# An Interface for Coupling Optimization Algorithms With EPANET in Discrete Event Simulation Platforms

Lawrence K. Letting*, Yskandar Hamam*, Adnan M. Abu-Mahfouz*†
*Department of Electrical Engineering
Tshwane University of Technology
Pretoria, South Africa
E-mail: LettingLK@gmail.com, HamamA@tut.ac.za
†CSIR Meraka Institute, Pretoria, South Africa
Email: AAbuMahfouz@csir.co.za

***Abstract*—The application of simulation optimization in water distribution network analysis and design is a promising method for generating solutions to existing challenges. The absence of a standard interface for coupling the open source EPANET software package to optimization algorithms increases the implementation effort and limits the comparison of results. This work presents a methodology for implementing an interface for coupling optimization algorithms with EPANET. The proposed technique uses the internal simulation clock events in a discrete event simulation platform to co-ordinate optimization loops and data exchange. The utilization of intermediate input/output files is avoided in order to increase the simulation speed. A water distribution network implemented in the EPANET solver is considered as a discrete event to be interfaced with optimization algorithms. The interface module is implemented as a C/C++ mex-file for EPANET in the MATLAB/Simulink platform. The methodology enables the user to evaluate the fitness of the design parameters with easy access to data logging and visualization tools at run-time. The proposed technique is used to implement the particle swarm optimization algorithm (PSO) and applied to design a benchmark water distribution network.**

## I. Introduction

Water distribution systems (WDSs) perform the crucial role of supplying safe drinking water to the public. The main goal in WDS operation is to meet the desired water demand while ensuring the appropriate water quality and pressure is met in all the nodes. A water network model is a hybrid system consisting of both continuous dynamics and discrete events. Water flow from the sources to the demand nodes is a continuous event with time varying flow rates. Discrete events include operation of control valves, operation of pumps, and the scheduling of water supply patterns.

The EPANET hydraulic modelling software package [1] is a standard open source tool for modelling water distribution systems. EPANET software is available as a standalone package with a graphical user interface or as a dynamic library of functions with source code in C language. Parts of this software have been rewritten by different users in order to conform to various user needs. An object–oriented implementation using C++ [2] and Python language [3] have been developed. An optimization interface enables the faster implementation of techniques to address water network distribution problems such as pipe diameter selection, leakage detection, state estimation, and sensor placement. The EPANET solvers and other water network analysis tools have been used to implement algorithms to solve problems such as leakage minimisation [4], sensor placement [5], and pressure control [6]–[9]. The contribution of this work is the formulation and implementation of a methodology for interfacing EPANET to optimization algorithms. A water distribution network implemented in the EPANET solver is considered as a discrete event.

### A. Discrete event simulation

A discrete event simulation is one in which the model is updated at a discrete set of simulated time points, which may be of unequal size, called event times [10]. The simulation clock tracks the passage of time and advances in discrete steps when all the actions defined in the current event (such as a call to the EPANET engine) have been carried out. The actual length of time needed to perform computations may vary from one event to the next.

A MATLAB based discrete event simulation system is implemented in [11] and switch-case statements are used to control events. In this work, discrete event simulation is implemented using C-Mex S-functions which enables implementation of flow control using timed events. A user defined C-Mex S-function provides the most advanced programming flexibility [12]. The developed interface provides a seamless connection to the EPANET engine functions whose source code is available in C language.

### B. Functions of an optimization interface

In order to utilize EPANET to evaluate an objective function during parameter design, an interface or wrapper module is required to perform: (1) initialization and calls to the EPANET functions; (2) data exchange between EPANET and the optimization algorithm during parameter update; (3) controlling the flow and determining whether the optimization process is

978-1-5386-0836-4/17/$31.00 ©2017 IEEE

complete e.g. using the set number of iterations or when the objective function cost is below a given threshold; and (4) data logging and visualization of the results at run-time. The way the identified functions are addressed in a discrete event simulation platform determines the efficiency and accuracy of the optimization process.

### C. Related work

An EPANET–MATLAB toolkit [13] provides an *m*-file programming interface for accessing the EPANET functions, simulating, and visualizing simulation results. The methodology developed in this work is dedicated to the problem of interfacing EPANET with optimization algorithms. This enables interfacing of EPANET with the Mathworks Simulink engine [14] and other software platforms that support discrete event simulation. The functionality of data logging and visualization is not addressed in this work since it can be effectively performed by the simulation software platform where EPANET is hosted. The proposed interface addresses the challenges of initialization, flow control and data exchange when interfacing EPANET with an optimization algorithm in a discrete event simulation environment.

The next section presents the formulation of the water distribution network design problem. Section III presents the implementation of the optimization interface while section IV presents an introduction to the PSO algorithm. Simulation results and discussion are presented in section V and section VI concludes the study.

## II. Water distribution network design problem

The optimal design problem in water distribution systems has been researched extensively with the aim of minimizing the construction cost [15], [16]. The major parameter of interest is sizing of pipe diameters for different WDS links. The optimal design problem is therefore modelled as a static optimization problem. The long-term management decisions which include operation of pumps and scheduling of water supply and demand patterns also need to be addressed. The water pipe diameter selection problem has been addressed using different methods and is classified as NP-Hard. The application of population based meta-heuristic algorithms has gained prominence over the years. Meta-heuristic algorithms solve the WDS design problem by considering the water network model solution as a black-box objective function. In this work, the particle swarm optimization (PSO) algorithm is used.

Simulation optimization aims to minimize the total construction cost of the water distribution network given by (1), where $n$ is the number of pipes in the network, $C(d_i)$ is the unit cost of a commercial pipe of diameter $d_i$, and $L_i$ is the length of the pipe. The pipe selection is constrained by the set of commercially available diameters, [D], given by (2), where $N$ is the number of available diameters. The optimization problem is solved subject to the constraints given by (3)–(6). The satisfaction of the constraints is tested using the water network solution from EPANET.

$$T_c = \sum_{i=1}^{n} C(d_i) L_i \tag{1}$$

$$d_i \in [D] \quad \forall i \in N \tag{2}$$

$$\sum_i Q_{in} - \sum_i Q_{out} = Q_{ex} \tag{3}$$

$$\sum_i K_i Q_i^p = 0 \tag{4}$$

$$H_i \geq H_{min} \tag{5}$$

$$V_{min} \leq V_{ij} \leq V_{max} \tag{6}$$

The first constraint is the law of conservation of mass at each node which requires that the net flows must be equal to zero. The application of nodal conservation of mass expressed in terms of pipe flow rates is given by (3). The sum of flows into the node ($Q_{in}$) minus the flows out of the node ($Q_{out}$) is equal to the external demand ($Q_{ex}$).

The constraint in (4) is the law of conservation of energy in a closed loop, where $K_i$ is the hydraulic resistance in pipe link $i$ and $p$ is an exponent whose value depends on the head-loss formula utilized in the analysis. The conservation of energy constraint requires that the sum of frictional energy lost over any path, belonging to a closed loop is zero if there are no power pumps.

Constraint (5) refers to the minimum pressure head requirements, where $H_n$ is the pressure at node $n$, and $H_{min}$ is the minimum pressure required. A minimum pressure head at each node is required to satisfy the water demand while guaranteeing appropriate network operation. The minimum and maximum water velocity constraint (6) in each link may also be imposed.

The benchmark Hanoi network [17] is implemented in EPANET. The Hanoi water distribution network has 32 nodes and 34 pipes organized in 3 loops. There are no pumps and there is one fixed head source at an elevation of 100 m. All the nodes are at the same ground level and the minimum pressure head requirement at all nodes is fixed at 30 m. There are 6 commercially available diameters of 12, 16, 20, 24, 30, 40 inches. The layout of the Hanoi network is shown in Fig. 1. The network data is obtained from [18].

## III. Implementation of the optimization Interface

The optimization scheme is implemented as shown in Fig. 2. The developed interface ensures that there is flow control without using explicit loop control statements. The interface enables discrete event driven flow control during simulation optimization.

The events of the simulation-optimization scheme of Fig. 2 are co-ordinated as shown in Fig. 3 using the optimization interface. The simulation engine of the host software issues the commands to start the simulation and periodic updates of the clock time. The first step involves setting up the fundamental simulation sample time $T_s$ seconds. The value of $T_s$ is determined by performing an experiment to determine the length of time the EPANET solvers take to complete a single

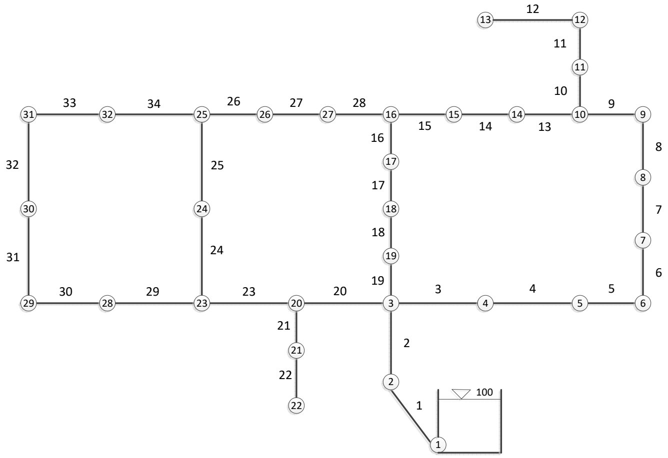

Fig. 1. The layout of the Hanoi network [18]

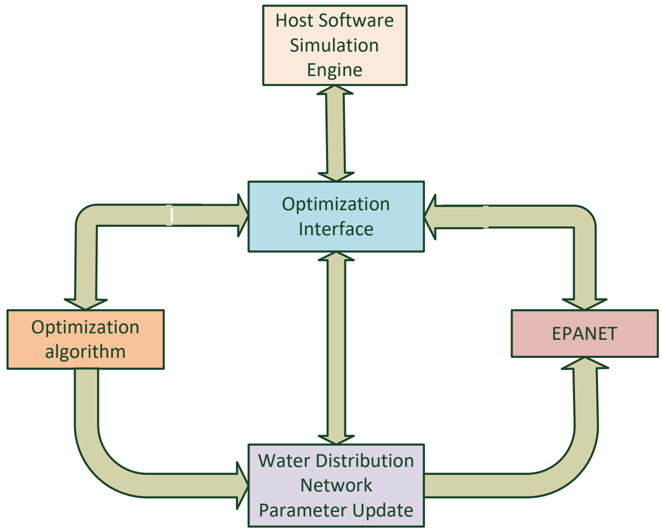


Fig. 2. Architecture of the optimization scheme

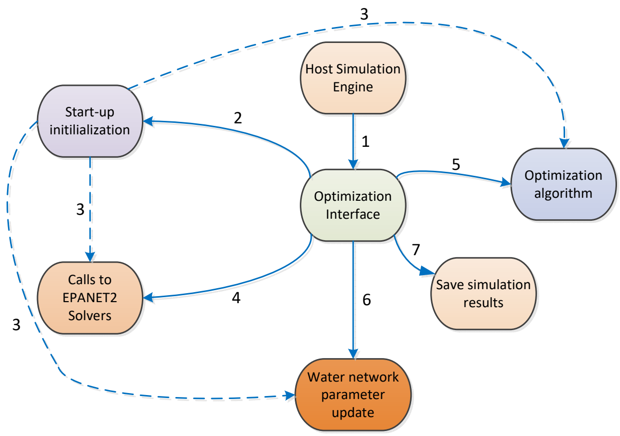


Fig. 3. Co-ordination of events during simulation

period analysis call. The host simulation engine advances the clock time periodically. The interface is used to co-ordinate events after every $T_s$ seconds. At the start of the simulation, the initialization module is invoked in order to initialize the optimization algorithm, the water distribution network parameter routine, and initialization of EPANET hydraulic analysis module. The initialization is carried out as shown in Fig. 4. A reference for the EPANET function calls is available in the programmer's toolkit [19].

The optimization algorithm generates a population of particles which are used to set the water network parameters in EPANET at the beginning of each call. The EPANET solvers are also initialized before each call in order to ensure that the initial conditions are the same for each of the set of parameters used to run the system. The procedure for water network simulation using EPANET is summarized in Fig. 5. At the end of each call to EPANET, the water parameter update routine retrieves the hydraulic results and calculates the fitness of the set of parameters used to run the water network.

The optimization algorithm events are processed as shown in Fig. 6. When all the potential set of parameters in the population have been used to run the water network design model, the PSO algorithm is used to update the population and a new iteration is initiated. The process ends when the set number of iterations is reached and the results are saved.

## IV. Particle Swarm Optimization Algorithm

Particle swarm optimisation (PSO) is a population–based metaheuristic algorithm based on the movement and intelligence of swarms. It applies the concept of social interaction to problem solving. The algorithm was developed by Kennedy and Eberthart in 1995 for simulating the flight patterns of birds, which is governed by three factors: avoiding collision, matching the velocity, and flock centering [20]. The main goal of the flight pattern is to find a place with enough food. Kennedy and Eberthart observed that the bird flocking behaviour can be applied in optimisation using a population of potential solutions called particles that are flown through a *D*–dimensional search space. At time–step $k$ each particle $i$ is represented by the position vector $x_k$ and velocity $v_k$.

The instantaneous position of the particles in a swarm that consists of $N$ particles is given by:

$$X_k = [x_1, x_2, \ldots, x_N] \tag{7}$$

where $x_i$ represents a parameter of the problem that has to be optimised, and $k$ is the time-step. Initially, each particle position is randomly generated and the particle then moves with a random velocity subject to the boundary conditions. At the time step $k+1$, the velocity of the $i_{th}$ particle is given by:

$$\begin{aligned} v_{k+1} = & w v_k + C_1 \cdot rand() \cdot [P_{best} - x_k] + \\ & C_2 \cdot rand() \cdot [G_{best} - x_k] \end{aligned} \tag{8}$$

where $w$ is the inertia factor; $C_1$ and $C_2$ are the social and cognitive rate, respectively; $P_{best}$ is the best position ever found by the particle during its motion; $G_{best}$ is the best position discovered by the entire swarm, and *rand()* is a

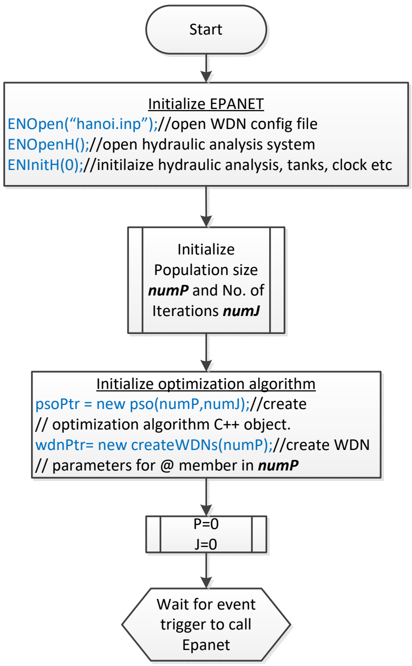


Fig. 4. Start-up initialization events

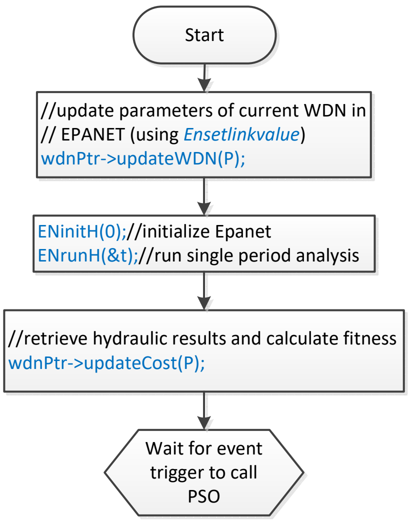


Fig. 5. Processes during calls to EPANET

uniform random number generator between 0 and 1. The new position of each particle is then evaluated using:

$$x_{k+1} = x_k + V_k \tag{9}$$

If a particle position violates the boundary conditions, the current velocity is set to zero and a new velocity is evaluated using (10) and a new position is re-evaluated using (9).

$$\begin{aligned} v_{k+1} = C_1 \cdot rand() \cdot [P_{best} - x_k] + \\ C_2 \cdot rand() \cdot [G_{best} - x_k] \end{aligned} \tag{10}$$

The performance of the PSO algorithm is sensitive to the particle velocity. The inertia factor $w$ is therefore selected

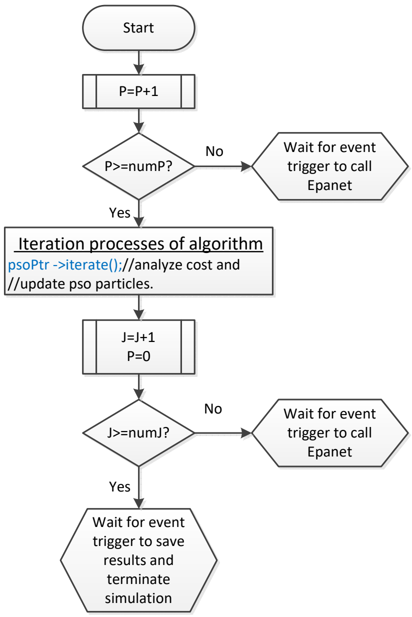


Fig. 6. Optimization algorithm

using (11) [21]:

$$w = w_{max} - \frac{k\,(w_{max} - w_{min})}{N} \tag{11}$$

where $w_{max}$ and $w_{min}$ are the maximum and minimum values of inertia factor respectively, $k$ is the iteration counter, and $N$ is the total number of iterations.

## V. Simulation Results and Discussion

The optimization process was carried out in Simulink using the discrete event simulation scheme presented in Fig.3. PSO initialization parameters are given in Table I. The cost function consists of the pipe cost and the nodal pressure head penalty cost expressed as (12).

$$f_{cost} = T_c + P_{pc} \tag{12}$$

$$P_{pc} = \sum_i (T_p - P_i) * K_{ph1} + (P_i - T_p) * K_{ph2} \tag{13}$$

Where, $T_c$ is the total pipe cost given by (1), $P_{pc}$ is the pressure penalty cost, $T_p$ is the target pressure, $P_i$ is the measured pressure at node $i$, $K_{ph1}$ is the below target pressure penalty, and $K_{ph2}$ is the above target pressure penalty. The choice of penalty scaling factors has a major influence on the convergence of the simulation. The above target pressure penalty $K_{ph2}$ was set to zero since the major aim is to meet minimum nodal pressure head of 30m at minimum construction cost. To ensure fast convergence, the absolute value of $P_{PC}$ should be in the range of $1 \times 10^6$ since the construction cost is in millions of dollars. Therefore, $K_{ph1}$ was set at $10 \times 10^6$. The default settings of EPANET 2 software package were used in the simulation.

Since the movement of the PSO particles is continuous, the 6 pipe diameters available for designing the network are

TABLE I
INITIALIZATION OF PSO PARAMETERS

| Parameter | Value |
|---|---|
| Population ($P$) | 80 |
| Number of iterations ($N$) | 500 |
| Inertia factor ($w_{max}, w_{min}$) | 0.5, 0.05 |
| Social rate ($C_1$) | 2.0 |
| Cognitive rate ($C_2$) | 2.0 |

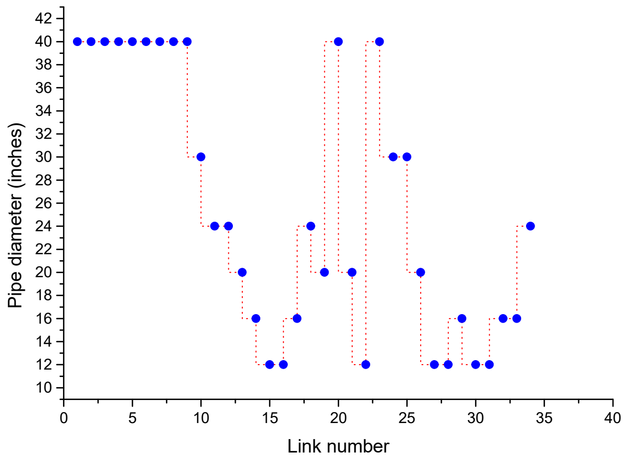


Fig. 7. Selected link pipe diameters for the Hanoi network

placed in a sorted array and the particles are to determine the optimal array index. The current particle position $x$ is therefore rounded to the nearest integer in the range 0 to 5 and used to select the corresponding pipe diameter. The particle movement is constrained by the range of the array index.

The selected optimal pipes for the Hanoi network are as shown in Fig. 7. The selected optimal pipes gives the nodal pressure distribution shown in Fig. 8. The pressure at all the nodes is above the target pressure head. The lowest heads are at nodes 12 and 28 with 30.007m and 30.013m respectively. The selected pipes give the global best known pipe cost of $6.081 \times 10^6$ dollars. Although PSO has been reported as the worst performing algorithm in terms of efficiency [16] during pipe sizing of the Hanoi network, it was able to obtain the global minimum after 220 iterations in this study. The convergence of the pipe cost during optimization is shown in Fig. 9. The optimization interface presented this work and the statistical comparison methods presented in [16] contributes to the development of a platform for fair comparison of algorithms in water network design.

## VI. CONCLUSION

This paper presents a methodology for implementing an optimization interface for EPANET water network analysis software. The optimization interface is implemented and coupled with EPANET in a discrete event simulation platform. The developed interface is validated by using it to couple EPANET and the particle swarm optimization (PSO) algorithm in MATLAB/Simulink. The optimization scheme is applied for pipe diameter selection in the benchmark Hanoi water

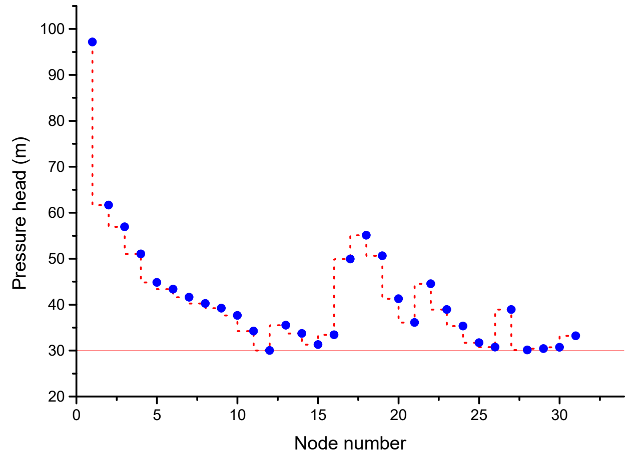


Fig. 8. Comparison of target pressure head and the actual values for the Hanoi network

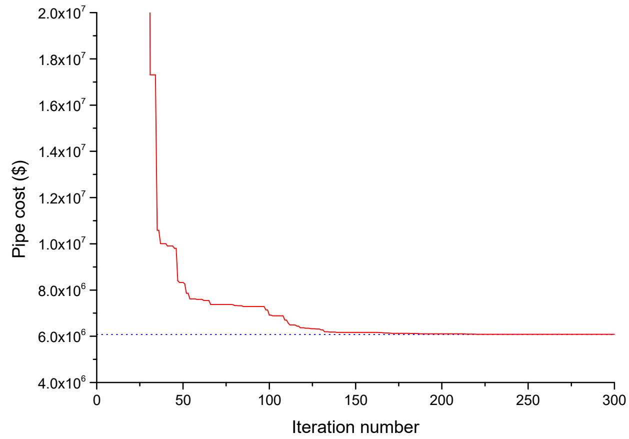


Fig. 9. Convergence of the Hanoi network pipe cost during optimization

distribution network and the best known global best cost is obtained. The proposed optimization interface contributes to the development of a platform for comparison and selection of algorithms for various water network design problems.

## ACKNOWLEDGMENT

The authors would like to thank The Council for Scientific and Industrial Research (CSIR), South Africa, for financial support.

## REFERENCES

[1] L. Rossman, *Epanet 2: users manual*. U.S. Environmental Protection Agency. Office of Research and Development. National Risk Management Research Laboratory, 2000.

[2] J. Van Zyl, J. Borthwick, and A. Hardy, "Ooten: An object-oriented programmers toolkit for epanet," in *Advances in Water Supply Management (CCWI03)*, 2003.

[3] D. Steffelbauer and D. Fuchs-Hanusch, "Oopnet: An object-oriented epanet in python," *Procedia Engineering*, vol. 119, pp. 710–718, 2015.

[4] K. Hindi and Y. Hamam, "Locating pressure control elements for leakage minimization in water supply networks: An optimization model," *Engineering optimization*, vol. 17, no. 4, pp. 281–291, 1991.

[5] S. Schal, L. S. Bryson, and L. E. Ormsbee, "A simplified procedure for sensor placement guidance for small utilities," *International Journal of Critical Infrastructures*, vol. 12, no. 3, pp. 195–212, 2016.

[6] P. R. Page, A. M. Abu-Mahfouz, and S. Yoyo, "Real-time adjustment of pressure to demand in water distribution systems: Parameter-less p-controller algorithm," *Procedia Engineering*, vol. 154, pp. 391–397, 2016.

[7] P. R. Page, A. Abu-Mahfouz, O. Piller, M. Mothetha, and M. Osman, "Robustness of parameter-less remote real-time pressure control in water distribution systems," in *SimHydro 2017: Choosing the right model in applied hydraulics*, June 2017.

[8] P. Page, A. Abu-Mahfouz, and M. Mothetha, "Pressure management of water distribution systems via the remote real-time control of variable speed pumps," *Journal of Water Resources Planning and Management*, vol. In Press, 2017.

[9] P. Page, A. Abu-Mahfouz, and S. Yoyo, "Parameter-less remote real-time control for the adjustment of pressure in water distribution systems," *Journal of Water Resources Planning and Management*, vol. In Press, 2017.

[10] T. J. Schriber, D. T. Brunner, and J. S. Smith, "Inside discrete-event simulation software: How it works and why it matters," in *2016 Winter Simulation Conference (WSC)*, Dec 2016, pp. 221–235.

[11] G. L. Curry, A. Banerjee, H. Moya, and H. L. Jones, "A modeling language generator for a discrete event simulation language in matlab," in *2016 Winter Simulation Conference (WSC)*, Dec 2016, pp. 1013–1023.

[12] The Mathworks Inc., *Simulink ® 7, Writing S-functions*, September 2007, R2007b.

[13] D. Elíades, "Epanet matlab toolkit," 2009. [Online]. Available: https://github.com/OpenWaterAnalytics/EPANET-Matlab-Toolkit

[14] The MathWorks Inc, *Simulink, User's Guide, Version 7*, 2009.

[15] J. Reca, J. Martínez, C. Gil, and R. Baños, "Application of several meta-heuristic techniques to the optimization of real looped water distribution networks," *Water Resources Management*, vol. 22, no. 10, pp. 1367–1379, 2008. [Online]. Available: http://dx.doi.org/10.1007/s11269-007-9230-8

[16] D. Mora-Melia, P. Iglesias-Rey, F. J. Martinez-Solano, and P. Ballesteros-Pérez, "Efficiency of evolutionary algorithms in water network pipe sizing," *Water Resources Management*, vol. 29, no. 13, pp. 4817–4831, 2015.

[17] O. Fujiwara and D. B. Khang, "A two-phase decomposition method for optimal design of looped water distribution networks," *Water resources research*, vol. 26, no. 4, pp. 539–549, 1990.

[18] Centre for water systems, University of Exeter, "Hanoi water distribution network." [Online]. Available: http://emps.exeter.ac.uk/engineering/research/cws/resources/benchmarks/layout/hanoi.php

[19] L. A. Rossman, "The epanet programmer's toolkit for analysis of water distribution systems," in *WRPMD'99: Preparing for the 21st Century*, 1999, pp. 1–10.

[20] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, nov/dec 1995, pp. 1942 –1948 vol.4.

[21] I.-Y. Chung, W. Liu, D. Cartes, E. Collins, and S.-I. Moon, "Control methods of inverter-interfaced distributed generators in a microgrid system," *Industry Applications, IEEE Transactions on*, vol. 46, no. 3, pp. 1078 –1088, may-june 2010.