Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 89 (2014) 13 – 20

Procedia Engineering

www.elsevier.com/locate/procedia

16th Conference on Water Distribution System Analysis, WDSA 2014

A Comparison of Population-based Optimization Techniques for Water Distribution System Expansion and Operation

M.S. Morley[a]*, C. Tricarico[b]

[a]Exeter Advanced Analytics LLP, 109 Pennsylvania Road, Exeter, Devon, EX4 6DT (United Kingdom)
[b]Dept. of Civil and Mechanical Engineering, University of Cassino and Southern Lazio, via Di Biasio 43, 03043 Cassino (Italy)

Abstract

This paper presents a water distribution system expansion and operation methodology employing population-based optimization algorithms applied to the Battle of Background Leakage Assessment for Water Networks competition. The problem is formulated as constrained single and multiple-objective optimization problems and implemented in a generic hydraulic optimization and benchmarking software application (Acquamark). To accelerate the evaluation of potential solutions, a distributed computing approach is employed, where necessary, to permit multiple solutions to be executed and evaluated in parallel. A pressure-driven demand extension to the EPANET hydraulic model is also employed to assist the optimization techniques in accurately ranking near-feasible solutions and to dynamically allocate leakage demand to the end nodes of each pipe.

Peer-review under responsibility of the Organizing Committee of WDSA 2014

Keywords: Multiple Objective Optimization; Evolution Algorithms; BBLAWN; Water Distribution Systems; Leakage

1. Introduction

Population-based optimization techniques have gained currency in recent years in their application to Water Distribution Systems (WDS) design and operation, with the emergence of genetic algorithms [4] and memetic algorithms such as the Shuffled Frog Leaping Algorithm [5] and Ant Colony Optimization [6]. This paper seeks to apply and compare a number of these algorithms to the Battle of Background Leakage Assessment for Water Networks (BBLAWN) challenge - part of the Water Distribution Systems Analysis (WDSA) conference, 2014.

* Corresponding author. Tel.: +44 7726006066
E-mail address: mark@morley.at

Peer-review under responsibility of the Organizing Committee of WDSA 2014
doi:10.1016/j.proeng.2014.11.154

2. Methodology

The optimization software developed closely couples a number of population-based optimization techniques implemented in C++ with the EPANET2 hydraulic solver [2] to model the effect on the performance of the hydraulic network when considering pipe replacement and duplication and the modification of pump and pressure reduction valve (PRV) operations.

2.1 Hydraulic Solver

The BBLAWN problem introduces a leakage model whereby leaks are calculated on a per-pipe basis and then aggregated into the demand nodes as per equations 1) and 2):

$$d_k^{leaks}\left(P_{k,mean}\right) = \begin{cases} \beta_k L_k P_{k,mean}^{\alpha_k} & P_{k,mean} > 0 \\ 0 & P_{k,mean} \leq 0 \end{cases} \tag{1}$$

Where: k = subscript of the k^{th} pipe; $P_{k,mean}$= model mean pressure along the k^{th} pipe in [m]; d_k^{leaks}= background leakage outflow along the k^{th} pipe in [m³/sec]; α_k & β_k = leakage parameters of the k^{th} pipe, dimensionless and [m$^{2-\alpha}$/sec]; L_k = length of the k^{th} pipe in [m];

$$\mathbf{d}_n^{leaks} = \frac{1}{2}\left|\mathbf{A}_{np}\right|\mathbf{d}_p^{leaks} = \frac{1}{2}\left|\mathbf{A}_{np}\right|\begin{bmatrix} d_1^{leaks} \\ \dots \\ d_k^{leaks} \\ \dots \\ d_{n_p}^{leaks} \end{bmatrix} \tag{2}$$

Where: $\boldsymbol{A}_{np}$= the network incidence matrix; $\boldsymbol{d}_n^{leaks}$= the aggregated leakage attributed to the n^{th} node in [m³/sec]

Since the leakage ascribed to a particular node is a function of the pressure both at itself and at the nodes at the end of each attached link, it is not possible to use the standard EPANET emitter component to model the leakage which operates on the basis of the available pressure at a single node. One approach would be to run the EPANET model normally and then adjust the demands to account for the leakage and to rerun the model repeatedly until convergence was reached. Whilst this has the advantage of not requiring any modifications to EPANET directly, it was discounted because of the extended run-times that such a strategy would necessarily entail.

Having successfully retrofitted a pressure-driven extension to EPANET previously [3] the authors have experience in adapting and extending the hydraulic solver and, accordingly, the leakage model described above has been incorporated directly into the C language source code of the EPANET toolkit. A number of functions have been modified (detailed in Table 1) to accommodate the leakage model as part of the normal iterative cycle employed by EPANET to produce the hydraulic solution. In addition, further variables were added to EPANET in order to store the leakage parameters *alpha* and *beta* for each link as well as the calculated leakage on a per-link and per-node basis. This approach has the advantage that by directly manipulating the solution matrices employed by EPANET, it is relatively straightforward to allocate leakage to tanks (as is required according to the rules). Ordinarily, EPANET does not allow the direct assignation of demands to tanks as would be necessary in this instance – requiring the introduction of additional dummy nodes and pipes in order to model this leakage correctly.

The use of EPANET with a stochastic optimization process commonly results in a large number of hydraulically-infeasible solutions being generated and subsequently evaluated by the hydraulic solver. The evaluation of these infeasible solutions takes additional time as, typically, the maximum number of solver iterations is expended attempting to converge the model and, additionally, large numbers of intermediate time steps may be introduced into the evaluation. Acquamark seeks to avoid the worst impacts of infeasible solutions by terminating their execution after the first time step in which they demonstrate hydraulic infeasibility. Instead of penalizing the solution heavily in order to hasten its departure from the population, the solution is marked as infeasible and estimates of its constraint violations are extrapolated, weighted by the proportion of the extended period simulation that had been successfully completed prior to the infeasibility.

This results in a commensurate reduction in the runtime "wasted" in evaluating infeasible solutions as well as preserving the genetic diversity of the population to the maximum extent possible.

Table 1. Overview of EPANET toolkit modifications to incorporate BBLAWN leakage model

File	Function	Modification description
epanet.c	*ENgetnodevalue*	Added routines to retrieve calculated per-node leakage for a given node.
	ENgetlinkvalue	Added routines to retrieve calculated per-link leakage, *alpha* and *beta* leakage terms for a given pipe.
	ENsetlinkvalue	Added routines to set *alpha* and *beta* leakage terms for a given pipe.
hydraul.c	*inithyd*	Initialize (zero) leakages for each node, leakages and average pressures for each pipe.
	newflows	Calculate magnitude of the leakage occurring in each on the basis of the nodal pressures at the end nodes – remembering that EPANET uses Imperial units internally and the BBLAWN model is metric.
		Traverse the adjacency list for each node to calculate the cumulative leakage ascribed to that node.
	nodecoeffs	Add leakage term to the demand flow subtracted for each node in the network.
input3.c	*pipedata*	Added routines to parse optional values specified for *alpha* and *beta* leakage terms for each pipe in the input file.

2.2 Objectives

The BBLAWN optimization has been formulated as a single or twin-objective optimization problem according to the needs of the optimization algorithms applied. In the case of the twin-objective formulation, the objectives are:

1. Total Cost – the sum of annualized infrastructure upgrade costs (pipe replacement and duplications, tank, pump and valve installation) and annual operational (pumping) costs.
2. Leakage – the absolute annual volume of water lost as leakage.

The single objective formulation combines the above objectives by assigning a cost to the annual leakage volume at a rate of €2/m^3.

2.3 Decision Variables

Table 2 enumerates the decision variable configuration employed for the optimization. In order to maximize the freedom afforded the optimization, no attempt has been made to simplify the problem by, and for example, grouping pipes. The potential sites for the 39 possible PRV installations were determined manually and, naturally, this will have biased the range of potential solutions accordingly.

Table 2. Decision Variable configuration

Decision Variable nos.	Description	Type	Range
1 – 432	Pipe replacement selection	Integer	12 pipe size options, plus “do nothing” and “closed”
433 – 864	Pipe duplication selection	Integer	12 pipe size options, plus “do not install”
865 – 872	Tank augmentation selection	Integer	6 additional volume options, plus “do not install”
873 – 884	Pump augmentation selection	Integer	4 new pump options, plus “do not install”
885 – 907	Control levels for existing pumps	Float (1dp)	Appropriate reservoir level range for each pump
908 – 930	Control levels for augmented pumps	Float (1dp)	Appropriate reservoir level range for each pump
931 – 932	Control levels for valve V2	Float (1dp)	
933 – 972	PRV installation selection	Boolean	
973 – 7,525	Hourly PRV settings	Float (1dp)	0.1m to 100.0m

2.4 Constraints

The optimization employs five “hard” constraints – violation of which result in a solution being marked as infeasible and, therefore, unlikely to play a significant role in the progress of the optimization. Firstly, the network produced must be hydraulically valid – that is to say that the EPANET solver solves the network without raising any errors. In addition, the solution of the network should not provoke any warnings to be emitted from EPANET. Of particular concern for the BBLAWN optimization are the warnings related to negative pressures, disconnected nodes and pumps operating beyond their normal flow regime. A minimum pressure constraint applies such that demand nodes must demand must satisfy a given pressure level (20m) for nodes with demand in order for a solution to be considered valid. In any event, there must be no negative pressures in the network at any point. Tanks are not permitted to empty, thus a constraint is also included to reflect this. To produce a solution that is repeatable over successive weeks, a further constraint is implemented such that the levels of any tanks in the system should be at least as high as they were at the beginning of the weekly extended period simulation.

Differential constraint weightings are used to signify the relative importance of meeting the optimization constraints. The EPANET Error and EPANET Warning constraints are given the highest priority in order to prioritise the generation of feasible solutions by the optimization.

2.5 Optimization techniques

The Acquamark environment decouples the implementation of the objective function for a problem from the operation of the algorithm. This makes it straightforward to implement and test the various algorithms without recourse to significant programming changes to accommodate the differing techniques. For example, the implementation of the objective function is able to adapt to being used with single and multiple-objective algorithms as well as discrete, continuous or mixed decision variable approaches.

A number of population-based optimization algorithms were evaluated for their suitability for application to the BBLAWN problem. Owing to the extended runtimes that were anticipated for the problem, it was decided to perform short tests on each algorithm to gauge its performance on the full problem. The procedures examined include a number of genetic and memetic algorithms as well as Parallel Differential Evolution [7] which differ markedly in their mechanisms for inheriting and sharing knowledge about the search space between members of their populations. The genetic algorithms employed were NSGA-II [8] and its closely related derivative, Omni-Optimizer (OO) [8]. The memetic algorithms used were a Discrete Shuffled Frog optimizer [10] based on the

Shuffled Frog Leaping algorithm [5] and a Discrete Particle Swarm Optimizer (DPSO) incorporating heterogeneous traits for individual particles [11].

The initial results for the memetic algorithms and Parallel Differential Evolution were disappointing. Whilst all of the techniques were able to resolve feasible solutions – and, indeed, more efficiently than the two genetic algorithm variants – none of the algorithms were able to significantly improve upon their early feasible solutions. It is unclear whether this is an issue relating to the scale of the problem encountered here or a short-coming in the authors' implementation of these algorithms. Certainly the DPSO has demonstrated itself on lower-dimensioned water distribution system problems without encountering such issues. Owing to time-constraints it was decided to postpone further evaluation of these techniques and to rely on the tested NSGA-II/OO algorithms at least until such time as a representative set of solutions had been derived in order to provide as baseline for further comparisons.

The application of OO to this problem did, however, highlight significant drawbacks to the technique which have not previously been encountered by the Authors. One of the principal differences between OO and NSGA-II is the former's incorporation of a crowding metric in decision space in addition to the metric in objective space common to the two algorithms. When applied to high numbers of decision variables, the statistics required by this additional metric entail considerable computational effort, particularly when calculating the Euclidean distance between each solution for each of the decision variables. A consequence of this was that over 50% of the runtime was spent in the computation of this metric. To minimize the effect of this problem, the statistical analysis was parallelized to run on all the available processor cores on the host machine executing the OO algorithm.

2.6 Inline heuristics

The BBLAWN problem introduces a pricing differential between the cost of replacing a pipe and that of duplicating it, resulting in a premium of 20% to be added to the cost of a duplicated, parallel pipe. In an optimization formulation, such as that outlined above, where the algorithm has complete freedom to independently select both replacement and duplication options, it is useful to ensure that the most cost-effective option is selected in each instance. To this end a number of heuristics were included in the genotype decoding step of the objective function. These include:

- If the decision variable for an existing pipe indicates that it should be closed *and* its corresponding duplicate pipe decision variable specifies the installation of a pipe, then the duplicate pipe diameter is chosen as a replacement pipe – given that this will necessarily be 20% cheaper.
- If a duplicate pipe is to be installed as well as a pipe replaced, if the duplicate pipe has a larger diameter than the replacement (and is therefore more expensive), the pipe diameters are reversed so that it is the cheaper pipe that attracts the 20% premium.
- If a duplicate pipe is to be installed and the original pipe *not* closed, a test is made to see if it is more cost-effective to install a single pipe with the same or greater cross-sectional area to the two pipes combined.

2.7 Post-processing heuristics

Following the completion of the evolutionary algorithm phase of the optimization, two heuristics are applied to the resulting solutions. Given the very high dimensionality of the optimization problem, as formulated above, these heuristics help to ensure that any feasible incremental improvements that are possible are implemented for a given solution.

The first heuristic operates by modifying the installed pipe diameters in a recursive fashion from the extremities of the network with a view to reducing cost at the expense of available pressure. This procedure works well for minimizing installation cost for purely dendritic networks. In the event of the recursion encountering a loop, each branch of the loop is evaluated separately in turn and the most cost-effective combination implemented.

Subsequently, a second iterative heuristic is applied to the network. This seeks to vary (normally downwards) the pressure settings of the PRVs at each time step in order to further reduce the available pressure in the network and, thus, to promote further reduction in leakage.

2.8 Optimization Environment

Given the extended runtimes that can be necessary with evolution algorithms, integrated into the software is the deEPANET [1] system for parallelizing the computation associated with hydraulic simulation. This software employs the industry standard MPI (Message Passing Interface) to implement a parallel-processing system which can distribute a pool of hydraulic networks awaiting simulation to local processors or remotely to computers on a LAN. Because of the relatively trivial data transfer speeds relative to computational effort required for an extended period hydraulic simulation, near linear improvements in GA runtime are achieved with the addition of processing cores. This, despite the unusually high number of decision variables that characterize this problem. For the purposes of this optimization the software was deployed across a cluster of three workstations, each equipped with two Intel Xeon E5645 CPUs packages which comprise six cores running at 2.4 GHz.

3. Issues

As with the Authors' entry for the Battle of the Water Networks – II [12], the variation in results between the single and double precision versions of EPANET remains an issue demonstrating differing results as the envelope of feasibility is explored. However, for the purposes of BBLAWN, this is no longer as critical as the solution does not need to be directly compared with the outputs of the standalone EPANET solver. The scale of the unconstrained problem as outlined above has introduced further challenges. For a population size of 2,500 individuals, the memory requirements for Omni-Optimizer, in particular, were very high requiring, at worst, ~6GB of RAM. This exceeds the single-process limit imposed by 32-bit Microsoft Windows of ~1.6GB. In order to run the full evaluation, therefore, it was necessary to move to a 64-bit implementation of the software. As with the single and double precision versions of EPANET, the 32-bit and 64-bit versions demonstrated appreciable differences in the results returned rendering interoperability between the versions unviable. It is considered that these variations, although numerically minor, occur due to the differing compilers and standard libraries employed by the two versions. For the purposes of the analysis herein, all results were evaluated using a 64-bit, double precision version of the EPANET solver.

The computation of pump energy consumption is somewhat problematic in EPANET. The result of retrieving the EN_ENERGY value for an individual pump returns an *instantaneous* value for energy consumption rather than one averaged over the reporting time step. As a consequence, it is more difficult to retrieve an accurate value for energy consumption in a network which has additional state changes necessitating the introduction of intermediate time steps. It is required, therefore, to calculate the energy consumption and, in the case of BBLAWN, leakage for each intermediate time step in order to get accurate values for both.

Subsequent to the optimizations being completed, it was discovered that in some instances, the evolution algorithms had opted to isolate some nodes without demands – in contravention of the rules. This transpired because EPANET does not regard isolation of non-demand nodes as a problem – although in the BBLAWN optimization, non-demand nodes are required to have non-zero pressure. Furthermore, it is not possible to use EPANET's built-in functions to verify disconnection in these instances. Instead an additional procedure had to be incorporated to verify each network nodes connectivity before starting the hydraulic simulation. Where affected, pipes were manually reinstated to meet the requirements of the competition. However, this reinstatement will have compromised the optimality of these solutions.

4. Discussion of Results

The optimal solutions produced through this methodology are largely characterized by replacement of most pipes in the network and the absence of any duplicated pipes. In part, this is due to the inline heuristic algorithm ensuring that duplicate pipes are employed to reinforce the network only where absolutely necessary – owing to the 20% cost penalty associated with such installations. More surprising is the absence of any supplementary tank storage. This characteristic was observed in the optimal solutions associated with all of the optimization techniques employed, being rapidly removed from feasible solutions towards the beginning of the optimization process.

Fig. 1 illustrates the Pareto-optimal fronts obtained using the NSGA-II and Omni-Optimizer (OO) algorithms - those allowed to run to completion. From right-to-left, these fronts represent, NSGA-II, OO and OO with the post-processing heuristics applied.

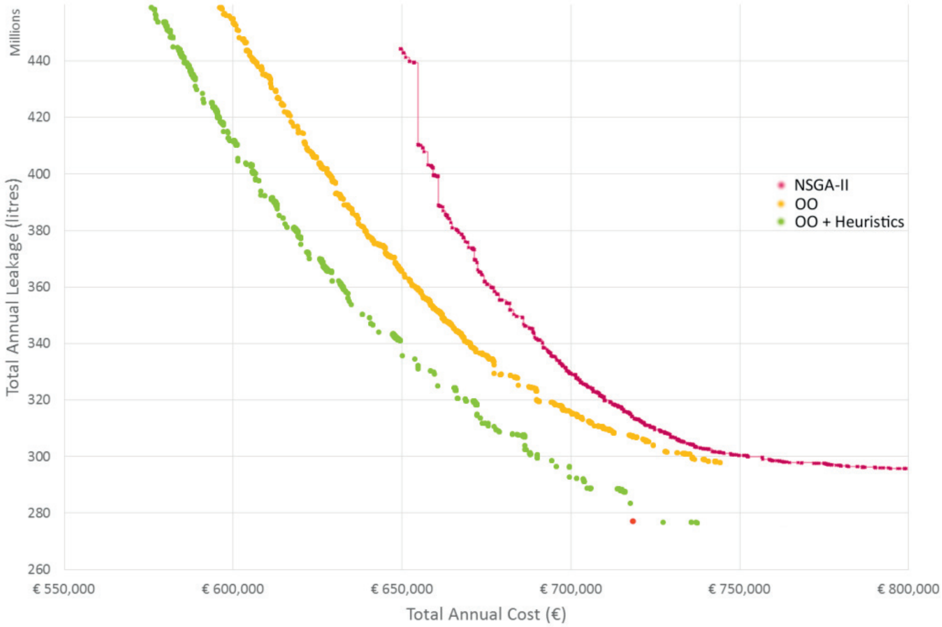

Fig. 1. Pareto-optimal results obtained with (right-to-left) NSGA-II, Omni-Optimizer and Omni-Optimizer + Heuristics

Table 3. Summary of selected optimal solution (all figures annualized)

Infrastructure:		
	Pipe replacement *(373 pipes replaced, 44 closed)*	€ 497,875.73
	Pipe duplication *(0 pipes)*	€ 0.00
	Tank augmentation *(0 tanks)*	€ 0.00
	Pump augmentation *(3 pumps)*	€ 11,491.00
	PRV installation *(12 PRVs)*	€ 2,144.00
	Sub-total	*€ 511,510.73*
Operation:		
	Total pump power consumption*	1,769,080 kWh
	Energy cost	*€ 205,860.95*
Leakage:		
	Total volume of lost water*	327,161.79 m^3
	Leakage cost (@ €2.00/m^3)	*€ 654,323.58*
Total Solution cost		**€ 1,371,695.26**

It should be noted that the calculations for both pump power consumption and leakage volume are undertaken for each intermediate time step rather than just the simulation time steps. When assuming the values at the beginning of each simulation time step are constant for the entire hour, these values are 1,762,109 kWh (better) and 327,172.30 m^3 (worse) respectively. As can be seen in Fig. 1, the results of NSGA-II were dominated by those obtained by OO, showing both superior absolute values and a better spread of solutions along the Pareto front. The improvement attributable to the heuristic routines when applied to the OO Pareto front is clear and it is from this resultant set that the final solution, highlighted in red, had been selected. This solution was later found to be infeasible, as discussed above, and required manual tweaking to restore their feasibility. A cost summary for the final, feasible, selected solution is presented in Table 3.

The selected solution, while feasible, is further characterized by a large number of intermediate timesteps, incorporated in the hydraulic solution by EPANET as a reflection of state changes in the network. In this instance, the selection by the optimizer of near-equal tank control levels for some of the pumps results in excessive switching of the pump states. This is an undesirable situation given the increased wear this will cause for the affected pumps.

5. Conclusions

An optimization methodology for the Battle of Background Leakage Assessment for Water Networks (BBLAWN) problem has been formulated and solved. The BBLAWN leakage model has been directly incorporated into the EPANET hydraulic solver to maximize the efficiency of the leakage evaluation. A BBLAWN-compatible version of the EPANET toolkit DLL will be available for download from http://www.acquamark.it.

A number of genetic and memetic algorithms were evaluated on short runs of the optimization and two, NSGA-II and Omni-Optimizer were allowed to run to completion on the full-scale optimization. The poor initial results achieved by the memetic algorithms are surprising given their general good performance relative to genetic algorithms and may represent difficulties in scaling for large numbers of decision variables or inadequacies in the Authors' implementation of these algorithms - exposed by the scale of the problem under consideration. As time constraints have precluded full evaluation runs for these algorithms being performed, it is proposed to evaluate these further in future as well as incorporating emerging techniques [13] with a track record in application to WDS optimization. Evaluation of the problem has been distributed on a local cluster computing resource using the deEPANET software for parallelizing the hydraulic simulations associated with each individual solution generated by the optimization.

6. References

[1] M.S. Morley, C. Tricarico, Z. Kapelan, D.A. Savić, G. de Marinis, "deEPANET: A Distributed Hydraulic Solver Architecture for Accelerating Optimization Applications Working With Conditions of Uncertainty," in Proceedings of the 7th International Conference on Hydroinformatics, Nice, France. 2006. Pp. 2465-2472.

[2] L.A. Rossman, EPANET 2 Users Manual, United States Environmental Protection Agency, Washington, USA, 2000, 200pp.

[3] M.S. Morley, C. Tricarico, "Pressure Driven Demand Extension for EPANET (EPANETpdd)," Technical Report 2008/02, Centre for Water Systems, University of Exeter, UK, 2008, 11pp. http://hdl.handle.net/10871/14721

[4] D.A. Savić, G.A. Walters, "Genetic algorithms for least-cost design of water distribution networks," Journal of Water Resources Planning and Management – ASCE, 1997, **123**(2), pp. 67-77.

[5] M.M. Eusuff, K.E. Lansey, "Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm," Journal of Water Resources Planning and Management, 2003, **129**(3), pp. 210-225.

[6] H.R. Maier, A.R. Simpson, A.C. Zecchin, W.K. Foong, K.Y. Phang, H.S. Seah, C.L. Chan Lim Tan, "Ant Colony Optimization for Design of Water Distribution Systems," Journal of Water Resources Planning and Management, 2003, **129**(3), pp. 200-209.

[7] D.K. Tasoulis, N.H. Pavlidis, V.P. Plagianakos, M.N. Vrahatis, "Parallel Differential Evolution," in Congress on Evolutionary Computation CEC2004, Portland, U.S.A., 2004, vol. 2, pp. 2023-2029.

[8] K. Deb, A. Pratap, S. Agarwal, T.A.M.T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, 2002, **6**(2), pp. 182-197.

[9] K. Deb, S. Tiwari, "Omni-Optimizer: A Generic Evolutionary Algorithm for Single and Multi-Objective Optimization", European Journal of Operational Research, **185**(3), pp. 1062-1087.

[10] M.T. Vakil Baghmisheh, K. Madani, A. Navarbaf, "A discrete shuffled frog optimization algorithm," Artificial Intelligence Review, 2011, **36**, pp. 267-284.

[11] E.C. Keedwell, M.S. Morley, D.P. Croft, "Continuous Trait-Based Particle Swarm Optimisation," 8th International Conference on Swarm Intelligence (ANTS2012), Brussels, Belgium, 2012, **7461**.

[12] M.S. Morley, C. Tricarico, G. de Marinis, "Multiple Objective Evolutionary Algorithm Appoach to Water Distribution System Model Design," Proceedings WDSA 2012 – 14th Water Distribution Systems Analysis Conference, Adelaide, Australia, 2012, pp. 551-559.

[13] E. Barlow, T.T. Tanyimboh, "Multiobjective Memetic Algorithm Applied to the Optimisation of Water Distribution Systems," Water Resources Management, 2014, DOI: 10.1007/s11269-014-0608-0.