduce energy consumption cost. Including power demand charge in electricity tariff also in average shaves 9.7% of the peak power demand of the system.

  The similar scenarios were tested on a skeletonized hydraulic model of Richmond WDS. Result of these test showed that:

- PEPSO was able to optimize effectively skeletonized model of Richmond WDS with seven pumps in about half an hours with a computer system that can be found in typical WDS design or operation center ( see Section 2.3.1 for more details)

- PEPSO is able to find a solution with very low penalty and in some cases zero pressure penalty for this system.
- In those scenarios that $CO_2$ emission was an objective of optimization (Sc2 and Sc4 scenarios) considering pollution emission reduction as an objective of the optimization process pushes the Pareto frontier toward solutions with lower energy consumption and higher penalty value. In this solution, we can see the considerable low water level and pressure violation. In these solutions, tanks will be drained during the operation period.
- Optimization just based on the total penalty slightly increases the energy usage of the system that causes 5.8% reduction in total penalty.
- Like Monroe WDS, optimizing without tank level constraints reduces the electricity cost and $CO_2$ emission. However, it considerably (35.1%) increases water level penalty of tanks
- Using a flat rate energy consumption charge instead of the time of use tariff enables PEPSO to consume energy at the time of high demand. This eliminated the need to storing more water during off-peak hours that was causing energy losses. By this method, PEPSO reduced the total energy consumption of the system by 1.5% and reduced tank draining by about 10%.

Finally, comparison test between PEPSO and DS on Monroe WDS showed that:

- Electricity consumption and peak power demand of PEPSO solution are lower than DS while PEPSO solutions tend to drain tanks, but DS solutions tend to fill tanks.

- PEPSO is considerably better than DS in peak power demand shaving. Peak power demand of PEPSO solutions is 44.2±15.1 lower than DS solutions.
- Even without constrained number of pump switches in a day, both PEPSO and DS in average started pumps about 4 to 5 times in a day.
- PEPSO is about two times faster than DS in completing a solution evaluation of Monroe WDS. In addition, PEPSO is able to reach to an acceptable near optimum solution with less number of solution evaluation.

## 4.2. Future Research

In Section 4.1 it was concluded that in a reasonable amount of time, PEPSO is able to optimize and provide logical results for a medium size WDS model with 13 pumps and thousands of system components under different scenarios. It also was mentioned that this tool in many aspects can provide better results in comparison to famous commercial optimization tools in the market. However during the PEPSO development and testing process, it was realized that there are other potential techniques that can be used to improve speed and accuracy of PEPSO. Some of these ideas are listed here:

- Adding batch run and sweeping option to PEPSO modules for finding the best set of optimization and ANN trainer parameter for a WDS. This can help to find the optimum options of PEPSO for each problem and adjust them automatically without involving the user.
- Considering the change of binary coding to trigger based coding to reduce size of solution space and making the optimization process faster and less computation intensive

- Store all network and optimization input and output data of PEPSO into a database that make storing processing and retrieving data more efficient.
- Adjusting optimization parameter like mutation and crossover rate on the fly and based on the different phase of optimization.
- Using multithreading structure to do the optimization calculation in parallel (e.g. EPANET hydraulic simulation, initial training of ANNs and their re-trainings, etc.)

In addition to the changes that can make PEPSO faster and more accurate, there are other capabilities that can be added to PEPSO to increase its usability. Here is a list of some of these capabilities that can make PEPSO a more powerful WDS operation optimization tool:

- Adding more tank level control like desired water level in the tank at the specific time of day
- Adding an option to connect pumps to specific tank, strategic junction or pipes to be able to adjust the tank level, pressure or velocity of network components effectively
- Add an option to do the above-mentioned task automatically and find the effect of status of pumps of the network on different component of the network (i.e. tanks, junctions, pipes)
- Adding an ability PEPSO for training and using time series ANNs. These type of ANNs can be more accurate for simulating tank level and pressure at junctions.
- Adding clustering tool that can find the area of the water network that the water pressure at their junctions are completely related and can be presented with a representative junction. This tool can help to select strategic junctions

automatically and define desired and allowed water pressure range based on requirements of the system.

- Considering effect of valves in optimization and add a component to PEPSO which optimizes valve operation parallel to pump operation
- Add a pump comparison tool to categorize pumps and find the similar pumps that can make the final solution polishing step more efficient. It helps to report more practical near optimum solution.
- Add an option to adjust the operation of each pump at a time block based on the operation of the same pump at some previous and next time blocks. This may help to have simpler optimized pump schedule with less number of utilized pumps and less number of pump starts
- Although the penalty calculation concept is used for water level, pressure and velocity constraints, the same method can be used for water quality constraints. In this case, PEPSO can find an optimum pump schedule that reduces water age (especially by draining and filling tanks). Therefore, the possibility of adding a quality constraint to PEPSO and evaluating its effectiveness can be investigated in future studies.

Although the second version of PEPSO which is developed in this study is more user-friendly than the initial version but there are other options that can be added to PEPSO to make it even more appealing for users. Adding these options to PEPSO or other similar tools can make them a good choice for both research purposes and operation optimization of real WDS. The below list shows some of our suggestion to improve the interface of PEPSO:

- Providing more graphing and reporting options like reporting result in the format of Excel files. Adding more interactive graphs, optimization graph evolvement video, etc.
- Adding more flexible and sophisticated tools for selecting the final optimum solution among the solutions of the final Pareto frontier. This can be used as an automatic alternative to expert judgment for selecting the best solution from the final Pareto frontier
- Increase forgiveness of software by double checking user inputs, suggesting a possible correction or changing them automatically to prevent fatal errors.
- Adding more accessible and on-demand help and examples for the user while adjusting optimization parameters and inputting data. Also providing more detailed explanation of outputs and possible ways for interpreting them.

Finally, our search for finding a suitable benchmark test case showed that there is not a perfect benchmark model which can be used for testing tools like PEPSO. As it was explained in Section 0, a lot of models that are used by researchers are very simple or even does not have a pump. Another network that has pumps are mostly used for design optimization problems and does not have required characteristics for testing an operation optimization tools. For instance, most of these WDS models do not have variable speed pumps, time of use electricity rate or enough elevated storage capacity for shifting energy consumption. Accordingly a considerable number of researches that have been done on operation optimization of pumps used a simplified WDS model or a WDS that is not available for public use and cannot be used for comparing results of different tools and methods. Therefore, developing a benchmark test case for comparing WDS operation optimization tools seems necessary for future researches in this field. Here are some basic suggestions for a benchmark water distribution system that can be used as a benchmark model of this type of research:

- The test case should have more than 10 constant and variable speed pumps
- There should be multiple pump station and multiple sources of water with different characteristics
- The model should have some elevated tanks, and their storage capacity should be realistic and comparable to daily demand. Demand point should have more than one demand pattern. Network topography should have both flat and steep area.
- The network should have both loop and branch structure. Also having some isolated flow of pressure zones provide more flexibility for testing.
- The network should have some values with defined characteristic curves.
- Some similar pumps should be located in the same pump station.
- Pumps should have different operation range. All pumps should have realistic head-flow and efficiency-flow curves.
- The network should have booster pump station. Some pump station should have parallel and series structure.
- There should be at least pump station with more than one electricity meter
- Electricity tariff should have 24 hours, 7 days, monthly and annual pattern. There should be both time of use and flat rate electricity tariffs. All of them should have peak power demand charge. The difference between on-peak and off-peak electricity rates should be realistic but considerable.

- Latitude and longitude of pumps should be defined, and a default emission factor report should be prepared. The emission factor report should cover emission of a different source of energy during hours of a day.

Definitely, testing PEPSO with a suitable test case and using test cases with more diverse topology and hydraulic conditions will help to have more accurate and clearer picture of potentials of this tool. It also will help to find better paths of improvements.

# APPENDIX A

## Glossary

***Average Energy Consumption Charge:*** The weighted average of on-peak and off-peak energy consumption charge based on the length of on-peak and off-peak periods of an electricity tariff.

***Constraint Importance Multiplier:*** Is a user-defined factor which will be multiplied by calculated penalty value that is corresponding to a component of the WDS to increase or decrease its effect on the total penalty value of a solution. For instance is water pressures at two strategic junctions of a WDS show the same amount of violation but constraint importance multiplier of the first junction is two times more than the second junction, penalty value that is associated with pressure violation of the first junction is twice more that penalty value of the second junction

***CPU time:*** is the amount of time that CPU spent on a processing instructions of a section of code of PEPSO and calculated by multiplying real time of completing the process by average CPU usage percentage at that period.

***Emission Factor (Emission Rate):*** a number with the pollution weight over energy consumption dimension (e.g. lb/kWh) that if multiplied by energy consumption results in pollution emission associated with energy consumption

***Energy Consumption Charge:*** Is cost of consuming one unit of energy (e.g. $/kWh). Multiplying energy consumption charge by the amount of consumed energy by a pump results in total energy consumption cost of the pump.

***Energy Intensity (EI):*** The average amount of energy needed to transport water from source to demand points per unit of water volume (kWh/m$^3$)

***EPANET input file:*** is a *.inp file that has all information of the hydraulic model of a WDS. PEPSO needs this file to optimize a WDS. For more information, please refer to EPANET user manual (Rossman 2000).

***Exploitation:*** Fine tuning good solution to improve their quality and get closer to the optimum point or visiting surrounding area of the current solution to find a slightly better solution that is located around them. The crossover operator of GA is mostly used for exploitation process (ˇCrepinˇSEK, Liu et al. 2011).

***Exploration:*** Searching for new solutions by visiting new areas of the solution space that have not been discovered. It helps algorithm to prevent getting stuck in a local optimum and increase the chance of finding the global optimum in non-convex problems. The mutation operator of GA can be used to help exploration process (ˇCrepinˇSEK, Liu et al. 2011).

***External Cost:*** An external cost arises when the social or economic activities of one group of persons have an impact on another group and when that impact is not fully accounted, or compensated for, by the first group. For instance, the external cost of electricity can arise from environmental footprint of generating energy (e.g. air pollution of burning coal in coal power plants) which is not included in electricity price

***LEEM report file:*** Is a comma separated value (*.CSV) file which has emission factors of the current, past and future times of the requested location. Durations of data in the past and future that are reported depend on the location and time of the query. For instance LEEM 2.5 is able to report between 6 to 37 hours of emission factor prediction based on latitude and longitude of the query.

***Net Energy Consumption:*** The total energy consumption of the system considering the effect of accumulating or draining energy based on a change in volume of stored water in elevated tanks of the system.

***Optimized EPANET file:*** is the final output of optimization process of PEPSO in the form of a *.inp file which is similar the initial EPANET input file but its pump control section is filled based on the pump schedule of the optimum solution of PEPSO

***Optimum Solution:*** usually a local optimum and occasionally a global optimum solution of an optimization problem. In this specific case, the optimum solution is an optimum pump schedule that satisfies the hard and soft constraint of the problem (e.g. tank level controls, pressure limits, etc.) and minimizes the other objectives (e.g. electricity cost, pollution emission,etc.).

***Pareto Frontier:*** Pareto frontier is a set of Pareto optimal solutions that are better than other solutions with respect to all objectives but cannot dominate each other in respect to all different objectives. All solutions that are members of a Pareto frontier are better that other solutions in respect to at least one objective value.

***Peak Power Demand:*** peak power demand of an electricity meter can be calculated as a maximum power demand of the electricity meter during a defined billing period (e.g. one month) that is measured in a defined time intervals (e.g. 30 minutes intervals). For calculating peak power demand of an electricity meter at a time block, required power of all pumps that are connected to the electricity meter at that time block will be added up.

***Penalty:*** a numeric value that is calculated based on the amount of violation from a defined constraint. For instance, if maximum allowed pressure of a junction is 25 meter of water head, a junction pressure equal to 30 meter shows 5 meter violation and when the violation raised to the power of 1.5 (or any other defined arbitrary number as a penalty power) final amount of pressure violation penalty is $5^{1.5}$=11.18

***Population:*** collection of a group of solutions

***Power Demand Charge:*** Is cost of demanding one unit of power (e.g. $/kW). Multiplying power demand charge by the peak power demand of a pump results in total peak power demand cost of the pump.

***Project file:*** Is a file that is created by PEPSO based on project definition which is provided by the user via the user interface. This file can be manually edited by text editors. The project file has required information for running an optimization simulation by PEPSO and includes, electricity tariffs, electricity meter data, pollution emission scenarios, optimization options, reporting options, initial population, WDS component constraints, etc.

***Proportional Importance (PI):*** A value that shows the importance of an element with respect to other elements of an array. An element with higher PI has a higher chance to be selected by the roulette wheel sampling method.

***Relative Rotational Speed:*** Rotational speed of a variable speed pump with respect to its maximum rotational speed. It can be a number between 0 to 100% which 100% is maximum rotational speed of the variable speed pump

***Solution:*** a pump operation schedule that define on or off status of fix speed pumps and rotational speed of variable speed pumps

***Solution Space:*** The *solution space* of pump operation optimization problem is a collection of all possible combination of the operational status of pumps of a system. For instance solution space of a pair of constant speed pump and variable speed pump that the variable speed pump can work at 0%, 75% and 100% of its nominal rotational speed is: [(off,0%),(off,75%),(off,100%),(on,0%),(on,75%),(on,100%)]

***Strategic Junction / Strategic Pipe:*** strategic junction or pipe is an important component of a WDS which can act as an indicator of the status of surrounding component or the whole WDS. It means that for instance, by adjusting the pressure of a strategic junction within the desired range, we can make sure that pressures of other surrounding junction or even all junction in WDS are within acceptable range.

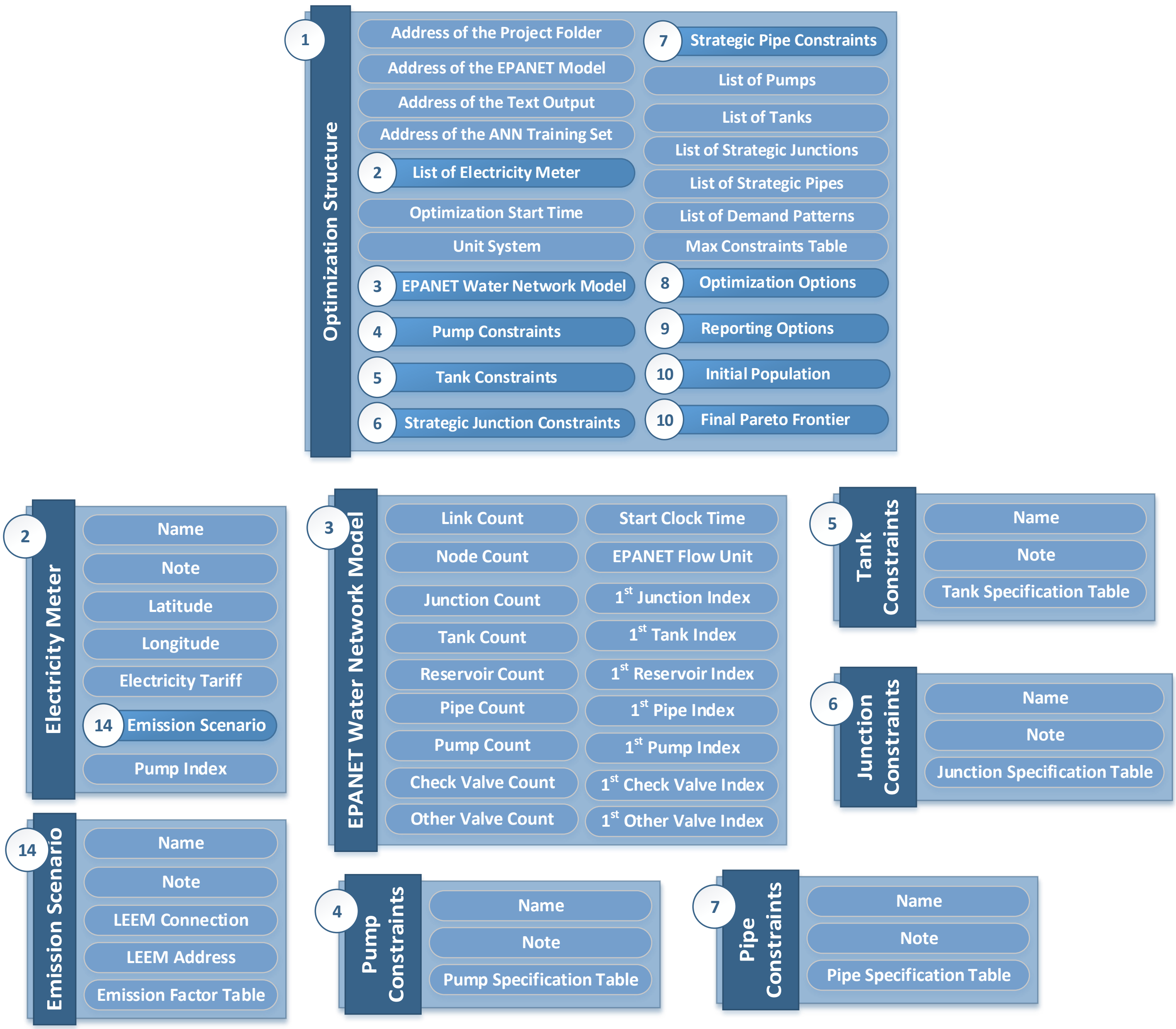
***Training Set:*** a set of input and output values for training an ANN. After training ANN, it is expected to give inputs to ANN and receive outputs within an acceptable range of error.

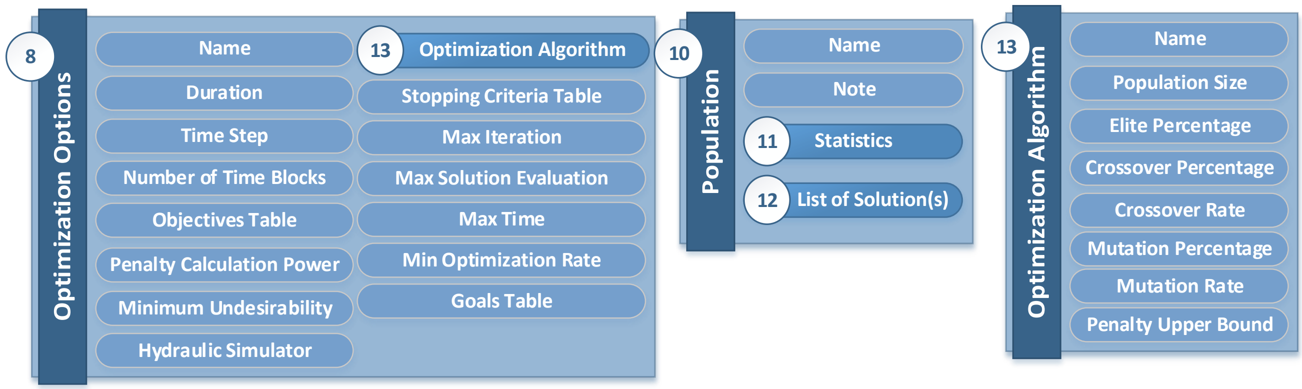
***Undesirability Index (UI):*** an arbitrary value that shows how far a pump schedule is from the ideal condition. UI can have positive, negative or zero value. Zero UI is an ideal condition. In an ideal condition, pump schedule does not have any NPW, PHW, PFW, TLP, JPP. Each cell of a pump schedule can have an UI value

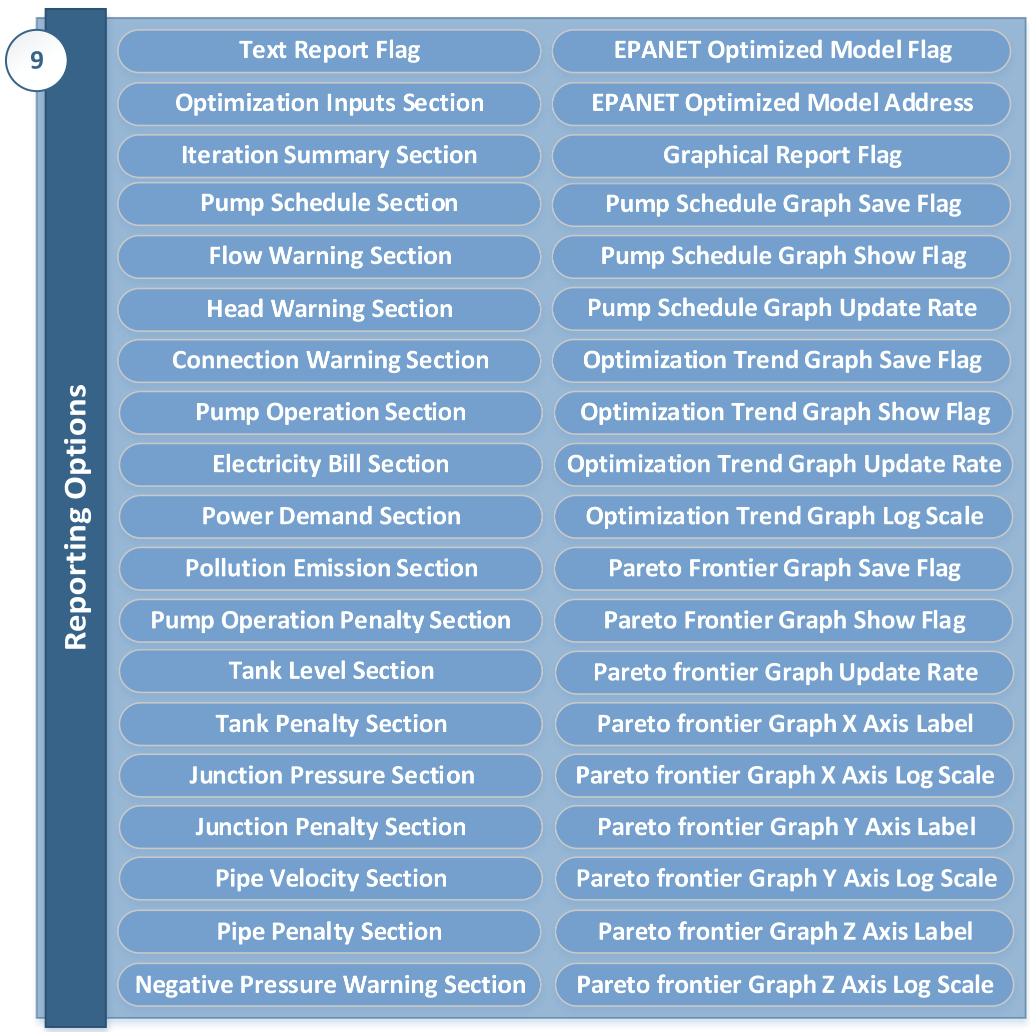
# APPENDIX B

## PEPSO Data Structures

PEPSO uses the optimization structure to input & output data of an optimization run. The following diagrams show the optimization structure and all its sub-structures. Each item that is numbered has been expanded later.




8
Optimization Options
Name
Duration
Time Step
Number of Time Blocks
Objectives Table
Penalty Calculation Power
Minimum Undesirability
Hydraulic Simulator
13
Optimization Algorithm
Stopping Criteria Table
Max Iteration
Max Solution Evaluation
Max Time
Min Optimization Rate
Goals Table
10
Population
Name
Note
11
Statistics
12
List of Solution(s)
13
Optimization Algorithm
Name
Population Size
Elite Percentage
Crossover Percentage
Crossover Rate
Mutation Percentage
Mutation Rate
Penalty Upper Bound


9
Reporting Options
Text Report Flag
Optimization Inputs Section
Iteration Summary Section
Pump Schedule Section
Flow Warning Section
Head Warning Section
Connection Warning Section
Pump Operation Section
Electricity Bill Section
Power Demand Section
Pollution Emission Section
Pump Operation Penalty Section
Tank Level Section
Tank Penalty Section
Junction Pressure Section
Junction Penalty Section
Pipe Velocity Section
Pipe Penalty Section
Negative Pressure Warning Section
EPANET Optimized Model Flag
EPANET Optimized Model Address
Graphical Report Flag
Pump Schedule Graph Save Flag
Pump Schedule Graph Show Flag
Pump Schedule Graph Update Rate
Optimization Trend Graph Save Flag
Optimization Trend Graph Show Flag
Optimization Trend Graph Update Rate
Optimization Trend Graph Log Scale
Pareto Frontier Graph Save Flag
Pareto Frontier Graph Show Flag
Pareto frontier Graph Update Rate
Pareto frontier Graph X Axis Label
Pareto frontier Graph X Axis Log Scale
Pareto frontier Graph Y Axis Label
Pareto frontier Graph Y Axis Log Scale
Pareto frontier Graph Z Axis Label
Pareto frontier Graph Z Axis Log Scale

11

**Population Statistics**

- Best Solution Inadmissibility
- Population Average Inadmissibility
- Best Solution Combined Obective
- Population Average Combined Objective
- Best Solution Electricity Cost
- Population Average Electricity Cost
- Best Solution Energy Usage Cost
- Population Average Energy Usage Cost
- Best Solution Power Demand Cost
- Population Average Power Demand Cost
- Best Solution Pollution Emission
- Population Average Pollution Emission
- Best Solution Total Penalty
- Population Average Total Penalty
- Best Solution Pump Penalties
- Population Average Pump Penalties
- Best Solution Tank Penalties
- Population Average Tank Penalties
- Best Solution Junction Penalties
- Population Average Junction Penalties
- Best Solution Pipe Penalties
- Population Average Pipe Penalties
- Iteration Counter
- Solution Evaluation Counter
- New Crossovered Solution Count
- Best Solution Index
- New Mutated Solution Count

12

**Solution**

- Pump Schedule
- Pump Operation Penalty Table
- Tank Level Table
- Peak Power Demand Table
- Power Demand Table
- Electricity Bill Table
- Pump Efficiency Table
- Pollution Emission Table
- Strategic Junction Pressure Table
- Total Penalty
- Strategic Pipe Velocity Table
- Inadmissibility Table
- Tank Penalty Table
- Rank
- Strategic Junction Penalty Table
- Previous Position
- Strategic Pipe Penalty Table
- Dominate Count
- Pump Flow Warning Table
- Crowding Distance
- Pump Head Warning Table
- Undesirability Table
- Pump Connection Warning Table
- Roulette Wheel Table
- Negative Pressure Warning Table
- Desirability Calculation Flag
- Pump Operation Table

## REFERENCES


Abiodun, F. T. and F. S. Ismail (2013). Pump Scheduling Optimization Model for Water Supply System Using AWGA. Computers & Informatics (ISCI). Langkawi, IEEE: 12-17.

Alighalehbabakhani, F., S. M. S. Abkenar, S. X. Jin, C. J. Miller, P. T. Fracasso and S. P. McElmurry (2014). "Comparative evaluation of three distinct energy optimization tools applied to real water network (Monroe)." Sustainable Computing: Informatics and Systems.

Bagirov, A., S. Ahmed, A. Barton, H. Mala-Jetmarova, A. Al Nuaimat and N. Sultanova (2012). Comparison of metaheuristic algorithms for pump operation optimization. Water Distribution Systems Analysis Conference (14th: 2012: Adelaide, S.A.). Barton, A.C.T., Engineers Australia: 886-896.

Barán, B. n., C. von Lücken and A. Sotelo (2005). "Multi-objective pump scheduling optimisation using evolutionary strategies." Advances in Engineering Software 36(1): 39-47.

Behandish, M. and Z. Y. Wu (2014). "Concurrent Pump Scheduling and Storage Level Optimization Using Meta-models and Evolutionary Algorithms." Procedia Engineering 70: 103-112.

Bentley. (2014). "WaterGEMS V8i." Retrieved 08/06/2014, 2014, from http://www.bentley.com/en-US/Products/WaterGEMS/.

Bi, W. and G. C. Dandy (2013). "Optimization of Water Distribution Systems Using Online Retrained Metamodels." Journal of Water Resources Planning and Management: 131025185630003.

Błaszczyk, J., A. Karbowski, K. Krawczyk, K. Malinowski and A. Allidina (2013). Optimal Pump Scheduling by Non-Linear Programming for Large Scale Water Transmission System. Informing Science and Information Technology Education Conference.

Broad, D. R., H. R. Maier and G. C. Dandy (2010). "Optimal Operation of Complex Water Distribution Systems Using Metamodels." Journal of Water Resources Planning and Management 136(4): 433-443.

CrepinSEK, M., S.-H. Liu and M. Mernik (2011). "Exploration and Exploitation in Evolutionary Algorithms: A Survey." ACM Computing Surveys 1(1): 33.

Chu, C.-W., M.-D. Lin, G.-F. Liu and Y.-H. Sung (2008). "Application of immune algorithms on solving minimum-cost problem of water distribution network." Mathematical and Computer Modelling 48(11-12): 1888-1900.

Cureton, E. E. (1968). "Unbiased Estimation of the Standard Deviation." The American Statistician 22(1): 22.

CWS. (2014). "Benchmarks." Retrieved 08/06/2014, 2014, from http://emps.exeter.ac.uk/engineering/research/cws/resources/benchmarks/.

Dandy, G. C., A. R. Simpson and L. J. Murphy (1996). "An Improved Genetic Algorithm for Pipe Network Optimization." Water Resources Research 32(2): 449-458.

Deb, K. (2001). Multi-Objective Optimization using Evolutionary Algorithm. England, John Wiley & Sons, Ltd.

DOE (2006). Energy Demands on Water Resources.

Fracasso, P. T., F. S. Barnes and A. H. R. Costa (2014). "Optimized Control for Water Utilities." Procedia Engineering 70: 678-687.

Fu, G. and Z. Kapelan (2011). Embedding Neural Networks in Multiobjective Genetic Algorithms for Water Distribution System Design. Water Distribution Systems Analysis 2010, American Society of Civil Engineers: 888-898.

Giacomello, C., Z. Kapelan and M. Nicolini (2013). "Fast hybrid optimization method for effective pump scheduling." Journal of Water Resources Planning and Management 139(2): 175-183.

Guidolin, M., G. Fu and P. Reed (2012). Parallel evolutionary multiobjective optimization of water distribution system design. Water Distribution Systems Analysis Conference (14th: 2012: Adelaide, S.A.). Barton, A.C.T., Engineers Australia: 113-125.

Hashemi, S. S., M. Tabesh and B. Ataeekia (2013). "Ant-colony optimization of pumping schedule to minimize the energy cost using variable-speed pumps in water distribution networks." Urban Water Journal: 1-13.

Holland, M. and P. Watkiss (2002). Benefits Table database: Estimates of the marginal external costs of air pollution in Europe, European Commission DG Environment: 20.

Jamieson, D. G., U. Shamir, F. Martinez and M. Franchini (2007). "Conceptual design of a generic, real-time, near-optimal control system for water-distribution networks." Journal of Hydroinformatics 9(1): 3.

Jolly;, M. D., A. D. Lothes;, L. S. Bryson; and L. Ormsbee (2014). "Research Database of Water Distribution System Models." Journal of Water Resources Planning and Management 140(4): 410-416.

Jowitt, P. W. and G. Germanopoulos (1992). "Optimal Pump Scheduling in Water-Supply Networks." Journal of Water Resources Planning and Management 118(4): 406-422.

Kalanithy Vairavamoorthy and M. Ali (2000). "Optimal Design ofWater Distribution Systems Using Genetic Algorithms." Computer-Aided Civil and Infrastructure Engineering 15(5): 374 - 382.

Kriesel, D. (2007). A Brief Introduction to Neural Networks: 244.

Laurie J Murphy, G. C. Dandy and A. R. Simpson (1994). Optimum Design and Operation of Pumped Water Distribution Systems. Hydraulics in Civil Engineering. Australia, Brisbane, Institution of Engineers: 1-13.

Liu, H., Y. Yuan, M. Zhao, H. Fang and H. Zhao (2011). Hybrid Multi-Objective Genetic Algorithm for Optimal Design of Water Supply Network. Water Distribution Systems Analysis 2010, American Society of Civil Engineers: 899-908.

Lopez-Ibanez, M. (2009). Operational Optimisation of Water Distribution Networks. PhD, Edinburgh Napier University.

Lopez-Ibanez, M., T. D. Prasad and B. Paechter (2008). "Ant Colony Optimization for Optimal Control of Pumps in Water Distribution Networks." Journal of Water Resources Planning and Management 134(4): 337-346.

Marchi, A., G. Dandy, A. Wilkins and H. Rohrlach (2014). "Methodology for Comparing Evolutionary Algorithms for Optimization of Water Distribution Systems." Journal of Water Resources Planning and Management 140(1): 22-31.

Martínez, F., V. Hernández, J. M. Alonso, Z. Rao and S. Alvisi (2007). "Optimizing the operation of the Valencia water-distribution network." Journal of Hydroinformatics 9(1): 65.

MathWorks (2015). MATLAB. Natick, Massachusetts, The MathWorks Inc. R2015a.

Milan Čistý and Z. Bajtek (2009). Hybrid Method for Optimal Design of Water Distribution System. International Symposium on Water Management and Hydraulic Engineering. Ohrid/Macedonia: 235-246.

Miller, C. J., M. M. Rogers and S. M. S. A. (2014). "PEPSO." Sustainable Water Delivery Retrieved Nov. 2014, 2014, from http://engineering.wayne.edu/wsuwater/hydraulics/pepso.php.

Morley, M. S. (2008). A Framwork for Evolutionary Optimization Applications In Water Distribution Systems. Ph.D., University of Exeter.

MSDN. (2015). "Desktop App User Interface." Windows desktop apps Retrieved 07 December 2015, 2015, from https://msdn.microsoft.com/enus/library/ff657751(v=vs.85).aspx.

Narzisi, G. (2008). Multi-Objective Optimization, A quick introduction. New York, Courant Institute of Mathematical Sciences, New York University: 35.

Nissen, S. (2003). FANN. Fast Artificial Neural Network Library. Copenhagen, Denmark, University of Copenhagen.

Nissen, S. (2003). Implementation of a Fast Artificial Neural Network Library (fann). Copenhagen, Denmark, Department of Computer Science of University of Copenhagen: 92.

Osborn, C. E. (2006). Statistical Applications for Health Information Management, Jones & Bartlett Learning.

Pelikan, B. (2009). The pump book: a pump installer's guide to the design installation and servicing of groundwater pumping systems. Morrisville, SC, Lulu Press.

Preis, A., M. Allen and A. Whittle (2011). On-Line Hydraulic Modeling of a Water Distribution System in Singapore. Water Distribution Systems Analysis 2010: 1336-1348.

Rao, Z. and F. Alvarruiz (2007). "Use of an artificial neural network to capture the domain knowledge of a conventional hydraulic simulation model." Journal of Hydroinformatics 9(1): 15.

Razavi, S., B. A. Tolson and D. H. Burn (2012). "Review of surrogate modeling in water resources." Water Resources Research 48(7): n/a-n/a.

Rogers, M. M., Y. Wang, C. Wang, S. P. McElmurry and C. J. Miller (2013). "Evaluation of a rapid LMP-based approach for calculating marginal unit emissions." Applied Energy 111: 812-820.

Rossman, L. A. (1999). The EPANET Programmer's Toolkit for Analysis of Water Distribution Systems. WRPMD'99: Preparing for the 21st Century. Tempe, Arizona, United States, American Society of Civil Engineers: 10.

Rossman, L. A. (2000). EPANET 2 USERS MANUAL. USEPA. Cincinnati, OH, US EPA, Water Supply and Water Resources Division, National Risk Management Research Laborator.

Rossman, L. A. (2000). EPANET 2 User’s Manual. Cincinnati, OH, US EPA, Water Supply and Water Resources Division, National Risk Management Research Laborator.

Rossman, L. A. (2008). EPANET. Cincinnati, Ohio, United States Environmetal Protection Agency.

Sadatiyan Abkenar, S. M., S. D. Stanley, C. J. Miller, D. V. Chase and S. P. McElmurry (2014). "Evaluation of genetic algorithms using discrete and continuous methods for pump optimization of water distribution systems." Sustainable Computing: Informatics and Systems.

Samani, H. M. V. and A. Mottaghi (2006). "Optimization of water distribution networks using integer linear programming." Journal of Hydraulic Engineering-Asce 132(5): 501-509.

Shamir, U. and E. Salomons (2008). "Optimal Real-Time Operation of Urban Water Distribution Systems Using Reduced Models." Journal of Water Resources Planning and Management 134(2): 181-185.

Siew, C. and T. Tanyimboh (2011). "Design of the Anytown Network Using the Penalty-Free Multi-Objective Evolutionary Optimization Approach." 190-198.

Siew, C. and T. T. Tanyimboh (2010). Penalty-free multi-objective evolutionary optimization of water distribution systems. Proceedings of the 12th Annual Water Distribution Systems Analysis Conference, WDSA.

Siew, C., T. T. Tanyimboh and A. G. Seyoum (2013). "Assessment of Penalty-Free Multi-Objective Evolutionary Optimization Approach for the Design and Rehabilitation of Water Distribution Systems." Water Resources Management 28(2): 373-389.

Simpson, A. R., G. C. Dandy and L. J. Murphy (1994). "Genetic Algorithms Compared to Other Techniques for Pipe Optimization." Journal of Water Resources Planning and Management 120(4): 423-443.

Stokes, C. S., A. R. Simpson and H. R. Maier (2012). An improved framework for the modelling and optimisation of greenhouse gas emissions associated with water distribution systems. 6th International Congress on Environmental Modelling and Software. Leipzig, Germany, International Environmental Modelling and Software Society (iEMSs): 818-825.

Streimikiene, D., I. Roos and J. Rekis (2009). "External cost of electricity generation in Baltic States." Renewable and Sustainable Energy Reviews 13(4): 863-870.

Tang, Y., G. Zheng and S. Zhang (2014). "Optimal control approaches of pumping stations to achieve energy efficiency and load shifting." International Journal of Electrical Power & Energy Systems 55: 572-580.

Ting, C.-K. (2005). On the Mean Convergence Time of Multi-parent Genetic Algorithms Without Selection. 8th European Conference, ECAL 2005, Canterbury, UK, September 5-9, 2005. Proceedings, Springer Berlin Heidelberg: 403-412.

Tolson, B. A., M. Asadzadeh, H. R. Maier and A. Zecchin (2009). "Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization." Water Resources Research 45(12): n/a-n/a.

Ulanicki, B., J. Kahler and H. See (2007). "Dynamic Optimization Approach for Solving an Optimal Scheduling Problem in Water Distribution Systems." Journal of Water Resources Planning and Management 133(1): 23–32.

van Zyl, J. E. (2001). A methodology for improved operational optimization of water distribution systems. Ph.D., University of Exeter.

van Zyl, J. E., D. A. Savic and G. A. Walters (2004). "Operational Optimization of Water Distribution Systems Using a Hybrid Genetic Algorithm." Journal of Water Resources Planning and Management 130(2): 160-170.

Vogt, W. P. and R. B. Johnson (2011). Dictionary of Statistics & Methodology: A Nontechnical Guide for the Social Sciences: A Nontechnical Guide for the Social Sciences, SAGE Publications.

Walski, T. and W. Hartell (2012). Understanding energy pricing for water pumping. Water Distribution Systems Analysis Conference (14th: 2012: Adelaide, S.A.). Barton, A.C.T., Engineers Australia: 877-885.

Wang, H., S. Liu, F. Meng and M. Li (2012). "Gene Expression Programming Algorithms for Optimization of Water Distribution Networks." Procedia Engineering 37: 359-364.

Wang, J.-Y., T.-P. Chang and J.-S. Chen (2009). "An enhanced genetic algorithm for bi-objective pump scheduling in water supply." Expert Systems with Applications 36(7): 10249-10258.

Wang, J. Y., F. G. Chen and J. S. Chen (2013). "A green pump scheduling algorithm for minimizing power consumption and land depletion." Concurrent Engineering-Research and Applications 21(2): 121-128.

Wang, Q., M. Guidolin, D. Savic and Z. Kapelan (2014). "Two-Objective Design of Benchmark Problems of a Water Distribution System via MOEAs: Towards the Best-Known Approximation of the True Pareto Front." Journal of Water Resources Planning and Management: 04014060.

Wikipedia-contributors. (2002, 31 July 2014 16:45 UTC). "Pareto efficiency." Retrieved 20 August 2014, 2014, from http://en.wikipedia.org/w/index.php?title=Pareto_efficiency&oldid=619311620.

Wikipedia-contributors. (2002, 25 July 2014 15:59 UTC). "Three Gorges Dam." Retrieved 25 July 2014, 2014, from http://en.wikipedia.org/w/index.php?title=Three_Gorges_Dam&oldid=618431582.

Wikipedia-contributors. (2003, 24 July 2014 20:56 UTC). "observable universe." Retrieved 25 July 2014, 2014, from http://en.wikipedia.org/w/index.php?title=Observable_universe&oldid=618325231

Wikipedia-contributors. (2004, 7 June 2014 13:51 UTC). "Deterministic algorithm." Retrieved 17 August 2014 00:25 UTC, 2014, from http://en.wikipedia.org/w/index.php?title=Deterministic_algorithm&oldid=611948347.

Wikipedia-contributors. (2005, 16 April 2014 05:01 UTC). "Margolus–Levitin theorem." Retrieved 25 July 2014, 2014, from http://en.wikipedia.org/w/index.php?title=Margolus%E2%80%93Levitin_theorem&oldid=604404418.

Wikipedia-contributors. (2013, 7 December 2013 16:02 UTC). "EXPTIME." Retrieved 30 July 2014, 2014, from http://en.wikipedia.org/w/index.php?title=EXPTIME&oldid=585003601.

Wikipedia-contributors. (2014, 18 October 2014). "Metaheuristic." Retrieved 22 November 2014, from http://en.wikipedia.org/w/index.php?title=Metaheuristic&oldid=630095472.

Wikipedia-contributors. (2015, 1 December 2015). "User interface." 693219930. Retrieved 7 December 2015, 2015, from https://en.wikipedia.org/w/index.php?title=User_interface&oldid=693219930.

Wu, W., H. R. Maier and A. R. Simpson (2010). "Single-Objective versus Multiobjective Optimization of Water Distribution Systems Accounting for Greenhouse Gas Emissions by Carbon Pricing." Journal of Water Resources Planning and Management 136(5): 555-565.

Wu, W., A. R. Simpson and H. R. Maier (2012). "Sensitivity of Optimal Tradeoffs between Cost and Greenhouse Gas Emissions for Water Distribution Systems to Electricity Tariff and Generation." Journal of Water Resources Planning and Management 138(2): 182-186.

Wu, W., A. R. Simpson, H. R. Maier and A. Marchi (2012). "Incorporation of Variable-Speed Pumping in Multiobjective Genetic Algorithm Optimization of the Design of Water Transmission Systems." Journal of Water Resources Planning and Management 138(5): 543-552.

Wu, W. Y., H. R. Maier and A. R. Simpson (2013). "Multiobjective optimization of water distribution systems accounting for economic cost, hydraulic reliability, and greenhouse gas emissions." Water Resources Research 49(3): 1211-1225.

Wu, Z. Y. and M. Behandish (2012). Comparing methods of parallel genetic optimization for pump scheduling using hydraulic model and GPU-based ANN meta-model.

Wu, Z. Y. and A. R. Simpson (2001). "Competent genetic-evolutionary optimization of water distribution systems." Journal of Computing in Civil Engineering 15(2): 89-101.

Zecchin, A. C., H. R. Maier, A. R. Simpson, M. Leonard and J. B. Nixon (2007). "Ant Colony Optimization Applied to Water Distribution System Design: Comparative Study of Five Algorithms." Journal of Water Resources Planning and Management 133(1): 87-92.

Zheng, F. (2013). Advanced hybrid approaches based on graph theory decomposition, modified evolutionary algorithms and deterministic optimisation techniques for the design of water distribution systems. Ph.D., University of Adelaide.

Zheng, F., A. R. Simpson and A. C. Zecchin (2014). "An efficient hybrid approach for multiobjective optimization of water distribution systems." Water Resources Research 50(5): 3650-3671.

Zheng, F., A. Zecchin and A. Simpson (2012). A performance comparison between genetic algorithms and differential evolution variants for water network optimisation. Water Distribution Systems Analysis Conference (14th : 2012 : Adelaide, S.A.). Barton, A.C.T., Engineers Australia: 956-965.

Zheng, Y. W. and B. Morad (2012). Real-time pump scheduling using genetic algorithm and artificial neural network based on graphics processing unit. Water Distribution Systems Analysis Conference (14th : 2012 : Adelaide, S.A.). Barton, A.C.T., Engineers Australia: 1088-1099.

# ABSTRACT


## ENHANCED PUMP SCHEDULE OPTIMIZATION FOR LARGE WATER DISTRIBUTION NETWORKS TO MAXIMIZE ENVIRONMENTAL AND ECONOMIC BENEFITS

by

**S. MOHSEN SADATIYAN A.**

**May 2016**

**Advisor:** Dr. Carol J. Miller

**Major:** Civil & Environmental Engineering

**Degree:** Doctor of Philosophy

For more than four decades researchers tried to develop optimization method and tools to reduce electricity consumption of pump stations of water distribution systems. Based on this ongoing research trend, about a decade ago, some commercial pump operation optimization software introduced to the market. Using metaheuristic and evolutionary techniques (e.g. Genetic Algorithm) make some commercial and research tools able to optimize the electricity cost of small water distribution systems (WDS). Still reducing the environmental footprint of these systems and dealing with large and complicated water distribution system is a challenge.

In this study, we aimed to develop a multiobjective optimization tool (PEPSO) for reducing electricity cost and pollution emission (associated with energy consumption) of pump stations of WDSs. PEPSO designed to have a user-friendly graphical interface besides the state of art internal functions and procedures that lets users define and run customized optimization scenarios for even medium and large size WDSs. A customized version of non-dominated sorting genetic algorithm II is used as the core optimizer algorithm. EPANET toolkit is used as the hydraulic solver of PEPSO. In addition to the

EPANET toolkit, a module is developed for training and using an artificial neural network instead of the high fidelity hydraulic model to speed up the optimization process. A unique measure that is called “Undesirability” is also introduced and used to help PEPSO in finding the promising path of optimization and making sure that the final results are desirable and practical.

PEPSO is tested for optimizing the detailed hydraulic model of WDS of Monroe city, MI, USA and skeletonized hydraulic model of WDS of Richmond, UK. The various features of PEPSO are tested under 8 different scenarios, and its results are compared with results of Darwin Scheduler (a well-known commercial software in this field). The test results showed that in a reasonable amount of time, PEPSO is able to optimize and provide logical results for a medium size WDS model with 13 pumps and thousands of system components under different scenarios. It also is concluded that this tool in many aspects can provide better results in comparison with the famous commercial optimization tool in the market.

## AUTOBIOGRAPHICAL STATEMENT

When I was a teenager, I realized that, there were a lot of people in the past and around the world that their works affected my life. The positive effect of other's life on my living (beyond the international borders and time periods) plant the seed of an idea in my mind. The idea that I also can gain knowledge, work hard to add something good to the common heritage of humanity. It motivated me to study and use the practical knowledge to create a better self and surrounding world.

I was born and raised in Tehran, Iran. By the time of going to high school, I had a great passion to learn more about science. In parallel, my interest about playing video games, absorbed my attention to computer graphics. Before going to university, I was at a dilemma to choose one of the two paths of study engineering or digital graphic design.

Goethe said "The person born with a talent they are meant to use, will find their greatest happiness in using it." I believed that I was in talented in engineering field so I started to be a civil engineering. Although at first I was not sure about my choice, after a while, I was attracted by practicality of materials in civil engineering field. I finished the BSc in 3.5 years. During undergraduate studies, I attended some courses about water & wastewater engineering that were tough by exceptional instructors. This motivated me to follow the graduate education in the field of water & wastewater engineering in Power & Water University of Technology of Tehran, Iran. After two years I gained a MSc. degree. Going to this university was a great opportunity for me to learn practical knowledge from instructors and advisors who had a lot of practical experiences.

I started to work in a water and wastewater engineering consultant firm. I worked for two years there and cooperated in designing industrial and urban water and wastewater treatment plants. I also was lucky to find a position in National Water and Wastewater Company of Iran. It was a valuable experience for me to work on some national projects that were financed and managed by public sector. Working encouraged me to pursue my study to gain more knowledge and get a PhD degree.

To pursue my studies in an international institute, I applied for a PhD position at WSU and Dr. Miller accepted me to start working under her supervision. Most of the research project that I worked on during the PhD studies, was related to using pollutant emission data to find the optimum pump operation plan of a water distribution facilities to consume less energy and reduce related pollutant emission. While I was working on this research, Dr. Miller, helped me to find a position to work on the operation plan of the Detroit water distribution system that is one the largest water distribution systems in the US. It was a practical and valuable experience that helped me a lot to make my study closer to the practical needs of water operators to reduce energy usage of their systems. Finally, I finished this PhD thesis by developing a computer software that helps operators to find the optimum pump schedule of a water distribution system. I hope this study and the tool that is developed (PEPSO) I can help researcher and water system operators to get one step closer to a more sustainable water system.