COMPUTATIONALLY INTENSIVE DESIGN OF WATER DISTRIBUTION SYSTEMS

by

Manuel Alejandro Andrade-Rodriguez

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF AGRICULTURAL AND BIOSYSTEMS ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2013
As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Manuel Alejandro Andrade-Rodriguez, titled Computationally Intensive Design of Water Distribution Systems, and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

_______________________________________________ Date: 8/7/13
Dr. Christopher Y. Choi

_______________________________________________ Date: 8/7/13
Dr. Murat Kacira

_______________________________________________ Date: 8/7/13
Dr. Lingling An

Final approval and acceptance of this dissertation is contingent upon the candidate’s submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

_______________________________________________ Date: 8/7/13
Dissertation Director: Dr. Christopher Y. Choi
STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Manuel Alejandro Andrade-Rodriguez
ACKNOWLEDGEMENTS

Many thanks to my advisor, Dr. Christopher Choi, for showing me that, although you will never be perfect, you should still try to achieve perfection everyday.

I would like to express my gratitude to my colleagues in the Heat and Mass Transfer Lab at The University of Arizona. In particular, thanks to Fernando Rojano, Mario Mondaca, Jessica Drewry, and Yifan Liang, for all of their support to this work.

Thanks to my committee members, Dr. Murat Kacira and Dr. Lingling An, for their valuable suggestions.

Thanks to Dr. Kevin Lansey, and all the members of the Emerging Frontiers in Research and Innovation (EFRI) group at The University of Arizona, for their several contributions during all stages of this work.

This work was supported by the National Science Foundation (under Grant No. 083590).
DEDICATION

A mi esposa, Janeth, porque has confiado en mí, y porque me has apoyado para alcanzar mis metas, sin importar que éstas me llevaran lejos de ti. Te amo.

A mi hija, Ximena Sarai, porque contigo mi corazón vive una inmensa alegría que nunca antes experimentó.

A mis padres, Raúl y Carmela, y mis hermanos, Ulises, Santos, y Ayrton, porque siempre los llevo conmigo.

A los que vienen, porque sus sonrisas serán mi mayor recompensa.
TABLE OF CONTENTS

ABSTRACT .................................................................................................................. 7

1. INTRODUCTION .................................................................................................. 9
   1.1 The single-objective pipe-sizing problem .................................................. 10
   1.2 State of the art ................................................................................................ 12
   1.3 Research objectives ...................................................................................... 18

2. PRESENT STUDY .................................................................................................. 20
   2.1 Overall summary ........................................................................................... 20
   2.2 Overall conclusions and recommendations ................................................ 25

REFERENCES ........................................................................................................... 31

APPENDIX A: HEURISTIC POST-OPTIMIZATION APPROACHES FOR DESIGN
OF WATER DISTRIBUTION SYSTEMS ....................................................................... 34

APPENDIX B: ENHANCED ARTIFICIAL NEURAL NETWORKS ASSISTING THE
OPTIMAL DESIGN OF WATER DISTRIBUTION SYSTEMS ...................................... 64
ABSTRACT

The burdensome capital cost of urban water distribution systems demands the use of efficient optimization methods capable of finding a relatively inexpensive design that guarantees a minimum functionality under all conditions of operation. The combinatorial and nonlinear nature of the optimization problem involved accepts no definitive method of solution. Adaptive search methods are well fitted for this type of problem (to which more formal methods cannot be applied), but their computational requirements demand the development and implementation of additional heuristics to find a satisfactory solution. This work seeks to employ adaptive search methods to enhance the search process used to find the optimal design of any water distribution system. A first study presented here introduces post-optimization heuristics that analyze the best design obtained by a genetic algorithm—arguably the most popular adaptive search method—and perform an ordered local search to maximize further cost savings. When used to analyze the best design found by a genetic algorithm, the proposed post-optimization heuristics method successfully achieved additional cost savings that the genetic algorithm failed to detect after an exhaustive search. The second study herein explores various ways to improve artificial neural networks employed as fast estimators of computationally intensive constraints. The study presents a new methodology for generating any large set of water supply networks to be used for the training of artificial neural networks. This dataset incorporates several distribution networks in the vicinity of the search space in which the genetic algorithm is expected to focus its search. The incorporation of these networks improved the accuracy of artificial neural networks trained with such a dataset.
These neural networks consistently showed a lower margin of error than their counterparts trained with conventional training datasets populated by randomly generated distribution networks.

**Keywords**: water distribution system, optimization, pipe sizing, genetic algorithm, artificial neural network, post-optimization.
1. INTRODUCTION

Most of the cost of any municipal water distribution system derives from the pipes and fittings in the network used to carry water to all parts of the city. The main transmission and secondary distribution components of the system can account for as much as 60% of the infrastructure investments required over a 20-year span in order to guarantee the public health of most communities in the United States (EPA 2009). Achieving an inexpensive design for a Water Distribution System (WDS) is thus a major concern for any typical municipality committed to supplying enough water, of a satisfactory quality to all its inhabitants. When the construction cost is the main concern, finding the optimal design depends on selecting as design constraints those characteristics of the system that will achieve the best design the lowest expense.

The simplest—and also the most often studied—problem associated with the optimal design of a WDS involves selecting the dimensions of the pipes to be used. These must be chosen from a discrete set of commercially produced sizes and done in such a way that guarantees (i) water can be provided to all points of consumption at a minimum required pressure, even under extreme conditions of demand, and that (ii) the selected diameters produce the cheapest distribution network capable of satisfying such pressure constraints. Despite the conceptual simplicity of the pipe-sizing problem, the use of discrete diameters sizes (together with the nonlinearity of the physical equations ruling the movement of water within pipes) yields a complex problem with no conclusive algorithm of solution.
1.1 The single-objective pipe-sizing problem

Regardless of whether a new Water Distribution System (WDS) is to be built, or some deteriorated pipes replaced, the designer wants to find an inexpensive solution that will at the same time guarantee an achievement of all constraints imposed by a set of drinking water regulations. A mathematical formulation of the optimization problem involved, where the only objective is to minimize the construction cost of the pipes to be installed and where pressure is the only operational constraint, can be stated by the following equation (where Eq. (2) restricts every diameter $d_k$ to be selected from a discrete set of commercial sizes; Eq. (3) requires that the pressure at all nodes must be above a limit; and Equations (4) and (5) account for the conservation of energy and flow, respectively):

Minimize $f(D) = \sum_{k=1}^{M} (C_k \cdot L_k)$

Subject to: $d_k \in S_d$

$P_j \geq P_{\text{min}}, \ j \in \{1, ..., J\}$

$\sum_{k=1}^{K} H f_k = \sum_{k=1}^{K} f_k L_k \frac{v_k^2}{d_k^2} = 0$

$\sum Q_k^j - \sum Q_k^{j'} = 0$

where $f$ is the objective function to be minimized (represented as a function of the vector of diameters $D=\{d_1, d_2, ..., d_M\}$ proposed for the $M$ pipes in the water supply network); $C_k$ is the cost per unit length of pipe $k$; $L_k$ is the length of pipe $k$; $d_k$ is the diameter of pipe $k$;
$S_d$ is the set of commercial pipe sizes from which each diameter $d_k$ must be selected; $P_j$ is the pressure at the point of consumption (node) $j$ in the network; $J$ is the total number of nodes in the network; $P_{\text{min}}$ is the minimum acceptable pressure determined by drinking water regulations; $Hf_k$ is the pressure loss (due to friction between the wall and the flow) occurring along pipe $k$; $K$ is a set of pipes forming a loop; $f_k$ is a dimensionless friction factor; $v_k$ is the average velocity of water moving along pipe $k$; $g$ is the gravitational acceleration; $Q^j_k$ is the flow (moving towards node $j$) of pipe $k$; $Q'^j_k$ is the flow (leaving node $j$) of pipe $k$; and $q$ is the flow delivered to customers through node $j$.

The use of commercial pipe sizes—a constraint in Eq. (2)—gives the problem a combinatorial nature and for which exhaustive search is the only known definitive solution. However, the practical application of optimization by simple enumeration is limited to unrealistic systems (meaning those consisting of only a few pipes), due to the immense amount of possible network configurations that should be tested for a system made up of even a few dozen pipes. The designer of a realistic water supply network should therefore rely on heuristic methods that are well fitted for these types of combinatorial problems, where more precise optimization methods cannot be applied.

Such heuristic methods, also known as adaptive search methods (ASMs), are inspired by natural processes and hence apply experience (and common sense) to perform an ordered exploration of the total search space, in the hope of finding the global optimum solution to the pipe-sizing problem posed by equations (1) to (5). Given that the effectiveness of ASMs is highly dependent of an exhaustive search, parallel heuristics that can reduce their computational demands should be highly appealing to designers of WDS.
1.2 State of the art

*Optimization methods applied to the pipe-sizing problem*

According with Mays (2000), “virtually every optimization method has been applied to the problem of water distribution optimization.” Linear programming has been used to solve a modified (and unrealistic) version of the problem where each pipe is divided in multiple segments. Each segment is assigned one of the commercially available diameter sizes considered for the problem. This simplification linearizes the problem because equation (4) is linear with respect to the modified decision variable that is now the length of each segment in a pipe (Alperovits and Shamir 1977; Fujiwara et al. 1987).

Applications of nonlinear programming (NLP) (Shamir 1974; Lansey and Mays 1989) are also limited because they cannot guarantee the achievement of a global optimum when a problem, such as the one presented here, has multiple local optimum solutions. Furthermore, NLP formulations require the use of continuous diameters, a restriction that must be overcome by making the erroneous assumption that if each optimal continuous diameter—obtained with NLP—is rounded to its immediate superior pipe size in a discrete set of commercial diameters, the solution to the original problem remains optimal.

Yates et al. (1984) revealed the true complexity of the pipe-sizing problem: developing an algorithm that can select discrete diameters on the basis of minimizing the cost of a water supply network poses a problem of a mathematical class known as ‘non-deterministic polynomial-time hard’ (NP-hard). When the computational complexity of a problem is classified as NP-hard, two immediate conclusions can be derived: (i) using a
rigorous algorithm to find an optimal solution to the problem is not a practical possibility because the computational time required by such an algorithm would be, at best, an exponential function of the number of decision variables; and (ii) the search for a cheap WDS with an efficient performance must be focused on the development of approximation methods capable of finding a solution that is reasonably close to being optimal.

Adaptive Search Methods (ASMs) have emerged as an effective alternative way to solve the pipe-sizing problem. These methods are important to the search for an optimal water distribution network design because there is a dearth of robust and rigorous methods that can provide a satisfactory solution to the problem (Walski et al. 2003). To search for an optimal solution, an ASM relies on computational algorithms drawn from natural processes. According with Hillier and Lieberman (2010): “An adaptive search method is a general kind of solution method that orchestrates the interaction between local improvement procedures and higher level strategies to create a process that is capable of escaping from local optima and performing a robust search of a feasible region.” Adaptive search methods can easily handle discrete decision variables because they base their searches solely on an evaluation of the objective function, as opposed to the linear and nonlinear programming methods that use gradient information to achieve the same ends (Mays 2000).

The genetic algorithm (GA) is one of the most popular ASMs; it has been extensively used to find the optimal design of a WDS since Simpson et al. (1994) first applied it to the problem. A GA involves a methodology inspired by such processes as the natural
selection and mutation of genes (drawn from evolutionary theory), in the hope of obtaining a solution that is, iteration after iteration, ever closer to the global optimum. The algorithm starts with a set of possible solutions—i.e., a set of vectors of diameters $D$ to be applied to the objective function, Eq. (1)—named the initial population. The cost of each member of the initial population is calculated with Eq. (1), and a hydraulic model implementing equations (4) and (5) is run to determine the feasibility of each member of the initial population. If a member of the population fails to accomplish Eq. (3), i.e. is unfeasible, a large cost penalty will be added to the cost estimated with Eq. (1). Such a penalty has the objective of preventing unfeasible solutions from being considered as the best members of the population. The GA then performs an ascending sorting of the population based on the total cost of each member (determined by the sum of its actual cost and any applied penalty). A portion of the members having the lowest total cost is then selected to create the second generation of possible solutions, one that possesses a combination of characteristics ‘inherited from their parents.’ By passing on the characteristics of the best members of a population to the following generation, the algorithm assumes that new members will be better suited to solving the optimization problem (or, to draw on the analogy to natural selection, better fitted to their environment). The creation of a new generation also involves a random component with the objective of preventing the algorithm from being trapped in a local minimum. The introduction of random variations to a small portion of the population is called ‘a mutation.’ Although most mutations are expected to be detrimental, a few may introduce beneficial changes to the elements of the next generation. Besides creating the second
generation from a portion of the elements of the initial population, the algorithm will pass a smaller portion of the best elements to the second generation without changes. These unmodified members are known as the ‘elite’ of the population, and the algorithm retains them without changes with the objective of keeping the best overall solution until the end of its search.

In summary, a generic GA performs two basic steps: it sorts a population based on the total cost of each of its members, and it generates a new population based on the best members of the previous. Given that the members of the new population are expected to be closer to the global optimum than the members of the previous generation, the best overall member tested by the GA is expected to be a good approximation to the global optimum. Unlike nonlinear programming, where the search for the global optimum requires the use of gradients, the search performed by the GA is only dependent on the value of the objective function. This results in a major limitation: a GA needs to analyze a large number of generations that will slowly converge towards optimality as the search progresses. For a GA, as well as for other adaptive search methods (ASMs), the larger the number of possible solutions tested, the better the final solution is expected to be. A more detailed description of the internal functioning of a GA can be found in Hillier and Lieberman (2010) and Savic and Walters (1997). Other types of ASMs that have been applied to the design of water supply networks include ant colony optimization (Maier et al. 2003), simulated annealing (Cunha and Sousa 1999), harmony search (Geem 2006), and shuffled frog-leaping algorithm (Eusuff and Lansey 2003).
Computationally demanding constraints

The exhaustive search required by GAs—and ASMs in general—severely limits the type of constraints that can be considered when solving the pipe-sizing problem. If, for example, a disinfectant is injected into the system, and a minimum concentration $C_{\text{min}}$ must be maintained to prevent bacterial re-growth, the following condition should be included in addition to equations (1) to (5):

$$C_j \geq C_{\text{min}}, \ j \in \{1, \ldots, J\}$$

(6)

where $C_j$ is the concentration at node $j$, and $J$ is the number of nodes in the WDS to be optimized.

In order to determine the disinfectant concentration at any node $C_j$, at a time $t$, it is customary to solve the following one-dimensional transport equation for each pipe in the water supply network:

$$\frac{\partial C}{\partial t} = -u \frac{\partial C}{\partial x} + rC^n$$

(7)

where $C(x,t)$ is the concentration at a position $x$ in the pipe, at a time $t$; $u$ is the average velocity of water traveling through the pipe; $r$ is a reaction rate coefficient; and $n$ is the order of the reaction. Equation (7) determines the change in the concentration $C$, with respect to time $t$, as a function of advection (first term in the right hand side) and the reaction (second term) happening due to the contact between the constituent and the pipe wall and/or the reaction occurring within the bulk flow.

An analytical solution to a system of Equation (7) is only practical for unrealistic WDSs consisting of a few pipes (Axworthy and Karney 1996). Using a water quality
model on larger WDSs requires numerical solutions. Epanet (Rossman 2000), the software most widely used to model water supply networks, incorporates a Lagrangian method to solve Eq. (7). This method assumes that a constituent flowing through a pipe travels in a series of segments with uniform concentration but varying volume. These segments advance through the pipe in small time steps to account for the effects of advection. The effects of reaction are also considered by changing the segments’ concentration to reflect the chemical changes within the segment.

When searching for the optimal design of a WDS, testing pressure constraints is usually achieved by performing a few hydraulic simulations of a network under extreme conditions of demand (such as a peak demand or a fire flow). This process usually requires fractions of a second and hence can be included in the pipe-sizing problem with no difficulty. Testing water quality constrains, on the other hand, requires the application of a numerical solution, like the aforementioned Lagrangian method, which can take several seconds or several minutes (for WDSs with thousands of pipes) to be completed. Hence, the incorporation of water quality constraints constitutes a bottleneck in any adaptive search method (ASM) that requires a long search to obtain a good solution.

Broad et al. (2005) proposed the use of an artificial neural network (ANN) to replace the water quality model incorporated by Epanet. Put simply, an ANN, is a computational device inspired by the way that a human brain processes information to acquire knowledge. ANNs are considered ‘universal approximators’, i.e., they are capable of approximating any measurable function to any desired degree of accuracy (Hornik et al. 1989). ANNs are capable of handling the complex associations that can exist between
inputs and outputs, and they are used extensively in problems involving function fitting, pattern recognition, the classification of data, and time series analysis. In water supply engineering, ANNs have been used mainly to classify and function fit field data (Bowden et al. 2006; Polycarpou et al. 2002; May et al. 2008; Dandy et al. 2005). However, ANNs can also serve as a substitute for deterministic models in tasks with high computational demands, such as optimization routines (Lingireddy et al. 2005).

1.3 Research objectives

The optimization problem posed by Equations (1) to (5) requires the application of advanced mathematical techniques. Genetic algorithms and artificial neural networks are two of the techniques that designers of WDS have been using to provide a satisfactory solution to the problem. Both mathematical tools were inspired by natural processes and, as with any biological system, undergo constant change in an effort to improve the quality of their outcomes.

This work seeks to enhance the process that designers use when searching for an optimal design for an urban water distribution system. This objective is achieved by means of two studies that both deal with specific features of genetic algorithms and artificial neural networks:

- A first study (Appendix A) proposes a methodology that will further improve the design of a water supply network obtained using a genetic algorithm in a post-optimization stage.
A second study (Appendix B) performs a comparative analysis of the factors that impact the performance of artificial neural networks when applied to the optimal design of a WDS.
2. PRESENT STUDY

2.1 Overall summary

This work deals with the process of selecting the optimal pipe diameters for a water supply network. Given the demanding computational requirements of adaptive search methods (ASMs), and genetic algorithms (GAs) in particular, this work explores two approaches that improve the performance of a GA searching for an optimal solution to the pipe-sizing problem. A first study (Appendix A) presents heuristics methods for detecting potential cost savings that a GA may have failed to recognize, even after an exhaustive search. A second study (Appendix B) presents a methodology for enhancing the accuracy of artificial neural networks (ANNs) applied as faster substitutes for the demanding water quality model used to analyze water supply networks.

Post-optimization heuristics applied to the optimal design of a water supply network

The first study (Appendix A) challenges the common assumption that the design of a WDS obtained using a GA cannot be further improved after the GA has finished its search. A proposed methodology performs a local search for a network design generated with a GA. The local search looks for pipes that the GA failed to designate as candidates for an additional reduction in diameter that, despite the pressure loss caused by the reduction, can still maintain the feasibility of the proposed solution. A local search can be achieved by changing the dimensions of the pipes’ diameters (either by increasing or decreasing them). However, the proposed methodology follows the assumption that further savings can be only achieved by reducing diameters. This presumption gives the
local search the well-defined purpose of finding a set of pipes that, when their diameters are reduced, will yield a feasible WDS at the lowest cost.

If a set $S$ contains all the diameter reductions that can be performed without neglecting pressure constraints, a convenient way to bring the solution obtained by the GA closer to the global optimum is to find a subset of $S$ (named $S'$) that will meet pressure constraints and maximize cost savings. Similar to the original optimization problem, finding the optimal subset $S'$ is a combinatorial problem where finding a solution by enumeration is prohibitive. Although it is evident that applying a second ASM could find such an optimal subset, the proposed method could yield a solution more quickly and efficiently and avoid the application of an additional, computationally intensive, optimization method.

The proposed method performs a pipe modification at each iteration based on a ranking. The highest-ranked pipe reduction is considered part of the final design, and the set $S$ is updated to reflect the resulting pressure loss. The iterative process of ranking pipes, then reducing the diameter of the top-ranked pipe, and finally updating set $S$ is repeated until $S$ becomes an empty set. To rank the pipes reductions, two algorithms were proposed.

The first algorithm ranks reductions using a decision rule expressed as a binomial. The first term of the decision rule accounts for the cost savings achieved by performing the reduction. Similarly, the second term accounts for the pressure loss that will occur due to the same reduction. The decision rule, so defined, assumes that the best pipe to be reduced will be the one providing the highest sum of both terms, i.e., the pipe that offers
the best balance between cost savings and pressure losses. This algorithm is computationally efficient; it can be executed in a few seconds.

In order to prioritize speed, the first algorithm is forced to make a decision without considering its impact on the final design. In other words, the first algorithm assumes that the reduction of a diameter at an initial iteration will contribute to finding the optimal subset $S'$. Appendix A shows that this only happens when considering a proper balance between the pressure and the cost savings terms in the decision rule. The second algorithm puts more emphasis on the appropriate selection of reductions. It takes advantage of the first algorithm’s speed by using it as a subroutine. The second algorithm ranks reductions based on the cost savings that can be achieved after downsizing a pipe. Such cost savings are estimated using the first algorithm, and they represent the maximum future benefits that can be expected if a pipe is selected for downsizing. The highest-ranked pipe then becomes the one offering the highest potential savings. This algorithm sacrifices computational speed for the sake of a robust search of the optimal subset $S'$.

*Artificial Neural Networks applied to the optimal design of a water supply network*

The second study (Appendix B) seeks to improve the performance of the artificial neural networks (ANNs) that assist in finding the optimal design of a WDS. The assumption (Broad et al. 2005; Odan and Ribeiro 2012) that ANNs should be trained with a random dataset was evaluated by comparing it against an alternative approach involving a probabilistic dataset. The comparison has the main objective of determining if a higher accuracy can be expected from ANNs trained with the latter dataset.
The first two networks in a random dataset are the two extreme cases in the search space: all of the diameters in one network are set to the minimum size, while all the diameters in the second set are set to the maximum size. All other networks in the dataset are assigned random diameters. By including extreme and random cases, it is presumed that networks in a random dataset will be distributed over the entire search space. This assumption is based on the expectation that ANNs trained with such a dataset will always behave as interpolators and hence avoid the poor performance of ANNs acting as extrapolators (Flood and Kartman 1994). Although the two extreme networks will certainly prevent extrapolation, it is unlikely that the remaining random networks will be properly distributed throughout the search space. Furthermore, the overwhelming majority of networks in the dataset will be largely infeasible because the probability is remote that a sufficiently large diameter will be assigned to most main pipes.

The second study (Appendix B) presents an alternative methodology for generating networks in the training dataset. To avoid extrapolation, two extreme networks are used for the first two elements. All other networks in the dataset, however, are assigned diameters by using a probabilistic approach. This approach has the objective of creating a large number of feasible and almost-feasible networks—defined here as networks that require only a few modifications to achieve feasibility—that better resemble the types of networks analyzed by a GA after the initial iterations. The probabilistic approach creates such networks by using a feasible ‘reference network,’ located in the vicinity of almost-feasible networks, in such a way that even a minor modification to the network will
change its feasibility status. This ‘reference network’ serves to indicate the hypothetical boundary dividing the search space into feasible and unfeasible elements.

Assuming that changes to diameters in the ‘reference network’ are expected to yield a new feasible or almost-feasible network, the proposed approach generates a large set of these networks by making systematic modifications to copies of the ‘reference network.’ These modifications are achieved by means of a discrete probability distribution that determines which pipes in a copy of the ‘reference network’ will be modified, and which diameter will be assigned to them. Such a distribution assigns smaller probabilities to drastic changes in a diameter size, and higher probabilities to mild changes. As an example, consider a set of commercial sizes $S_d = \{6 \text{ in (152.4 mm)}, 8 \text{ in (203.2 mm)}, \text{ and 10 in (254 mm)} \}$. If a given pipe in the ‘reference network’ has a diameter of 8 in, a valid discrete distribution for that pipe would be $(1/4, 1/2, 1/4)$—as opposed to the $(1/3, 1/3, 1/3)$ distribution applied for a random assignment of diameters. In such a case, a diameter of 6 in will be assigned to that pipe for approximately 25% of the networks in the probabilistic dataset, a diameter of 8 in for 50% of the networks, and a diameter of 10 in for the remaining 25%.

The proposed methodology requires the finding of a good ‘reference network,’ one that is not only similar to almost-feasible networks, but also similar to the unknown networks that a GA will analyze during its search for a solution to the pipe-sizing problem. The global optimum is a perfect example of an appropriate ‘reference network’: it is the cheapest feasible network in the search space, it is the solution that the GA is trying to find, and even the smallest modification can make the network unfeasible.
However, the global optimum is unknown and a feasible network with a slightly higher cost can be used instead. The approximation to the global optimum should be obtained with little computational expense to avoid a significant extension of the optimization process. Appendix B presents a methodology to obtain a fast approximation to the solution of the pipe-sizing problem.

The methodology used to obtain the ‘reference network’ has been adapted from the first algorithm in Appendix A. Instead of departing from the solution obtained by a GA, the algorithm starts with a network wherein all pipes are assigned the largest diameters in the set of commercial sizes. The algorithm performs the reduction of a particular diameter per iteration. The pipe to be downsized is the one providing the best balance between cost savings and the loss of constraints’ redundancies. The algorithm stops when the modification of any pipe yields an unfeasible solution to the pipe-sizing problem. The resulting network is then designated as the ‘reference network,’ and the probabilistic approach will use it to generate a large dataset of feasible and almost-feasible networks. The resulting training dataset is then used to train probabilistic-ANNs, the performance of which will be compared with the performance of random-ANNs.

2.2 Overall conclusions and recommendations

This work presents two methods for improving the search for an optimal solution to the pipe-sizing problem. The fist approach (Appendix A) introduces the idea of post-optimization heuristics. These can be regarded as computationally inexpensive heuristic methods that analyze a solution found by more demanding heuristics (such as an adaptive search method) in the hope of achieving a better solution to the optimization problem. A
'better' solution is a concept that must be adapted to match the specific requirements imposed by the designer of a water supply network; some designers may want to reduce the capital cost of the designed system, while others wish to reduce the operational cost and still others seek to minimize the cost of potential future expansions of the system. In this work, the modification of a design is considered to be an improvement if it reduces the capital cost of the design without affecting the constraints imposed by the optimization problem.

To maximize the savings obtained by the downsizing of pipes, two post-optimization algorithms were proposed. Both assume that further improvement in the design of a water supply network can only be achieved by a reduction of pipes’ diameters. A real WDS served as case study to demonstrate the maximum benefits (savings) that can be expected with regard to either algorithm. The WDS consisted of 1274 pipes, corresponding to the total number of decision variables. In a preceding study by Kang and Lansey (2011), a GA solving Equations (1) to (5) for this WDS converged on the same near-optimal design in nine out of ten independent runs. Each GA run evaluated 250,000 networks (500 generations with a population of 500 individuals) in order to achieve such a near-optimal design.

The two post-optimization algorithms were applied to bring this design closer to the global optimum. The results of both algorithms are dependent of two parameters: a pressure multiplier $m_P$ and a savings multiplier $m_C$. These multipliers are required in order to normalize the decision value used to rank all the potential pipes’ reductions, in such a way that the best reduction will be assigned a decision value of one and the worst
reduction a value of zero. Hence, the sum of both multipliers should be always one. Both algorithms were executed eleven times, with increments of 0.1 in the savings multiplier that was changed from zero to one, and the corresponding decrements of 0.1 for the savings multiplier that was accordingly changed from one to zero. Both algorithms achieved the maximum post-optimization savings of $56,592. The first algorithm proved to be fast because it required a maximum of 42 seconds for its execution, although it only found the maximum savings for two out of the eleven runs (when the savings multiplier is 0.6 and 0.7 and the pressure multiplier is 0.4 and 0.3). The second algorithm proved to be less dependent on the savings and pressure multipliers (it found the maximum savings for nine out of eleven runs). The second algorithm required a maximum of 44 minutes to find the maximum savings and hence sacrifices speed for the sake of robustness.

The two post-optimization algorithms proposed in this work were applied to the single-objective optimization problem posed by Equations (1) to (5). However, further research is recommended in order to adapt these algorithms to a multi-objective problem. Similarly, the main objective of these algorithms is to maximize cost savings; to achieve other benefits, we recommend similar approaches, such as maximizing the reliability of the system under a certain budget.

The second approach (Appendix B), proposed as a way to enhance the search for an optimal WDS, tackles the computational expense required by demanding constraints. The addition of water quality constraints to the pipe-sizing problem (Equations (1) to (5)) requires numerous numerical solutions of the advection-reaction equation in (7). Repeated execution turns the water quality model into a bottleneck, slowing down the
optimization problem. Artificial neural networks (ANNs) have proven to be comparably fast estimators of disinfectant concentrations throughout a water supply network (Broad et al. 2005). However, the success of an ANN is highly dependent of the dataset used for its training. A random dataset may be appropriate for estimating the behavior of networks tested during the initial iterations of a GA, but as the number of iterations advances, a GA will focus its search in the neighborhoods of the search space where the optimal solution is expected to be. As it is unlikely that a network with random diameters will be part of such neighborhoods, an alternative methodology is needed to find networks that will better resemble the networks tested by a GA after the initial iterations.

This work presents such a methodology. A fast approximation to the optimal solution of the optimization problem posed by Equations (1) to (6) is required as a ‘point of reference’ that will indicate one of the multiple boundaries of a hypothetical feasibility zone within the search space. This network, known as the ‘reference network,’ serves as a convenient indicator of the area where the GA is expected to focus its search. A non-uniform discrete probability distribution is repeatedly applied to duplicates of the ‘reference network’ to generate the training dataset. The pipes’ diameters in a duplicate of the ‘reference network’ will be either changed or left unmodified according with such a distribution. This discrete distribution assigns a higher probability to a minor change of diameter (or no change at all) and a smaller probability to a drastic change. The networks resulting from these changes are expected to be semi-random networks surrounding the aforementioned hypothetical feasibility zone, and hence are also expected to be more representative of the search space scanned by a GA.
The proposed methodology was applied to a water supply network consisting of 34 pipes. The first algorithm presented in Appendix A was used as an optimization method in dealing with a pipe-sizing problem that included water quality constraints. The algorithm required 311 seconds to find a feasible network that cost only 2.56% above the cost of the best solution found by a GA. Ten independent runs of the GA were executed. A GA run required an average of 52,812 seconds to evaluate $10^5$ networks (200 generations with a population of 500 each), and the water quality solver assisting the GA required 96% of this time. The set of $10^6$ networks evaluated by the ten GA runs served as a validation dataset to compare the accuracy of those ANNs trained with a random dataset (random-ANNs) to the accuracy of ANNs trained with a probabilistic dataset (probabilistic-ANNs). The two types of ANNs required an average of 923 seconds to estimate the minimum disinfectant concentration for all the networks tested by a GA run, i.e., the ANNs in this study required around 1.8% of the computational time demanded by a water quality model. In terms of accuracy, the probabilistic-ANNs show a better accuracy than the random-ANNs for five out of the six comparisons performed, thus demonstrating the probabilistic approach presented here is advantageous when searching for the optimal design of a water supply network.

The case study considered in Appendix B consists of a few dozens pipes. To validate the findings presented here, we recommend further involving a realistic WDS (one consisting of thousands of pipes). Similarly, this study is focused on determining the impact that the selection of networks in a training dataset has on the final performance of ANNs. Similar approaches should be considered to determine the impact of other factors.
For example, the ANNs in this study were trained to forecast the magnitudes of disinfectant concentrations at important nodes—known as critical nodes—where the minimum concentration is expected to occur. However, ANNs can be easily trained to forecast the minimum concentration occurring in a network, regardless of its location. Such an approach could substantially simplify the application of ANNs to the pipe-sizing problem. Previous applications (Broad et al. 2005; Odan and Ribeiro 2012) of ANNs to finding the optimal design of a WDS have trained one ANN per critical node. The number of critical nodes that can exist in a realistic WDS is significant and can therefore increase the computational demands imposed by using multiple ANNs beyond an acceptable limit. Although the use of minimum concentrations to train ANNs makes the performance of ANNs independent of the number of critical nodes, its effect on accuracy should be quantified to reveal its true potential.
REFERENCES


APPENDIX A

HEURISTIC POST-OPTIMIZATION APPROACHES FOR DESIGN OF WATER DISTRIBUTION SYSTEMS

Manuel A. Andrade¹, Doosun Kang², Christopher Y. Choi³, and Kevin Lansey⁴

Published by the ASCE Journal of Water Resources Planning and Management

¹ Graduate research assistant, Dept. of Agricultural and Biosystems Engineering, The Univ. of Arizona, Tucson, AZ 85721, email: andrader@email.arizona.edu
² Assistant professor, Dept. of Civil Engineering, The Univ. of Suwon, 445-743, San 2-2 Wau-ri, Bongdam-eup, Hwaseong-si, Gyeonggi-do, South Korea, email: doosunkang@suwon.ac.kr
³ Professor, Dept. of Biological Systems Engineering, University of Wisconsin, Madison, WI, 53706, email: cchoi22@wisc.edu
⁴ Professor, Dept. of Civil Engineering and Engineering Mechanics, The Univ. of Arizona, Tucson, AZ 85721, email: lansey@email.arizona.edu
Abstract

This work presents a post-optimization methodology for refining the solutions found by adaptive search algorithms used in the design of large water-distribution networks. The approach employs two heuristics to search for an optimal combination of pipes that, after a reduction of their diameters, will maximize cost savings while continuing to meet design constraints.

Adaptive search methods are often used to design urban water distribution networks when the number of pipes in the network is insignificant. For complex, real-world networks, however, such methods are computationally demanding and they have difficulty finding near-global optima. To identify a solution as close to the global optimum (and in which no pipe can be reduced without violating pressure constraints), requires a high-speed computer potentially running for a long time and also probably some good fortune. The post-optimization approach presented here is shown to be an efficient complement to heuristic search algorithms used in the design of real-world networks. In a network created with the aid of a genetic algorithm, the proposed heuristics found that 4.37% of the pipes with a diameter above the minimum could be further reduced without causing hydraulic failure.

Keywords: WDS design; Optimization; Greedy algorithm.
Introduction

Identifying the most efficient and least costly design for a water distribution system (WDS) is a complex problem. A large number of pipes must be selected from a discrete set of commercial diameters; consequently, the combinatorial problem becomes enormous. Yates et al. (1984) showed that, in theory, developing an algorithm that can select discrete diameters on the basis of minimizing the cost of a water supply network poses a problem of a mathematical class known as non-deterministic polynomial-time hard (NP-hard). According to Templeman (1982), a rigorous algorithm to find an optimal combination of pipes with discrete diameters is not a practical possibility because the computational time required by such an algorithm would be, at best, an exponential function of the number of pipes in the network. Therefore, heuristic methods to determine a solution near to the optimum are appropriate for problems involving pipes of discrete sizes.

In the process of achieving an optimal design for pressurized WDS, the primary objective is to find the combination of elements (pipe diameters, the number and sizing of pumps, etc.) that will minimize one or more objective functions while still meeting operational constraints. A general, single objective-optimization problem, one that aims to minimize the construction cost of the pipes required for any given distribution network, can be formally stated as follows:

Minimize : \( \sum_{k=1}^{M} (C_k \cdot L_k) \)  

Subject to: \( A_j^i \geq A_{min}^i, \quad i=1, \ldots, I; \quad j=1, \ldots, J \)
where $M$ is the number of pipes in the network; $C_k$ is the cost per unit length of pipe $k$; $L_k$ is the length of pipe $k$; $A_j^i$ is the value of the constraint $A$ at node $j$, under demand condition $i$; and $A_{\text{min}}^i$ is the minimum admissible value for $A$, under demand condition $i$; $I$ is the number of demand conditions to be analyzed; and $J$ is the number of nodes in the network. Nodal pressure is the constraint most commonly applied in Eq. (2), and is usually evaluated by performing hydraulic simulations of the network under $I$ extreme conditions of demand (peak, fire flow, etc.). However, additional constraints can be included, such as minimum levels of residual chlorine (Dandy and Hewitson 2004; Broad et al. 2005), which requires a simulation of hydraulic and water quality conditions.

Several optimization methods have been used to find a solution to the problem posed by equations (1) and (2). Traditional methods, such as linear and nonlinear programming, are severely limited because they require that pipe diameters be continuous and they assume that rounded solutions are nearly optimal (or in the final design they split pipes into sections with varying diameters). Furthermore, these methods often either reach a local optimum or fail to solve NP-hard problems involving a large number of variables (Elbeltagi et al. 2005). Further details about the application of these classical methods to the pipe-sizing problem can be found in Walski et al. (2003).

A different type of optimization technique involves what is called an adaptive search method (ASM). ASMs rely on computational algorithms drawn from natural processes, and they have emerged as an effective alternative for solving the problem. ASMs base their search for the optimal solution solely on the evaluation of the objective function (i.e., Eq. (1)), as opposed to the linear and nonlinear programming methods that use
gradient information for the same purpose, and thus the ASMs can easily handle discrete decision variables (Mays 2000). The genetic algorithm (GA) was one of the first ASMs to be introduced (Holland 1992), and it has been used extensively in the design of water supply networks since its application by Simpson et al. (1994). Seeking to reduce the demanding computational needs of GA and to avoid being trapped in local optima, other ASMs have been developed (Elbeltagi 2004). For example, ant colony optimization (Maier et al. 2003), simulated annealing (Cunha and Sousa 1999), harmony search (Geem 2006), and shuffled frog-leaping algorithm (Eusuff and Lansey 2003) are ASMs that have been used in the designing of urban water distribution systems.

Previous attempts to improve the performance of the GA have focused on adjusting its search parameters and internal functions (Dandy et al. 1996; Wu and Simpson 2001). Further enhancement could be achieved if the initial population employed by the GA closely resembled the optimal solution. Such a population could be produced by means of a fast (in comparison with a GA) pre-optimization algorithm (Keedwell and Khu 2005; Kang and Lansey 2011). As yet, however no one has published a methodology for improving a locally optimal design obtained with an ASM. This work does so by presenting a problem statement describing how such a goal could be reached (i) if additional modifications in the design of a network were limited to reducing the diameter of pipes, and (ii) if two heuristic algorithms were used to reach the best combination of such adjustments that would bring the modified design into closer alignment with the global optimum.
Problem statement

A near-optimal WDS design (referred here as original design) — one achieved by using an ASM to find a solution to the optimization problem posed by equations (1) and (2) — may be further improved by changing the diameter of a few selected pipes. If any modification is restricted to a reduction of diameters, then the original design can be improved by lowering the objective function value — if there exists at least one pipe with a diameter that can be reduced without violating a constraint imposed for the original design. If a set $S$, that consists of pipes that individually can be decreased in size without violating such constraints, contains more than one element, then the best possible modification to the original design can be obtained by solving the following problem: find the subset of pipes in $S$ that, when their diameters are collectively reduced, yields the greatest decrease in the objective function relative to the near-optimal original design.

Like the initial design problem, this is a combinatorial problem. If the set $S$ contains $n$ elements, then the total number of possible modifications will be the sum of each combination of $n$ and $i$, for $i=1,\cdots, n$. For small networks, $S$ may contain only a few elements (or none), and the solution is trivial. However, for real water distribution networks, an enumeration scheme becomes infeasible, even for a modest sized set. Thus, finding the best combination of adjustments will require solving a computationally intensive optimization problem. Nevertheless, a good approximation of the best solution can be quickly achieved by using a heuristic approach that selects a pipe ($k$) in $S$, one at a time, and reduces its diameter to determine if doing so will improve the solution. This
approach is known as a ‘greedy algorithm’ and the following section proposes two heuristics of this kind.

Two heuristic approaches

Consider two sets of pipes, $S$ and $T$. The first set ($S$) contains all the pipes in a given network with diameters that can be reduced individually while still meeting the design constraints imposed by Eq. (2). The second set ($T$) is a subset of $S$ consisting of pipes that can be reduced collectively without violating the same constraints. According to the objective function, Eq. (1), an optimal solution will be the set $T$ that contains those pipes that, when their diameters are decreased, will provide the greatest cost reduction. The general greedy algorithm to obtain $T$ can then be described in pseudo-code as follows:

Greedy algorithm

Objective: improve the design of a water supply network obtained using an ASM to deal with the optimization problem defined by equations (1) and (2).

Restrictions: The following algorithm is only intended to refine the diameter of pipes; the dimensions or numbers of other network components, such as pumps, should be specified in advance. The algorithm may be used when analyzing water supply networks with realistic dimensions; however, little or no improvement may be achieved if applied to networks involving only a few components.
1. [Create set $S$]
   Repeat steps a) to d) for each pipe in the network:
   
   a) Reduce the diameter of the current pipe by one commercial size
   b) Perform a hydraulic analysis for each demand condition analyzed by the ASM
   c) Add the current pipe to set $S$ if, for each demand condition, all the nodes in the
      network satisfy the constraints considered by the ASM (Eq. 2)
   d) Return the diameter of the current pipe to its previous size

2. [Identify the best candidate pipe in set $S$]
   Rank pipes in set $S$ by repeating steps a) to d) for each pipe in set $S$:
   
   a) Reduce the diameter of the current pipe in set $S$ by one commercial size
   b) Perform a hydraulic analysis for each demand condition analyzed by the ASM
   c) Assign a decision value to the current pipe, based on a decision rule (defined in the
      following section)
   d) Return the diameter of the current pipe to its previous size

3. [Reduce the diameter of the best candidate pipe in set $S$]
   Decrease by one commercial size the diameter of the pipe $k$ with the highest decision rule
   value in set $S$, and add $k$ to set $T$

4. [Update set $S$ to reflect the modification of pipe $k$]
   Repeat step 1 to create a new set $S$ while the size of pipe $k$ is reduced by one commercial
   diameter

5. [Apply a stopping criteria]
   Go to step 2 if $S$ is a nonempty set; otherwise, stop the algorithm

Defining an appropriate decision rule is the most critical component of the
algorithm because achieving a good solution using this method strongly depends on the
selection of the best candidate pipe $k \in S$ at each decision step. If maintaining the
minimum allowable nodal pressure head is the only constraint, the ideal candidate $k$ in $S$
should have two characteristics: (i) it should provide the highest savings after reducing its
diameter by one commercial size, and (ii) it should produce the least pressure drop at the
critical node (with the lowest pressure in the network). However, given that reducing the diameter of longer pipes results in greater savings and greater pressure reductions, finding a pipe with both attributes would be unlikely, especially if the pipe lies in a pathway leading to the critical node. Therefore, given the tradeoffs between these two attributes, care should be taken when defining the decision rule for the greedy algorithm.

In general, the critical node refers to the node for which the predicted output is closest to the constraint bound (i.e., minimum constraint slack or surplus). For example, if Eq. (2) is a pressure constraint, and two demand conditions (peak and fire flow) are analyzed, modeling each demand yields the node with the lowest pressure in the network. If nodes $a$ and $b$ are, respectively, the nodes with the lowest pressures for the peak and fire flow demands, then the critical node is defined as the node with the minimum $\Delta p = P - P_{\text{min}}$, where $P$ is the pressure at either node $a$ or $b$, and $P_{\text{min}}$ is the threshold pressure for the respective condition of demand. It should be noted that there should be a critical node for each constraint considered in Eq. (2), and thus, for example, there may be a critical node for a pressure constraint and another for a water quality constraint (although the same node may serve for both constraints). Finally, the critical node(s) should be updated after every modification to the network (step 3 in the algorithm previously described) so as to remain the node with the lowest value.

In order to evaluate the computational effort required for the previous algorithm, we can consider the maximum number of executions it makes of a hydraulic simulation engine. Being $m$ the number of pipes in the network, $n$ the number of elements in the initial set $S$, and $d$ the number of demand conditions to be analyzed, the first iteration will
require $dm$ executions of such engine for step 1, and $dn$ for step 2. Similarly, the second iteration demands $dm$ executions for step 1, and a maximum of $d(n-1)$ for step 2. Given that there cannot be more than $n$ iterations, the described Greedy algorithm executes a hydraulic model a maximum number of $M_g$ times, where $M_g$ can be obtained with the following formula:

$$M_g = d \left[ mn + \sum_{i=1}^{n} i \right] = d \left[ mn + \frac{n(n+1)}{2} \right]$$

\hspace{1cm} (3)

First heuristic approach: a greedy algorithm with a weighted decision rule

To implement the above algorithm, it is necessary to define a decision rule that can measure the relative merit of a candidate pipe $k$ in comparison with all other elements in $S$. Assuming for the sake of simplicity that pressure is the only constraint in Eq. (2), the decision rule can be based on the tradeoff between: (i) the cost reduction obtained, and (ii) the pressure decline at the critical node (caused by decreasing the diameter of $k$).

Defining such a rule can be achieved by assigning relative weights to each candidate pipe $k \in S$. If, for example, $k$ is the pipe that yields the highest cost savings ($C_{max}$) when its diameter is reduced, then $k$ receives a relative weight due to savings of $W_{Ck} = 1.0$. On the other hand, if $k$ is the pipe that, under the same modification, supplies the lowest savings ($C_{min}$), then $k$ receives a relative weight of $W_{Ck} = 0$. Similarly, if reducing the diameter of pipe $k$ yields the least pressure reduction —or the highest pressure $P_{max}$, as a consequence— at the critical node, then $k$ receives a relative weight of $W_{Pk} = 1.0$. But if
reducing the diameter of pipe \( k \) produces the largest pressure drop—or the lowest pressure \( P_{\text{min}} \)—at the critical node, then \( k \) receives a relative weight of \( W_{pk} = 0 \).

To differentiate the importance given to each factor in the decision rule, two parameters, \( m_C \) and \( m_P \), are defined. \( m_C \in [0, 1] \) is a multiplier for \( W_{Ck} \), and \( m_P = (1 - m_C) \) is a multiplier applied to \( W_{pk} \). A convenient decision rule for the greedy algorithm can then be stated as: “Select the pipe \( k \in S \) with the highest decision value \( D_k \),” where \( D_k \) is determined by the weighted sum of the two factors:

\[
D_k = m_C \cdot W_{Ck} + m_P \cdot W_{Pk} = m_C \left[ \frac{C_k - C_{\text{min}}}{C_{\text{max}} - C_{\text{min}}} \right] + (1 - m_C) \left[ \frac{P_k - P_{\text{min}}}{P_{\text{max}} - P_{\text{min}}} \right]
\]

where \( C_k \) is the savings obtained by reducing pipe \( k \); \( P_k \) is the pressure at the critical node, after the modification of pipe \( k \). \( C_{\text{max}} \) and \( C_{\text{min}} \) are, respectively, the maximum and minimum cost savings that can result from a reduction of any pipe in set \( S \); \( P_{\text{max}} \) and \( P_{\text{min}} \) are, respectively, the maximum and minimum pressures, at the critical node, that can result from a reduction of any pipe in set \( S \).

In Eq. (4), a decision value \( (D_k) \) of 1.0 is assigned to an ideal candidate, and a value of 0 to the worst possible candidate, while all other pipes receive a weight of between 0 and 1. The pipe that combines the greatest savings with the least pressure reduction at the critical node receives the highest decision value \( (D_k) \) in \( S \) and is added to the solution set \( T \) (step 3 of the algorithm). Note that additional terms can be added to Eq. (4) if the decision rule requires more parameters to account for more constraints, such as minimum levels of water quality. The flowchart in Fig. 1 describes in detail how to create sets \( S \) and
and the integration of the weighted decision rule into the greedy algorithm described above with a pseudo-code.

-- Fig. 1. Flowchart of the greedy algorithm with a weighted decision rule and a single (pressure) constraint in Eq. (2). $D_i$ is the diameter of pipe $i$; $\text{MinDiam}$ is the minimum diameter that can be used for the design; $\text{Npipes}$ is the number of pipes in the network; $\text{Nnodes}$ is the number of nodes in the network; $P_n$ is the pressure at node $n$; and $P_{\text{threshold}}$ is the minimum nodal pressure for the given condition of demand.

Second heuristic approach: a partial enumeration with a greedy algorithm

Greedy algorithms are shortsighted; that is, once an element has been selected, there is no way to reconsider the previous decisions. However, the greedy algorithm presented
here as a first heuristic can be combined with a one step ahead enumeration scheme that can “foresee” the results of a pipe diameter change (Figure 2). Instead of ranking pipes in set \( S \) based on Equation (4), this second heuristic uses the concept of potential cost reduction \( C_{Pk} \) to determine the pipe that, when modified, will provide the best condition with respect to the final solution. Here, \( C_{Pk} \) is defined as the maximum cost reduction that can be achieved by the first method after the diameter of pipe \( k \) has been reduced by one commercial size. In other words, if the diameter of pipe \( k \) is reduced by a current iteration, \( C_{Pk} \) will indicate the benefits of this change in comparison with the benefits to be derived from decreasing the diameter of any other pipe in set \( S \).

The steps in this approach are:

1. [Create set \( S \)]
   Apply the first step of the first approach to create set \( S \)
2. [Identify the best candidate pipe in set \( S \)]
   Rank pipes in set \( S \) based on their potential to reduce cost \( C_{Pk} \) by repeating steps a) to d) for each pipe in set \( S \):
   
   a) Reduce the diameter of current pipe \( k \) by one commercial size.
   
   b) Follow the first step of the first method to create a subset of \( S \), named \( S_k \). The subset \( S_k \) should contain all pipes which diameter can be reduced individually while meeting design constraints, despite the modification of pipe \( k \).
   
   c) Apply steps 2 to 5 of the first method to subset \( S_k \). Fixed multipliers \( m_C \) and \( m_P \) are used to assign a decision value to each pipe in \( S_k \), according
with Eq. (4). The outcome of such steps is a new set, named $T_k$, containing pipes which collective reduction of diameters maximizes cost savings (according with the first method) without violating design constraints, despite the modification of pipe $k$.

d) Calculate the potential cost reduction ($C_{Pk}$) obtained if the diameter of pipe $k$ and the diameters of all pipes in set $T_k$ are reduced.

3. [Reduce (definitely) the diameter of the best candidate pipe in set $S$]
   Decrease by one commercial size the diameter of the pipe $k$ with the highest $C_{Pk}$, and add $k$ to the solution set $T$. If two or more pipes produce the highest $C_{Pk}$, then use the pipe’s length as a tie-breaking criteria and reduce the diameter of the pipe having the greatest length among such pipes.

4. [Update set $S$ to reflect the modification of pipe $k$]
   Repeat step 1 to update set $S$, which should include only those pipes that fulfill the definition of $S$, despite the modification of pipe $k$.

5. [Apply a stopping criteria]
   Go to step 2 if $S$ is a nonempty set; otherwise, stop the algorithm.

Note that the main difference between the two algorithms resides in step 2; while the first method selects a pipe in set $S$ based on a decision value obtained with Equation 3, the second does so by comparing potential cost savings ($C_{Pk}$) that are estimated with the help of the first method. As presented (to evaluate the computational effort of the first heuristic), the maximum number ($M_e$) of executions that the second algorithm can make out of a hydraulic simulation engine can be calculated as follows:

$$M_e = d \left[ mn + n^2 M_s \right]$$ (5)
where $m$ is the number of pipes in the network, $n$ is the number of elements in the initial set $S$, $d$ is the number of demand conditions to be analyzed, and $M_g$ is defined by (3).

Equation (5) can be obtained by considering that the second algorithm: (i) requires $dm$ executions of the hydraulic engine for step 1, (ii) executes $n$ times the greedy algorithm to determine the best candidate pipe at step 2, and (iii) has a maximum number of $n$ iterations.

![Flowchart](image)

**Fig. 2.** Flowchart of the partial enumeration with a greedy algorithm. This second method is based on a recursive calling of the first method to evaluate the potential cost reduction ($C_{Pk}$) that can be achieved in future iterations if pipe $k$ is reduced at a current stage. $n(S)$ is the number of elements in set $S$. 
Case study

To validate the proposed methodologies, they were tested on a real WDS that has been optimized using a GA in previous study by Kang and Lansey (2011), as depicted in Figure 3. This network consists of a pumping station, 935 nodes, and 1274 pipes. Kang and Lansey (2011) optimized the system with a combination of pipe sizes and pumping station capacity that minimized the sum of construction and operation costs for the network subject to a minimum allowable pressure of 28.122 m (40 psi) for the average and peak flow conditions, and 14.061 m (20 psi) for the fire flow. The pumping station has a different operation for each demand condition and provides a total head of 166 m, 190 m, and 173 m, for the average, peak, and fire flow demands, respectively. The average demand is given as 177 l/s (2,808 gpm), and a peak factor of 1.75 was applied to this demand to create an instantaneous peak condition of 310 l/s (4,915 gpm). The fire flow has a total demand of 342 l/s (5,432 gpm), which is originated by assigning a daily peak factor of 1.40 to the average demand and by adding a fire flow of 95 l/s (1,500 gpm) to a node located in the southeast corner (indicated with a circle in Figure 3.). Although multiple emergency demands are needed to ensure that all fire flow conditions will be satisfied for the given network, the heuristics proposed in this paper should test every demand condition analyzed by the ASM in order to provide the design from which the heuristics depart. Thus, in the current analysis we assigned the same fire flow condition in order to make sure it was consistent with the preceding study by Kang and Lansey (2011).
This network is particularly appropriate for testing a post-optimization approach because a pre-optimization scheme has been applied to improve the initial set of solutions in the GA search process. By providing an initial population, selected so to be far closer to the optimal solution than a randomly generated population, Kang and Lansey (2011) were able to improve the GA’s computational performance and provide a better solution. Ten independent GA runs were made, and each run was comprised of a total of 250,000 function evaluations (500 generations with a population of 500 individuals). From the ten independent GA runs, nine converged into the single near-optimal value shared by the network shown in Fig. 3. Furthermore, the objective function values were already very close to the final solution after 25,000 function evaluations.

The test network has a total pipe length of 252.45 km. In the near-optimal design obtained by the previous study, 923 pipes, representing 71.5% of the network’s total length, have a diameter of 152 mm (6 in) (the minimum allowable diameter to meet specifications for fire flow), 285 pipes (23.2%) are 203 mm (8 in) in diameter, and the remaining 66 pipes (5.2%) have diameters of 254 mm (10 in) or larger (Fig. 3). The critical node, according to the definition stated previously, always happened to be the same (see Fig. 3) in the case study network because it was the farthest node from the source and also had the highest elevation. It experiences pressure heads of 30.518 m, 30.139 m, and 14.069 m for average, peak and fire flow conditions, respectively. Since the system has been optimized for the three demand conditions, the proposed post-optimization approaches also seek solutions that are subject to the three pressure constraints used for the original design.
To estimate the pipe construction cost, Kang and Lansey (2011) applied the cost function proposed by Clark et al. (2002):

\[ y = a + b(x^c) + d(u^e) + f(xu) \]

where \( y \) is the cost of a particular component ($/ft, 1 \text{ ft}=0.3048 \text{ m}); x \) is a design parameter (for example, pipe diameter in inches, 1 inch=2.54 cm); \( u \) is an indicator variable (for example, class of pipe); and \( a, b, c, d, e, \) and \( f, \) are parameter values estimated using regression techniques. The total construction cost of a pipe (with a length of 1 ft) can be obtained as the sum of the outcomes of Eq. (6) for each of the following components: (i) pipe material, (ii) trenching and excavation, (iii) embedment, (iv) backfill and compaction, and (v) valve, fitting and hydrant cost. The same parameters applied by Kang and Lansey (2011) were used in this study. The savings obtained by reducing the diameter of a given pipe were estimated using the difference in the results obtained by Eq. (6) for the pipe’s original diameter and its diameter after reduction. Detailed information about the cost function can be found in Clark et al. (2002). Additional information about the system’s layout and pre-optimization process can be found in Kang and Lansey (2011).

**Results**

Both heuristics were implemented in the C programming language and linked with EPANET (Rossman 2000) to perform hydraulic calculations using the EPANET toolkit. The first step in both methods is to populate set \( S \) by reviewing all pipes with diameters...
greater than 152 mm (6 in). Total 69 pipes are identified to be included in set $S$ and all of them have a diameter of 203 mm (8 in). Thus, the total savings obtained after reducing pipe sizes in the solution set $T \subseteq S$ can be represented by the product $C_r L_T$, where $C_r$ is the unit cost saving produced by the reduction of unit length of pipe from 203 mm (8 in) to 152 mm (6 in), and $L_T$ is the total pipe length in set $T$.

*Greedy algorithm with a weighted decision rule*

The results of this method are strongly dependent on the multipliers ($m_P$ and $m_C$) applied in Eq. (4). Figure 4(a) shows the results in terms of the reduced pipe length $L_T$ and the number of pipes reduced as a function of $m_P$ and $m_C$. The results obtained for the extreme combinations ($m_P=1$, $m_C=0$) and ($m_P=0$, $m_C=1$) illustrate the consequences of using a single factor decision rule. The first case ($m_P=1$, $m_C=0$) finds the highest number of elements in set $T$ (16) at the expense of a modest total length $L_T=2.56$ km because it only considers the pressure reduction at the critical node. The second extreme case ($m_P=0$, $m_C=1$) uses cost savings as its only decision parameter and not only produces the smallest set $T$ (8) but also the shortest total length $L_T=2.21$ km. Better results arise when combinations fall between these two cases, especially when $m_C$ is slightly higher than $m_P$. The greedy algorithm finds the best results when $0.6 \leq m_C \leq 0.7$ and $0.4 \geq m_P \geq 0.3$. Within these ranges, the solution set $T$ contains 10 elements (indicated in Fig. 3) with a total reduced pipe length of $L_T=3.14$ km, which represents 1.24% of the network’s total length and 4.37% of the pipes with diameters above the minimum allowable (152 mm). After decreasing the diameters of all pipes in $T$, the pressure heads at the critical node comply
with the minimum allowable threshold values and remain nearly the same for average and peak conditions, respectively. The fire flow condition also remains above the pressure constraint at 14.061 m.

The unit cost saving \( (C_r) \) produced by the reduction of 1 m of pipe from 203 mm to 152 mm, according with Eq. (6), is $18 (US dollars). Thus, the greedy algorithm found a minimum cost reduction of $39,718 with multipliers \( (m_P=0, m_C=1) \) and a maximum cost reduction of $56,592 (see Fig. 3) with multipliers within the range \( (0.6 \leq m_C \leq 0.7) \) and \( (0.4 \geq m_P \geq 0.3) \). For the analyzed network, these maximum savings represent 1.04% of the amount required for pipe material of those pipes with diameters equal or greater than 203 mm (8 in).

When run on a computer with an ‘Intel Core 2 Duo’ processor (2.4 GHz) and 4 GB of RAM memory, the simulation required a minimum execution time of 11 s (when \( m_P=0, m_C=1) \) and a maximum of 42 s (when \( m_P=1, m_C=0) \).

A partial enumeration using a greedy algorithm

The second heuristic makes an extensive evaluation of all pipes in set \( S \) at the expense of a longer computation time. However, the second heuristic is capable of achieving the best solution obtained by the first heuristic, for most combinations of multipliers \( m_C \) and \( m_P \), i.e., less sensitive to the decision rule. Such performance is determined by the decision rule of the method: “reduce the diameter of the pipe which modification provides the highest potential cost reduction”. As explained in the description of this method, the potential cost reduction \( (C_{P_k}) \) is used to estimate the
savings that can be achieved by future iterations if the diameter of pipe \( k \) is decreased. By reducing the pipe with the highest \( C_{P_k} \) at a current iteration, the second heuristic guarantees the achievement of a higher \( C_{P_k} \) at the next iteration, and thus secures a final solution equal to (or near to) the optimal.

Table 1 shows the sequence in which the method adds pipes to the solution set \( T \), for three combinations of multipliers \( m_P \) and \( m_C \). When \((m_P=1, m_C=0) \) and \((m_P=0.7, m_C=0.3)\), step 2 of the approach employs pressure as its main selection criteria and thus uses shorter pipes to obtain the potential cost reduction \((C_{P_k})\). As a result, step 3 reduces the diameter of longer pipes at initial iterations. The selection sequence followed by the first case \((m_P=1, m_C=0)\) clearly illustrates the benefits of the second heuristic. The modification of pipe ‘h’ seems as a disadvantageous decision at iteration 1 because the magnitude of the potential savings \((C_{P_k})\) is far from its optimal. Nevertheless, the method is still capable to reach the peak value of \( C_{P_k} \) at iteration 6 and thus guarantees the achievement of the optimal solution at subsequent iterations. Note that the first approach could not find the optimal solution for those three cases (Fig. 4a).

Fig. 4(b) displays the results obtained by the second heuristic as a function of pressure \((m_P)\) and savings \((m_C)\) multipliers. For nine (out of eleven) cases, the second heuristic arrived to the same best solution ($56,592 saving) that the first method reached for only two combinations (Fig. 4a). Thus, a partial enumeration combined with a greedy algorithm provides more consistent results than a greedy algorithm without modifications. Besides the sequence of selection, multipliers \( m_P \) and \( m_C \) mainly affect the
execution time that, for the same computation environment, now has a minimum of 17 min (when $m_P=0$, $m_C=1$) and a maximum of 44 min (when $m_P=1$, $m_C=0$).

Although a strict demonstration will require the use of an enumeration scheme, it is very likely that the best solution to be achieved by both methods will be the optimal solution to the problem of finding the best combination of pipes in set $S$—a solution that maximizes savings without violating the pressure constraint. The second method achieves this solution more consistently because it makes an exhaustive analysis of each element in the set of candidate pipes, $S$. Here, only two decision factors are considered, i.e., cost and pressure. If more constraints are included in the problem, the first method will be less likely to find an optimal solution since it is highly dependent on the decision rule. The second method, on the other hand, will be competitive and consistent because it is less sensitive to the selection of a decision rule. Thus we recommend using an iterative application of the first method with controlled increments of the multipliers used in Eq. (4), as presented here. In such a case, the computational effort for the first method should be updated as follows:

$$M_g' = M_g \left( \frac{1}{\text{inc}} + 1 \right)^{Nt-1}$$

where $M_g'$ is the maximum number of executions of the hydraulic model when several multipliers are tested in Eq. (4), $\text{inc}$ is the increment between multipliers in Eq. (4), and $Nt$ is the number of terms in Eq. (4).
Table 1. Pipe selection sequence followed by the second heuristic for the solution set $T$, with different combinations of multipliers $m_P$ and $m_C$. $L_k$ is the length of the pipe $k$ which diameter is reduced by the second heuristic at a given iteration. $C_{Pk}$ is the highest potential cost reduction obtained by the step 2 of the second heuristic at the same iteration.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>$m_P=1$, $m_C=0$</th>
<th>$m_P=0.7$, $m_C=0.3$</th>
<th>$m_P=0$, $m_C=1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pipe</td>
<td>$L_k$ (m)</td>
<td>$C_{Pk}$ ($)</td>
</tr>
<tr>
<td>1</td>
<td>h</td>
<td>609</td>
<td>48,780</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>355</td>
<td>53,100</td>
</tr>
<tr>
<td>3</td>
<td>g</td>
<td>468</td>
<td>54,720</td>
</tr>
<tr>
<td>4</td>
<td>f</td>
<td>141</td>
<td>54,720</td>
</tr>
<tr>
<td>5</td>
<td>e</td>
<td>383</td>
<td>54,720</td>
</tr>
<tr>
<td>6</td>
<td>a</td>
<td>250</td>
<td>56,592</td>
</tr>
<tr>
<td>7</td>
<td>c</td>
<td>267</td>
<td>56,592</td>
</tr>
<tr>
<td>8</td>
<td>d</td>
<td>303</td>
<td>56,592</td>
</tr>
<tr>
<td>9</td>
<td>j</td>
<td>180</td>
<td>56,592</td>
</tr>
<tr>
<td>10</td>
<td>i</td>
<td>188</td>
<td>56,592</td>
</tr>
</tbody>
</table>
Fig. 3. Applied water distribution network. This near-optimal design was adopted from Kang and Lansey (2011). Arrows indicate the best combination of pipes which diameter could be further reduced by the proposed post-optimization approaches, while still meeting pressure requirements. A character between parentheses identifies each pipe. Length of pipes is shown at the right of identification characters and numbers below lengths correspond to cost reductions obtained with a unit cost saving ($C_r$) of $18 (US dollars) per meter of pipe decreased from 203 mm to 152 mm.
Fig. 4. Post-optimization results obtained with (a) the first heuristic (greedy algorithm) and (b) the second heuristic (partial enumeration using a greedy algorithm). The number of pipes in the solution set $T$, and the reduced pipe length $L_T$ are presented as a function of the combination of the pressure multiplier ($m_P$) and the savings multiplier ($m_C$) used in Eq. (4).

**Potential applications and limitations**

Although the proposed heuristics have been applied here to improve the results of an ASM used in a post-optimization stage, analogous approaches can be developed to improve the performance of an ASM used in previous stages of the optimization process. For example, both algorithms can be applied to obtain fast (and feasible) approximations of the optimal solution if, instead of a near-optimal solution, the algorithms depart from a network in which all the diameters have been set to the maximum commercial size. In such a case, the initial set $S$ contains all the pipes in the network, and the algorithms will reduce, one at a time, the diameter of any pipe the modification of which provides the highest decision value, as defined by the decision rule applied in the process. The described application can be useful in generating an initial population for a GA that, as mentioned in the introduction, achieves better solutions if this population is closer to the
optimal solution. We followed this approach and applied the first heuristic to obtain a fast approximation of the pipe-sizing problem proposed by Fujiwara and Khang (1990) for the Hanoi Network. In accordance with Eq. (7), a hundred and one feasible solutions were obtained by repeatedly applying the first heuristic with increments of 0.01 for the cost \( m_C \) and pressure multipliers \( m_P \). The best of these networks has a total cost of US $6,265,995 and was obtained using multipliers \( m_P=0.68 \) and \( m_C=0.32 \). In terms of percentage, the cost of this network is only 3.16% more than the best solution obtained by Savic and Walters (1997) using GAs, and 3.45% more than the best solution obtained by Geem (2006) using a harmony search.

The main purpose of this work is to introduce the application of heuristics that improve the outcomes obtained by ASMs dealing with the optimal design of WDS. Thus, for the sake of simplicity, this work only deals with the well-known, single-objective optimization problem posed by Equations (1) and (2). Within this framework, a network is evaluated only with respect to its total cost, and a given solution is better with respect to another if its construction represents any savings. However, it should be noted that a superior design for a WDS, one that maximizes net benefits, could only be achieved if several other factors were considered in the optimization process. According to Walski (2001), optimization models based on cost minimization will tend to eliminate system capacity and thus will reduce the project’s net benefit when demands increase in the future. In order to remain useful in solving different types of WDS design problems, the post-optimization heuristics presented here should be adapted so as to better fit the needs of the given problem. For example, the purpose of the post-optimization analysis might
be changed from “find the set of pipes which, when the diameters are reduced, maximizes savings” to “find the set of pipes which, when the diameters are increased, maximizes the system’s reliability without exceeding a certain cost,” and the terms in Eq. (4) should be also modified so as to assign the highest decision value $D_k$ to a pipe providing the best conditions for the new problem. For multi-objective optimization problems, however, more research is required in order to define proper adjustments for the proposed heuristics. Although additional terms should most likely be included in Eq. (4) – one for every objective function and another for every operational constraint (minimum pressure, minimum water quality, etc.) – the modification of a non-dominated solution may transform the new solution into another solution already present in the Pareto-front.

**Conclusions**

This work presents two heuristic methods to improve local optimal solutions in a post-optimization framework. The methods will be especially useful when refining the design of real networks with a significant number of pipes. The problem statement defined here can serve as a basis for future research aimed at finding post-optimization approaches for achieving better designs for large urban water distribution networks.

This work also introduces an original decision rule that can be adapted to include new parameters. Given the proper multipliers $m_C$ and $m_P$, a fast greedy algorithm (first method) that incorporates this decision rule is capable of finding the best solution achieved by the second method. However, the latter proved to be less sensitive to the decision rule when it was applied; thus, the second heuristic is a reliable method for
solving the problem at the expense of moderate computation time. Both methods are fast for the given example network, and computing time and costs are trivial with respect to real savings. Both heuristics successfully found a reduction of $56,592 US dollars in the example network.

**Acknowledgements**

This material is based in part upon work supported by the National Science Foundation under Grant No. 083590. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.
References


APPENDIX B

ENHANCED ARTIFICIAL NEURAL NETWORKS ASSISTING THE OPTIMAL DESIGN OF WATER DISTRIBUTION SYSTEMS

Manuel A. Andrade\textsuperscript{1}, Christopher Y. Choi\textsuperscript{2}, and Kevin Lansey\textsuperscript{3}

To be submitted to the ASCE Journal of Water Resources Planning and Management

\textsuperscript{1} Graduate research assistant, Dept. of Agricultural and Biosystems Engineering, The Univ. of Arizona, Tucson, AZ 85721, email: andrader@email.arizona.edu
\textsuperscript{2} Professor, Dept. of Biological Systems Engineering, University of Wisconsin, Madison, WI, 53706, email: cchoi22@wisc.edu
\textsuperscript{3} Professor, Dept. of Civil Engineering and Engineering Mechanics, The Univ. of Arizona, Tucson, AZ 85721, email: lansey@email.arizona.edu
Abstract

Achieving an optimal design for a typical water distribution system (WDS) essentially involves determining which combination of pipes and arrangements will produce the most efficient and economical network. Solving this problem is, however, a complex process, and including water quality constraints will add demanding, extended-period simulations. Consequently, the task is well suited to computationally intensive heuristic methods, and employing artificial neural networks (ANNs) can significantly reduce the amount of computation time needed. ANNs can in fact approximate disinfectant concentrations in a fraction of the time required by a conventional water quality model. This study presents a methodology for improving the accuracy of ANNs applied to the optimal design of a WDS, and it does so by means of a probabilistic approach based on the fast finding of a network similar to the optimal WDS. This study also presents a comparison of different ANN architectures and presents two case studies that were used to determine the extent to which a network’s size will affect the performance of an ANN. When applied to both case studies, those ANNs trained with a probabilistic dataset and having a non-conventional architecture were shown to be more accurate than their counterparts, which had been trained using a random dataset and had a customary architecture.

Keywords: WDS design; Artificial Neural Networks; Metamodeling.
Introduction

The Environmental Protection Agency of the United States has imposed increasingly stricter water quality (WQ) regulations on water distribution systems (WDS) to minimize public health risks. Such regulations demand that chlorine levels be kept within a certain range throughout the entire distribution network; also, a minimum chlorine level must be maintained to prevent bacterial regrowth. Similarly it is equally important to maintain that level below a given threshold to prevent byproducts, including carcinogens and a disagreeable taste and odor. These constraints must be considered when planning a water supply network, but doing so can prove difficult because of the combined effect of two factors: (i) the complexity of the optimization problem, and (ii) the computational requirements of available WQ models.

The optimization of a WDS is a combinatorial problem for which no definitive algorithmic solution has been found. The problem involves finding a combination of elements (pipe diameters, disinfectant dosages, the number and size of pumps, etc.) that will minimize one or more objective functions while still meeting operational constraints. The mathematical formulation of the simplest—and also most scrutinized—form of the problem, where the only objective is to minimize the construction cost of the pipes, can be formally stated as follows:

\[
\text{Minimize: } \sum_{k=1}^{n} (Uc_k \cdot L_k)
\]

Subject to: (i) \(A_j^q \geq A_{min}^q\), \(q \in [1,Q]\), \(j \in [1,m]\); and (ii) \(D_k \in S_D\)
where $n$ is the number of pipes in the network; $U_{ck}$ is the cost per unit length of pipe $k$; $L_k$ is the length of pipe $k$; $A^q_{min}$ is the value of the constraint $A$ at node $j$, under demand condition $q$; $A^q_{min}$ is the minimum admissible value for constraint $A$, under demand condition $q$; $Q$ is the number of demand conditions to be analyzed; $m$ is the number of nodal demands in the network; $D_k$ is the diameter of pipe $k$; and $S_D$ is the set of commercial dimensions from which $D_k$ must be selected.

Several optimization methods have been applied in the hope of finding a solution to the problem posed by Eq. (8). The traditional methods, which involve linear and nonlinear programming, require continuous decision variables and thus are not well suited to such a task, which includes selecting diameters from a discrete set of commercial sizes—Yates et al. (1984), revealed the true complexity of this pipe-sizing problem by demonstrating that equation (8) belongs to a mathematical class known as NP-hard (non-deterministic polynomial-time hard). According to Gupta (1999), using a rigorous algorithm to solve NP-hard problems is practically impossible because the computational time required by such an algorithm would be, at best, an exponential function of the number of pipes in the network. Hence, the lack of more robust techniques has forced researchers to look for alternative optimization methods. Among these, heuristic methods—also known as stochastic methods—have emerged as an effective approach to solve the pipe-sizing problem. Heuristic methods rely on computational algorithms drawn from natural processes and have been applied in practically all of their various forms, which include genetic algorithms (GA) (Simpson and Dandy 1994), ant colony optimization (Maier et al. 2003), simulated annealing
(Cunha and Sousa 1999), harmony search (Geem 2006), and the shuffled frog-leaping algorithm (Eusuff and Lansey 2003). Heuristic methods readily accept the use of discrete decision variables because their search is based solely on the evaluation of the objective function and thus (unlike linear and nonlinear programming methods) they do not need gradient information. However, as pointed out by Lansey (2000), the advantage gained by not requiring gradients is also the main shortcoming of heuristic methods: a large number of simulations must be generated to advance towards the optimal solution because these methods only use the information obtained from the objective function. The prolonged computation time inherent to heuristic methods severely limits the kind of constraint that can be considered for the solution of Eq. (8). The implementation of nodal pressure constraints requires the execution of a few hydraulic simulations of the network under extreme and steady demands (peak, fire flow, etc.), adding a moderate burden to the search for an optimal WDS. The implementation of water quality constraints, on the other hand, involves the generation of computationally expensive simulations of any network subjected to unsteady demands over a period of time, and this can extend the optimization process beyond an acceptable limit. Hence, a mathematical model capable of forecasting concentration levels with precision and speed would be of significant valuable to the designer of a water supply network.

A computational device known as an artificial neural network (ANN) could create such a model. An ANN processes data inspired by the way the human brain processes information to acquire knowledge. According to Hornik (1989), ANNs are universal approximators, i.e., are capable of approximating any measurable function to any desired
degree of accuracy. Therefore, ANNs are capable of handling complex associations between inputs and outputs, and for this reason they have been used extensively in analyses involving function fitting, pattern recognition, classification of data, and time series. In water supply engineering, ANNs have been used mainly to classify and function fit field data (Bowden et al. 2006; Polycarpou et al. 2002; May et al. 2008; Dandy et al. 2005). However, ANNs can also serve as a substitute for the deterministic models used in tasks that involve high computational demands, such as optimization routines (Lingireddy et al. 2005). Broad et al. (2005) used an ANN as a fast replacement of a computationally expensive WQ solver, and they then passed the estimations generated by the ANN to a genetic algorithm (GA) searching for the optimal design of a WDS. Given these applications, it seems reasonable to conclude that a properly trained ANN should be capable of approximating the minimum chlorine concentrations that the WQ solver would take much longer to predict. A GA calling on such an ANN should be able to achieve, in a reasonable time, a nearly-optimal solution, one that takes into account all WQ constraints. ANNs can also be trained to estimate the magnitude of pressure constraints (Broad et al. 2005), but this study only uses ANNs to forecast WQ constraints because of the efficient numerical methods implemented by widely-used WDS modeling software, such as Epanet (Rossmann 2000).

In this study, we sought to improve the accuracy and execution time of ANNs applied to the optimal design of a WDS. To do so, we considered the effects that three factors have on the outcomes of an ANN used to forecast the minimum residual disinfectant concentration in a WDS: (i) the type of architecture selected for the ANNs, (ii) the type
of data used for the training of ANNs, and (iii) the network’s size. The third factor (effect of the network’s size) was analyzed by comparing the outcomes of ANNs trained for two case studies. The first case study involved a small-sized WDS with 34 pipes, whereas the second case involved a large WDS with 1,090 pipes.

Previous applications of ANNs for the optimal design of WDS have required the training of multiple ANNs with a single output neuron (ON) (see Figure 1), where each ANN is trained to forecast the magnitude of a constraint at a critical node (Broad et al. 2005; Broad et al. 2006; Bi and Dandy 2012). Using several ANNs with one ON is believed to be the most convenient practice because ANNs with multiple ONs are expected to produce inferior results when compared to a network with a single output. As noted by Kaastra and Boyd (1996), an ANN trains by choosing weighted links such that the average error over all the ONs is minimized. Hence, the mathematical algorithms used for the training will concentrate most of the ANN’s effort on improving the forecasts for the ON with the highest error, and limited refinement will be achieved for all other ONs. When ANNs are forecasting disinfectant concentrations for small WDSs, the number of critical nodes may be small enough to train one ANN per critical node. For larger WDSs, however, the minimum concentration may occur in several locations, and the computational needs of the several ANNs required may overcome the advantage of using ANNs, as pointed out by Broad et al. (2005). Hence, three types of ANN architectures were considered in this study: (i) $n$ ANNs with a single ON each, named ‘$n$ ANN – 1 ON’, where $n$ is the number of critical nodes; (ii) one ANN with $n$ output neurons (one per critical node), named ‘1 ANN – $n$ ON’; and (iii) one ANN with a single
ON named ‘1 ANN – 1 ON’. As opposed to ANNs using the first two architectures—whose performance is dependent of the number of critical nodes—ANNs with the last architecture were trained to estimate the minimum concentration in the network, without regard of its location.

Regarding the data used to train the ANNs, two types were considered: a uniform random sampling (i.e., a random dataset) and a non-uniform random sampling, named here as a ‘probabilistic’ dataset. A random dataset allows the training of an ANN before execution time (i.e., an offline training). However, given the tremendous size of the search space, it is unlikely that a network involving random diameters would resemble the inexpensive feasible networks that a GA would be trying to find. The ANN trained online has been found to be more accurate than the ANN trained offline (Fu and Kapelan 2010; Bi and Dandy 2012). This study presents a methodology to generate a training dataset that retains the main advantage of offline training (i.e., no additional computations during execution time). Such a training dataset contains a large number of water supply networks that resemble the type of networks analyzed by a GA after the initial iterations. Although the probabilistic dataset thus generated is still used for offline training, its networks remain similar to networks conforming to an online training dataset.
Fig 1. Typical arrangement of an artificial neural network (ANN). When applied to the design of a WDS, the input layer allocates one neuron per decision variable in the optimization problem. Similarly, the output layer commonly allocates one output neuron (ON) in the output layer that corresponds to the concentration (or pressure) at a critical node.

Methodology

This study deals with an optimization problem based on Eq. (8), where the main objective is to minimize the construction cost of a water supply network that is subject to water quality and pressure constraints. The decision variables in the problem are the diameter $D_k$ of every pipe $k$, and the disinfectant dose $C_S$ provided at a single source. Hereafter, any mention of Eq. (8) refers to such a problem. A genetic algorithm (GA) applied to solve Eq. (8) must perform an iterative execution of both a hydraulic and a WQ model to test the feasibility of every network generated by the GA. If a network proves to be unfeasible, an additional cost, in the form of a penalty, will be applied in order to prevent the GA from keeping an unfeasible network as its best solution. Pressure constraints can be tested in a small fraction of the time required to test WQ constraints,
the analysis of which will consume the majority of the GA’s total execution time. Hence, this study used ANNs to estimate WQ constraints only, whereas Epanet (Rossman 2000), the most widely used WDS modeler, was used to determine the magnitude of every pressure at every node.

The ANNs were trained as a replacement for the one-dimensional transport equation implemented by Epanet (which neglects the effects of longitudinal dispersion):

\[
\frac{\partial C}{\partial t} = RC - U \frac{\partial C}{\partial x}
\]  

(9)

where \( C(x, t) \) is the average concentration in the cross-section of a pipe, at a given position \( x \) and time \( t \); \( U \) is the average flow velocity in the pipe; and \( R \) is a reaction rate. Epanet’s WQ model uses a Lagrangian scheme to solve Eq. (9) for each pipe in the network. This scheme discretizes pipes using segments of uniform concentration and variable size that advance with a velocity \( U \) at each time step to account for the advective transport. The reaction occurring within a pipe between two time steps is also considered by applying a reaction rate \( R \) to all the segments in the pipe.

The dataset used for the training of the ANNs consists of \( N \) vectors. Each vector contains three types of element: (i) the disinfectant dose \( C_S \) at the water source, (ii) the diameter \( D_k \) of each pipe in the network, and (iii) the residual disinfectant concentration \( C_j \) at relevant nodes (as obtained by Epanet when the former elements are provided to its WQ solver). The first two types of elements \( (C_S \) and \( D_k) \) correspond to the decision variables of the optimization problem and hold the information to be provided to the input layer of the ANN (see Fig. 1). The third type of element in the training dataset \( (C_j) \)
corresponds to the WQ constraints in the optimization problem and holds the information to be provided to the output layer of the ANN (see Fig. 1).

A typical ANN is comprised of at least three layers of nodes (referred to as “neurons”) and the links between the layers of nodes, as shown in Fig. 1. The first layer is the “input layer,” the last one is the “output layer,” and all others are “hidden” layers. If the objective of the ANN is to approximate a function $f(X_1, X_2, \ldots, X_n) = (Y_1, Y_2, \ldots, Y_m)$, an independent variable $X_i, i \in [1, n]$ is assigned to each neuron in the input layer; similarly, a dependent variable $Y_j, j \in [1, m]$, is assigned to each neuron in the output layer. A weight is also assigned to each link in the ANN, and a proper manipulation of the weights’ magnitudes during the training stage will determine the accuracy of the ANN. Training the ANN involves using mathematical algorithms that modify each link’s weight in order to minimize the error obtained when trying to replicate the information provided by the training dataset. A more detailed description of how an ANN works can be found in any one of several textbooks (Hagan et al. 1996; Kartam et al. 1997; Lingireddy et al. 2005).

**Factor I. Type of artificial neural network architecture**

This study analyzes two alternatives to the customary use of the ‘$n$ ANN – 1 ON’ architecture. One alternative is the ‘1 ANN – 1 ON’ architecture consisting of one ANN with $n$ ONs (one per critical node). The main advantage of this alternative is that only a single ANN must be trained and called upon during execution. Although in theory less precision can be expected from one ANN with multiple ONs, this alternative is presented
here as a way to quantify the degree of accuracy lost when this alternative is used to find the optimal design of a WDS. The other alternative is the ‘1 ANN – 1 ON’ architecture, consisting of one ANN with a single ON. When this alternative is used, the performance (in terms of execution time) is independent of the number of critical nodes, because the ANN, is trained, with a vector of minimum concentrations—instead of a vector of concentrations at specific locations that is required by the other architectures. It should be noted, however, that this alternative is only fitted for ANNs forecasting the minimum magnitude of a constraint in Eq. (1), i.e., a minimum pressure or a minimum residual concentration. If the location of the forecasted value must also be found, as would be the case when the ANN’s estimations were being compared to experimental data (see Bowden et al. 2006), the first two options (n ANN – 1 ON and 1 ANN – n ON) should be used instead.

The location and number of critical nodes n can be determined by analyzing the WQ of a large number of random networks. Because the location of the minimum concentration may change as the sizes of the network’s pipes change, the number of critical nodes n may be so large that the calculation will become unmanageable for the ‘n ANN – 1 ON’ and ‘1 ANN – n ON’ architectures. Consequently, in this study we limited the number of critical nodes by defining such nodes as the points of water consumption where the lowest residual disinfectant could be expected to occur with a probability of 0.9 or more.
Factor II. Type of data used for the training of the artificial neural networks

ANNs are considered good interpolators but bad extrapolators, i.e., ANNs can only be expected to provide satisfactory estimations when applied to cases enclosed by the training dataset. Thus, according with Poulton (2001), much of an ANN’s success depends on a clear understanding of how to construct an appropriate dataset. Using randomly generated datasets would seem a good way to provide an ANN with widely distributed data surrounding all possible instances; however, using a random function to generate the diameter $D_k$ of each pipe in the network is very likely to produce networks that are very far from being feasible. It is therefore necessary to include the two extreme conditions (the maximum and minimum possible residual disinfectant concentrations for our case) in the training dataset, but this inclusion is insufficient if we take into account the search pattern followed by a GA. The GA usually departs from an initial population (which commonly consists of randomly generated networks) and then advances its search towards feasible solutions. This suggests that, in addition to random networks, a convenient way to train an ANN would be to include feasible networks—or at least networks that are only slightly unfeasible—in order to better replicate the search pattern followed by the GA.

A randomly generated set largely differs from a set generated by a GA searching for the solution of Eq. (8) (see Fig. 2). The variability of a network in Fig. 2 is presented here as the euclidean distance between two points in an $n$-dimensional space, where $n$ is the number of decision variables in Eq. (1). The first point is the origin of such space—null vector with $n$ dimensions—that represents the best solution found by a GA dealing with
Eq. (1). The second point is a non-null vector that represents all other feasible or non-feasible solutions to Eq. (1). For example, consider a hypothetical WDS consisting of three pipes which diameters must be selected from the two commercial sizes 304.8 mm (12 in) and 406.4 mm (16 in). If the optimal solution requires the pipes’ dimensions [16, 12, 16], then the variability of a network with pipes’ dimensions [12, 16, 16] will be equal to the distance between points \{0, 0, 0\} and \{1, -1, 0\}. It should be noted that we present this approach merely to show the dissimilarity between two sets of networks. Therefore, we made the general assumption that, in the aforementioned \(n\)-dimensional space, points close to the origin will represent networks similar to the optimal network, and that points far from the origin will represent networks largely different from the optimal network. As Fig. 2 clearly shows, a set containing a large number of randomly generated networks is far from being similar to a set containing a large number of networks analyzed by a GA.

![Histograms depicting the variability of networks for a) a random training dataset, and b) the networks tested by a GA. Both sets were obtained for the same WDS. The random set consists of 10,000 networks, and the GA-generated set consists of 100,000 networks. The x-axis represents the euclidean distance between networks in the dataset with respect to the best network found by the GA. The y-axis represents a percentage with respect to the total number of elements in each set.](image)
This study presents a methodology to obtain a large number of networks that better resemble the type of networks analyzed by a GA looking for the optimal design of a WDS. As a consequence of the reasoning followed to obtain Fig. 2, it is also valid to assume that points in an $n$-dimensional space representing feasible and nearly-feasible networks (defined here as networks that require few modifications to achieve feasibility) will be close to the origin. Therefore, if multiple minor modifications are made to the global optimum, the resulting network will likely be a feasible or a nearly-feasible network. A non-uniform probability distribution pattern can be applied to perform such minor modifications. The pattern should assign a small probability to drastic changes of pipes’ sizes, and a higher probability to moderate changes. The hypothetical WDS—which consists of three pipes and has two commercial diameters of 12 in and 16 in—can again be used to illustrate this concept. Assuming that the global optimum has pipes with diameters of [16, 12, 16], and that the non-uniform probability distribution designates a probability of 0.7 to diameters coinciding to the global optimum, then the nearly-feasible networks generated using this approach will be assigned accordingly: for the first and third pipes, diameters of 16 in and 12 in with probabilities 0.7 and 0.3, respectively; for the second pipe, diameters of 12 in and 16 in with probabilities 0.7 and 0.3, respectively.

Clearly, this approach seems impractical because finding the global optimum is the final objective of the optimization problem in Eq. (1). However, a similar network can be used instead, and the resulting networks are still expected to be feasible or nearly-feasible. This network then becomes a point of reference, and is named a reference...
network or ‘RefNet’. In this study, we present a methodology for quickly finding a solution that approximates the optimal design. The methodology is based on a post-optimization heuristic technique proposed by Andrade et al. (2013). Known as a Greedy algorithm, this technique involves the application of a decision rule that, at each iteration, decreases the diameter of one pipe by one commercially available size. Here, provided in pseudo-code, is a detailed description of this algorithm:

Pre-processing algorithm (to reduce the computational effort required by the ensuing Greedy algorithm)

1. Set the diameters of all pipes to the minimum commercial size available
2. Repeat for all pipes with a velocity above the limit $Vel\_limit$
   a. Increase the diameter of the pipe in one dimension
3. Return to Step 2 until all pipes have a velocity below $Vel\_limit$
4. Move to Step 5 if pressures constraints plus an additional pressure $Add\_pres$ are met for all nodes in the network; otherwise reduce the magnitude of the velocity limit $Vel\_limit$ and return to Step 2
5. Set the concentration at the reservoir $C_S$ to the minimum magnitude in the group of disinfection doses $S_C$
6. Perform a WQ analysis with the resulting diameters and the disinfectant dose $C_S$
7. Stop if the residual disinfectant $C_j$ at all nodes is above the minimum allowable concentration $C_{min}$; otherwise set $C_S$ to the immediately higher magnitude in the group $S_C$ and return to Step 6
Greedy algorithm (to find a feasible solution to Eq. (1), one that is reasonably close to optimal, and do so in a fraction of the time required by an adaptive search method; the resulting network will be used as the ‘RefNet’)

1. Start with the diameters’ dimensions $D_k$ and concentration at the reservoir $C_S$ obtained from the pre-processing
2. Create a set $\alpha$ containing all the pipes which diameter can be reduced by one commercial diameter without violating pressure constraints in Eq. (8).
3. Sort all pipes in set $\alpha$ based on a decision value $V_k$ that takes into account two factors: the pressure changes occurring in the network after modifying the diameter $D_k$ of pipe $k$; and the cost savings achieved after such modification.
4. Perform a WQ analysis for each of the ‘$nWQruns$’ pipes ranked at the top of set $\alpha$ (note that the number of pipes’ modifications to be tested for WQ will largely determine the execution time required by the Greedy algorithm, and it is therefore recommended to make $nWQruns$ smaller than ten).
5. Reduce the diameter of any pipe the modification of which yields the highest residual disinfectant at the node with the minimum concentration, i.e., at the critical node.
6. Continue to Step 7 if set $\alpha$ contains a single element; otherwise return to Step 2
7. Reduce the concentration $C_S$ at the reservoir to the immediately smaller magnitude in the group of available disinfectant doses $S_C$.
8. Perform a WQ analysis with the resulting diameters and the disinfectant dose $C_S$.
9. Return to Step 7 if the residual disinfectant $C_j$ at all nodes is above the minimum allowable concentration $C_{min}$; otherwise set $C_S$ to the immediately higher magnitude in the group $S_C$ and stop the algorithm.

The Greedy algorithm described above depends on a decision rule that measures the relative merit of a candidate pipe $k$ in comparison with all other elements in set $\alpha$. 
According to step 3, this decision rule can be based on the tradeoff between (i) the cost reduction obtained, and (ii) the pressure decline at the critical node (caused by decreasing the diameter of $k$). A definition of such a rule can be achieved by assigning relative weights to each candidate pipe $k \in \alpha$ in such a way that the pipe size that affords the greatest savings at the least reduction in pressure at the critical node receives the highest decision value $V_k$. Hence, a decision rule can be defined as follows:

$$V_k = m_{CS} \frac{CS_k - CS_{min}}{CS_{max} - CS_{min}} + m_P \frac{P_k - P_{min}}{P_{max} - P_{min}}$$

(10)

where $m_{CS} \in [0, 1]$ is a multiplier that determines the importance given to the cost savings; $CS_k$ are the savings obtained by reducing the diameter of pipe $k$; $CS_{min}$ and $CS_{max}$ are, respectively, the minimum and maximum savings obtained after modifying any pipe in set $\alpha$; $m_P = 1 - m_{CS}$ is a multiplier that determines the importance given to the pressure decline; $P_k$ is the pressure at the critical node after the modification of pipe $k$; $P_{max}$ and $P_{min}$ are, respectively, those maximum and minimum pressures at the critical node that can result from a reduction of any pipe in set $\alpha$. More details can be found in Andrade et al. (2013). It should be noted that, although the critical nodes for the peak and the fire flow may be different, $P_{min}$ in Eq. (10) is defined as:

$$P_{min} = \begin{cases} p_{min}^p, & \text{if } (p_{min}^p - p_{lim}^p) \leq (p_{min}^f - p_{lim}^f) \\ p_{min}^f, & \text{if } (p_{min}^p - p_{lim}^p) > (p_{min}^f - p_{lim}^f) \end{cases}$$

where $p_{min}^p$ and $p_{min}^f$ are the minimum nodal pressures after the modification of any pipe in set $\alpha$, as obtained for the peak and fire flow, respectively; and $p_{lim}^p$ and $p_{lim}^f$ are the minimum allowable pressures for the peak and the fire flow, respectively.
Summary of the analysis

This study presents the analysis of three factors that affect the performance of ANNs used to assist the optimal design of WDSs. For the first factor, three types of ANN architecture were analyzed: the conventional approach (\( n \) ANN – 1 ON), and two alternatives (1 ANN – \( n \) ON and 1 ANN – 1 ON). For the second factor, two types of training datasets were considered: the conventional approach (random dataset generated with uniform sampling) and one alternative (probabilistic dataset generated with non-uniform sampling). This study also accounts for the interaction that occurs between the type of ANN architecture and the type of training dataset (see Fig. 3).

The third factor analyzed in this study quantifies the impact that the size of the network has on the performance of ANNs. The proposed methodology was applied first to a small WDS consisting of a few dozen pipes, and the relevant types of ANNs found in this stage were later tested on a large WDS consisting of more than a thousand pipes in order to quantify the effect that be produced by this number of decision variables. In both of the following cases studies, a GA using Epanet’s WQ model was executed, and all the networks that were analyzed by the GA were stored, as were as the magnitude of the minimum concentrations for each network (as estimated by Epanet). The dataset thus generated will be served to determine the accuracy of each type of ANN.
Fig. 3. Hierarchy chart with the six types of ANNs applied to forecast WQ constraints of two WDS. ANNs of type 1 have been used by previous studies relying on an offline training of neural networks. The other five types are presented here to analyze their impact on the accuracy of ANNs.

Parallel processing

The overall objective when applying neural networks to assist a heuristic method dealing with Eq. (1) is to reduce the overall time required by the process. Alternative methods can be used to further expedite the solving of the optimization problem (Roshani and Filion 2012; Wu and Behandish 2012). One such alternative involves using more than one computer processor when performing the most time-consuming activities, such as the generation of a large dataset required when training an ANN. This study applied the parallel processing toolbox available in the Matlab mathematical software.
(Mathworks 2012a) to (i) generate the training datasets, to (ii) execute (using Epanet’s WQ model) the GA that provided the networks to be used as a validation dataset, and to (iii) call upon the ANNs to estimate the minimum concentrations in the validation dataset.

**Case Study 1, the Hanoi Network**

A genetic algorithm provided the validation dataset that was used to minimize the construction cost of the water supply network depicted in Fig. 4. Fujiwara and Khang (1990) introduced this WDS—known as the Hanoi network. It consists of 34 pipes, 31 nodes, and a single water source. The network has been extensively analyzed (Savic and Walters 1997; Geem 2006) in an effort to find the design of lowest cost among all the feasible solutions. The cost per unit length $U_{ck}$ of a pipe $k$ in this network is a function of the pipe’s diameter $D_k$ and is determined with regard to the following relation:

$$U_{ck} = 1.1D_k^{1.5}.$$ 

The original optimization problem for the Hanoi network has been modified here to account for WQ constraints, so the decision variables are the set of diameters $\{D_k\}$, $k \in [1, 34]$, and the disinfectant (chlorine) dose $C_S$ provided at the single water source. A single demand, with a total magnitude of 19,940 m$^3$/h, was tested by creating single-period hydraulic simulations to ensure that the pressure $P_j$ at all nodes was above the minimum limit of 30 m (42.7 psi) when the head at the water source was 100 m (142 psi). The network was also tested by creating extended-period WQ simulations to
ensure that the residual chlorine \( C_j \) at all nodes was above the minimum limit of 0.4 mg/l.

The mathematical formulation of the problem is then:

\[
\text{Minimize: } \sum_{k=1}^{n} (U_c \cdot L_k), \quad k \in [1, 34]
\]

Subject to: (i) \( P_j \geq 30 \text{ m} \); (ii) \( C_j \geq 0.4 \text{ mg/l}, j \in [1, 31] \);

(iii) \( D_k \in S_D \); and (iv) \( C_S \in S_C \)

where \( n \) is the number of pipes in the network; \( S_D = \{304.8 \text{ mm (12 in)}, 406.4 \text{ mm (16 in)}, 508 \text{ mm (20 in)}, 609.6 \text{ mm (24 in)}, 762 \text{ mm (30 in)}, 1016 \text{ mm (40 in)}\} \) is the set of commercial dimensions from which the diameter of each pipe must be selected; and \( S_C = \{1, 1.1, \ldots, 1.5\} \text{ mg/l} \) is the discrete set of disinfectant doses from which the concentration at the water source \( C_S \) must be chosen.

The WQ modeling of the Hanoi network was performed with Epanet and required extended-period simulations with a total duration of 55 h. Hydraulic time steps of 1 h and WQ time steps of 5 min were used to discretize the simulation time. The network experiences a changing demand pattern that resembles the varying conditions expected during a WQ analysis. This pattern changes all nodal demands every hour and was obtained after multiplying, by a constant factor of 0.4065, each element in the demand pattern found in the Epanet Example network 2 (Net2.net). The network’s total demand thus varies between a maximum of 19,940 m\(^3\)/h (peak demand tested by the hydraulic simulations) and a minimum of 2,991 m\(^3\)/h. The decay of the chlorine injected at the water source was modeled assuming a first-order decay. A constant coefficient of 1.0 days\(^{-1}\) was assumed to be the reaction with the bulk flow, and the reaction with the pipe wall was assumed to occur at a constant rate of 0.5 days\(^{-1}\).
To search for the optimal solution to Eq. (11), this study applied the genetic algorithm implemented in the global optimization toolbox of the numerical computing software Matlab (Mathworks 2012b). The best solution found by the GA (shown in Fig. 4) corresponds to the decision variables—concentration at the water source and diameters for each pipe—conforming to the cheapest feasible network among all networks considered during three rounds of GA runs. A round comprises ten GA runs, and one validation dataset was generated during each round. The validation dataset for a round consists of a million vectors containing the decision variables, as well as the minimum residual chlorine $C_{\text{min}}$ obtained by Epanet’s WQ solver for each vector. The size of the validation dataset was determined by ten GA runs and 100,000 function evaluations performed by a GA run (200 generations with a population of 500 each). Both the number of generations and the population size were selected after preliminary runs that also determined the values for the crossover fraction (0.6) and the elite count (10%). The first two rounds found the same best solution (which cost US$6.216*10^6 and required a minimum chlorine dose $C_S=1.2$ mg/l to guarantee feasibility in terms of water quality). The last round found the best overall solution (shown in Fig. 4). Its estimated cost was US$6.151*10^6, and it required a minimum dose $C_S=1.1$ mg/l to achieve feasibility.
Fig. 4. Hanoi water supply network used for the case study. The minimum residual chlorine occurred at the three nodes enclosed by circles with a probability of 0.9. The pipes’ dimensions correspond to the best diameters found by three rounds of ten GA runs.

Results for Case Study 1, the Hanoi Network

Given that ANNs here analyzed are intended to assist heuristic optimization methods, a set of networks generated by a GA provides an appropriate validation dataset, one that clearly represents the range of inputs provided to the ANNs during execution. The collection of the elements in the validation dataset was achieved by means of a computer program that stored 500 vectors—corresponding to the population size—after each of the 200 generations analyzed by the GA. Each of these vectors contained 36 elements: (i) the diameters of the 34 pipes in the Hanoi network, (ii) the concentration $C_S$ at the reservoir, and (iii) the minimum residual concentration $C_{min}$ obtained by Epanet when the former elements were provided as input parameters. A histogram depicting the variability of the
diameters proposed by the best GA run (the one achieving the solution with the least cost) is presented in Fig. 2b.

A random and a probabilistic dataset were created to train the ANNs that were to be compared. Both datasets contained ten thousand sets of vectors. Similar to the vectors in the validation dataset, each of these vectors contained different magnitudes for all the decision variables in the optimization problem. However, the training data includes the residual chlorine levels at all the critical nodes in the network, as opposed to the validation data that includes only the lowest concentration $C_{min}$. As mentioned before, critical nodes are defined here as the nodes where $C_{min}$ is expected to occur with a probability of 0.9. Fig. 4 shows the location of the critical nodes that were obtained by analyzing ten thousand random networks. It should be noted, however, that when this definition was used, only three nodes were found to be critical, although $C_{min}$ also occurred at other eleven nodes.

The reference network ‘RefNet’ obtained by the greedy algorithm had a total cost of US$6.309*10^6, and this figure was only 2.56% above the cost of the best solution found by the GA. The greedy algorithm was executed iteratively with cost savings $m_{CS}$ and pressure $m_P$ multipliers—both of which are required in order to estimate the decision value $V_k$ in Eq. (10)—changing at intervals of 0.1. The best network was obtained after 311 sec, using multipliers $m_{CS}=0.38$ and $m_P=0.62$. Such a solution is feasible in terms of water quality if the chlorine concentration $C_S$ at the reservoir is equal to or higher than 1.2 mg/l. To reduce the computation time, each iteration of the greedy algorithm only performed analyses ($nWQruns$) for the four best-ranked pipes. The probabilistic training
The training of six types of ANNs shown in Fig. 3 was performed using the neural network toolbox implemented by the Matlab software (Mathworks 2012c). 70% of each training dataset (random and probabilistic) was used for the training, 15% for validation, and 15% for testing. The neural networks are three-layered feed-forward networks (as shown in Fig. 1) made up of sigmoid hidden neurons and linear output neurons. The input layer contained 35 neurons, one for the concentration at the water source and 34 for the pipes’ diameters, whereas the hidden layer contained 20 neurons. The Levenberg-Marquardt backpropagation algorithm was applied to modify the link’s weights in order to minimize the error obtained when forecasting the information provided by the training dataset. All of the ANNs shared these parameters that were determined by preliminary
runs. It should be noted that the main objective of this study is a comparison between ANNs, and thus the same parameters were used for all ANNs. Ten repetitions were performed for each ANN. The ANN with the lowest Mean Squared Error (MSE)—with respect to the 15% of the training data assigned for validation purposes—was the only neural network used to estimate the minimum concentration of the networks in the validation dataset.

Table 1. Non-uniform probability distribution used to assign the pipes’ sizes for the networks in the probabilistic training dataset

<table>
<thead>
<tr>
<th>Pipe diameter in ‘RefNet’ (mm)</th>
<th>Probability of being assigned the beneath pipe diameter (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>304.8</td>
<td>0.4 0.3 0.2 0.05 0.03 0.02</td>
</tr>
<tr>
<td>406.4</td>
<td>0.25 0.4 0.25 0.05 0.03 0.02</td>
</tr>
<tr>
<td>508</td>
<td>0.05 0.25 0.4 0.25 0.03 0.02</td>
</tr>
<tr>
<td>609.6</td>
<td>0.02 0.05 0.25 0.4 0.25 0.03</td>
</tr>
<tr>
<td>762</td>
<td>0.02 0.03 0.05 0.25 0.4 0.25</td>
</tr>
<tr>
<td>1016</td>
<td>0.02 0.03 0.05 0.2 0.3 0.4</td>
</tr>
</tbody>
</table>

**Fig. 5.** Histogram depicting the variability of the networks in the probabilistic training dataset. The x-axis represents the euclidean distance between the networks in the dataset with respect to the best network found by the GA. The y-axis represents a percentage with respect to the total number of elements in the dataset.
Three rounds of training and comparing of neural networks were performed, a total that coincided with the number of validation datasets obtained by the GA. Hence, each round applied the six ANN-types shown in Fig. 3 to forecast the minimum residual concentrations contained in its respective validation data. The Mean Squared Error (MSE) of each ANN-type defines its accuracy as follows:

\[
MSE = \sum_{i=1}^{10^6} (C_{\text{min},i}^{\text{EPA}} - C_{\text{min},i}^{\text{ANN}})^2
\]

where \(C_{\text{min},i}^{\text{EPA}}\) is the minimum concentration obtained by Epanet for the \(i\)-th vector in the validation dataset; and \(C_{\text{min},i}^{\text{ANN}}\) is the approximation to \(C_{\text{min},i}^{\text{EPA}}\) forecasted by the neural network. It should be noted that, for the ‘3 ANN – 1 ON’ architecture, \(C_{\text{min},i}^{\text{ANN}}\) corresponds to the minimum concentration among the three values forecasted by the three neural networks with a single output neuron per critical node. For the ‘1 ANN – 3 ON’ architecture, \(C_{\text{min},i}^{\text{ANN}}\) corresponds to the minimum concentration among the three values forecasted by a single neural network with an output neuron per critical node.

Fig. 6 presents a comparison between all the ANN-types. For all three rounds, the lowest MSE was achieved for the ANN-type trained with the ‘1 ANN – 1 ON’ architecture, and a probabilistic dataset. ANNs trained with a probabilistic dataset also showed a better accuracy than those trained with a random dataset. For only one out of the eighteen comparisons (six per round) the random dataset showed a lower MSE than the probabilistic dataset. The comparison of the ANN-types ‘3 ANN – 1 ON’ (denoted by circles) and the ‘1 ANN – 3 ON’ (denoted by squares) provided an unexpected result: for the random data (straight lines), the former ANN-type only surpassed the accuracy of the
latter during the third round. For the probabilistic data (dashed lines), the ‘1 ANN – 3 ON’ proved superior in all rounds. This contradicts the notion that neural networks with multiple output neurons will produce inferior results (Kaastra and Boyd 1996).

Fig. 6. ANN accuracy—measured as the Mean Squared Error (MSE)—and average execution time.

In terms of execution time, the GA searching for the best solution to Eq. (11) required an average of 52,812 sec to perform a round (each round consisting of ten GA runs). That approximately 96% of this time is consumed by the WQ solver supports the contention that artificial neural networks could significantly benefit WQ engineers. With respect to the time the ANNs required to estimate the minimum concentrations for each
GA round, the main factor ruling the execution time of a neural network type was the use of either one or three neural networks. ANN-types ‘1 ANN – 3 ON’ required an average of 9,230 sec to estimate the lowest chlorine level of the $10^6$ elements generated by a GA run. On the other hand, ANN-types ‘3 ANN – 1 ON’ required an average of 25,729 sec, being thus proportional to the number of neural networks required to forecast the lowest chlorine level for each element in the validation dataset. All of these calculations were performed with a personal computer (PC) with two Intel(R) Xeon(R) processors (2.4 GHz) and 23 GB of random access memory (RAM). The parallel processing toolbox available in the Matlab mathematical software (Mathworks 2012a) was applied to reduce the computation time required for the most consuming tasks, such as running Epanet’s WQ solver to obtain the datasets used for the training of ANNs.

Case Study 2. A Realistic Water Distribution System

The WDS used for the second Case Study is a modified version of a network analyzed by Kang and Lansey (2011). The network consists of a reservoir and 1090 pipes transporting water to 517 points of demand. The optimization problem for this system requires the determination of the pipes’ diameters and the minimum chlorine dose required to achieve the lowest construction cost, subject to pressure and water quality constraints, as follows:
Minimize: \( \sum_{k=1}^{n} (Uc_k \cdot L_k) \), \( k \in [1, 1090] \)

Subject to: (i) \( P_j^p \geq 28.122 \text{ m (40 psi)} \); (ii) \( P_j^f \geq 14.061 \text{ m (20 psi)} \);
(iii) \( P_j^a \geq 28.122 \text{ m (40 psi)} \); (iv) \( C_j \geq 0.4 \text{ mg/l, } j \in [1, 517] \);
(v) \( D_k \in S_D \); and (vi) \( C_S \in S_C \)

where \( P_j^p \), \( P_j^f \), and \( P_j^a \) are the minimum nodal pressures to be provided at each node \( j \) for the peak, fire, and average flow conditions, respectively; \( C_j \) is the minimum concentration to be provided for an extended period simulation; \( S_D = \{152.4 \text{ mm (6 in), 203.2 mm (8 in), 254 mm (10 in), 304.8 mm (12 in), 406.4 mm (16 in), and 508 mm (20 in)} \} \) is the set of commercial dimensions from which the diameter of each pipe must be selected; \( S_C = \{1, 1.1, \ldots, 4.0\} \text{ mg/l} \) is the discrete set of disinfectant doses from which the concentration at the water source \( C_S \) must be chosen.

To estimate the pipe-network construction cost \( Uc_k \), Kang and Lansey (2011) applied the cost function proposed by Clark et al. (2002):
\[
y = a + b(x^c) + d(u^e) + f(xu) \tag{7}
\]
where \( y \) is the cost of a particular component ($/ft, 1 \text{ ft} = 0.3048 \text{ m}); x \) is a design parameter (for example, pipe diameter in inches, 1 inch = 2.54 cm); \( u \) is an indicator variable (for example, class of pipe); and \( a, b, c, d, e, \) and \( f \), are parameter values estimated using regression techniques. The total construction cost of a pipeline per foot can be obtained as the sum of the outcomes of Eq. \( (7) \) for each of the following components: (i) pipe material, (ii) trenching and excavation, (iii) embedment, (iv) backfill and compaction, and (v) valve, fitting and hydrant cost. The same parameters applied by Kang and Lansey (2011) were used in this study.
Fig. 7. Realistic water distribution system used for the second case study. The lines in the figure indicate the diameter’s dimensions obtained with the Greedy Algorithm. A single reservoir located at the northeast corner provides water to the system. A preliminary analysis showed that the minimum residual concentration is likely to occur in one of the nine nodes identified by numbers. The lower the number, the higher the probability that that node will be the one with the minimum concentration.

The WQ modeling of this network was performed with Epanet and required extended-period simulations with a total duration of 288 h. Hydraulic time steps of 1 h and WQ time steps of 1 min were used to discretize the simulation time. The network experienced a changing demand pattern that resembled the varying conditions expected during a WQ analysis. The average and the peak demands were assigned magnitudes of 143.4 l/s (2,273 gpm) and 251 l/s (3977 gpm), respectively; whereas the fire demand had
a magnitude of 276.27 l/s (4379 gpm), with the critical node 1 in Fig. 7 being assigned an extraordinary demand of 75.7 l/s (1200 gpm). The decay of the chlorine injected at the water source was assumed to be a first-order decay. A constant coefficient of 0.2 days\(^{-1}\) was assumed for the reaction with the bulk flow, and the reaction with the pipe wall was assumed to occur at a constant rate of 0.05 days\(^{-1}\). More details about this network can be found in Kang and Lansey (2011).

**Results for Case Study 2, A Realistic Water Supply Network**

The minimum residual chlorine obtained for ten thousand random networks was analyzed to determine the critical nodes for this network. The minimum concentration occurred 90% of the time at the nine nodes shown in Fig. 7, whereas the other 10% of the events occurred at 45 other nodes. The best solution to Eq. (6) found by the Greedy Algorithm (shown in Fig. 7) has a capital cost of US$31,369,490 and requires a chlorine dose at the reservoir of 2.2 mg/L. The Greedy Algorithm used a pressure multiplier \(m_P=0.4\) and a savings multiplier \(m_{CS}=0.6\) for Eq. (3). The Greedy Algorithm required a total execution time of 14,146 seconds (3.93 hours). In contrast, a GA using Epanet to estimate WQ constraints—with 500 networks as a population and with 500 generations—could only achieve a cost of US$33,757,059 after 929,314 seconds (258 hours or 10.75 days). Other parameters used for this GA run, and defined after preliminary runs, were 0.6 as the crossover fraction and 10% as the elite count. Although several other GA runs were made with either a smaller population or a lower number of generations, the GA...
could not find a better solution than the one achieved by the Greedy Algorithm. This example thus shows the limitations of a GA dealing with computationally intensive constraints.

Two datasets of 100,000 networks each were generated for the training of the ANNs. To generate the probabilistic training dataset, the non-uniform probability distribution shown in Table 2 was repeatedly applied to the ‘RefNet’ found by the Greedy Algorithm. Parallel processing was used to generate both datasets. The random and the probabilistic datasets required 114.86 hours (4.78 days) and 88.46 hours (3.68 days) to be completed. When no parallel processing was used, the random dataset required 26.2 days to be completed, which clearly shows the advantage of using several processors. Three ANNs were trained for each of the six ANN-types analyzed for the second case study, and only the best ANN among these three was kept to be used later. For the ‘9 ANN – 1 ON’ type, three ANNs were also trained for each critical node, giving a total of 27 ANNs trained for this case. Table 3 presents the training time required by each ANN-type and shows the disadvantage of using the ‘9 ANN – 1 ON’ architecture instead of the two other architectures.

The training of the ANNs was performed using the neural network toolbox implemented by the Matlab software (Mathworks 2012c). All the neural networks trained shared the following parameters: 70% of the training dataset was used for the training, 15% for validation, and 15% for testing. The neural networks are three-layered feed-forward networks made up of sigmoid hidden neurons and linear output neurons. The input layer contained 1,090 neurons, one for the concentration at the water source and
1090 for the pipes’ diameters, whereas the hidden layer contained 35 neurons. The Levenberg-Marquardt backpropagation algorithm was discarded due to its poor performance (Mathworks 2012a) when applied to neural networks with a large number of neurons, and the scale conjugate gradient was used instead.

The set of 250,000 networks analyzed by the GA (resulting from 500 population times 500 generations), and the minimum concentration estimated by Epanet for each network was used to determine the execution time and accuracy of each ANN-type. Table 3 shows the results of this comparison. ANNs trained with a probabilistic dataset performed better than ANNs trained with a random dataset. Remarkably, the percentage of accurate classifications, i.e., networks that the ANN correctly identified as either feasible or unfeasible, had a minimum magnitude of 42.93% for the probabilistic dataset, a percentage that is still 32.5% better than the best case found by the ANNs trained with a random dataset. As Fig. 8 shows, using a random dataset may be convenient during the initial stages of a GA run, but as the number of iterations advances, a better performance can be expected from an ANN trained with a dataset that includes feasible and nearly-feasible networks, like the probabilistic dataset here proposed.

The impact of the network’s size (see Table 3) is also reflected in the MSE that, even in the best case, is more than double the MSE of approximately 0.003 that prevailed on most ANNs trained for the Hanoi network. Another noteworthy result is the confirmation that the ‘1 ANN – 9 ON’ architecture shows no significant disadvantage in terms of accuracy when compared with the ‘9 ANN – 1 ON’ that was expected to be more
accurate. Furthermore, the ‘1 ANN – 9 ON’ achieved the best overall result, producing accurate classifications for 90% of the networks in the validation dataset.

![Cumulative misclassifications made by ANNs forecasting minimum WQ levels of networks in the validation dataset of Case Study 2.](image)

**Fig. 8.** Cumulative misclassifications made by ANNs forecasting minimum WQ levels of networks in the validation dataset of Case Study 2. Each misclassification accounts for either a false positive or a false negative. A false positive indicates that the ANNs estimated wrongly that an unfeasible network (according with EPANET’s water quality model) was feasible. Similarly, a false negative indicates that the ANNs estimated wrongly that a feasible network was unfeasible.

<table>
<thead>
<tr>
<th>Pipe diameter in ‘RefNet’ (mm)</th>
<th>Probability of being assigned beneath pipe diameter (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>152.4</td>
</tr>
<tr>
<td>152.4</td>
<td>0.4</td>
</tr>
<tr>
<td>203.2</td>
<td>0.25</td>
</tr>
<tr>
<td>254</td>
<td>0.05</td>
</tr>
<tr>
<td>304.8</td>
<td>0.02</td>
</tr>
<tr>
<td>406.4</td>
<td>0.02</td>
</tr>
<tr>
<td>508</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 2. Non-uniform probability distribution applied to the assigning of pipe sizes for networks in the probabilistic training dataset for Case Study 2.
Table 3. Performance of ANNs applied to Case Study 2

<table>
<thead>
<tr>
<th>Type of selection of diameters</th>
<th>Architecture of ANN</th>
<th>MSE</th>
<th>Training time (seconds)</th>
<th>Execution time (seconds)</th>
<th>Accurate classifications (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>1 ANN – 9 ON</td>
<td>0.09155</td>
<td>2,450</td>
<td>346</td>
<td>10.54</td>
</tr>
<tr>
<td></td>
<td>9 ANN – 1 ON</td>
<td>0.08728</td>
<td>26,504</td>
<td>2,950</td>
<td>10.63</td>
</tr>
<tr>
<td></td>
<td>1 ANN – 1 ON</td>
<td>0.0358</td>
<td>2,617</td>
<td>338</td>
<td>10.33</td>
</tr>
<tr>
<td>Probabilistic</td>
<td>1 ANN – 9 ON</td>
<td>0.00827</td>
<td>1,923</td>
<td>338</td>
<td>90.54</td>
</tr>
<tr>
<td></td>
<td>9 ANN – 1 ON</td>
<td>0.04928</td>
<td>25,767</td>
<td>2,932</td>
<td>42.93</td>
</tr>
<tr>
<td></td>
<td>1 ANN – 1 ON</td>
<td>0.03825</td>
<td>2,982</td>
<td>337</td>
<td>57.01</td>
</tr>
</tbody>
</table>

Note: Execution time is the time required by the ANNs to estimate the minimum water quality for each of the 250,000 (500 populations and 500 generations) networks analyzed by the GA; any correct identification of a feasible or unfeasible network is considered an accurate classification.

Conclusions

This work examines three factors that affect the performance of ANNs assisting the optimal design of a WDS. A set of networks was tested using a genetic algorithm that deals with pressure and water quality constraints in order to create the validation dataset that was used to compare the impacts of each factor. With respect to the first factor (type of architecture selected for the ANNs), we found that using a single ANN with one output neuron per critical node should not be discarded a priori. However, in light of the findings of this study, we recommend that preliminary runs be conducted to determine the convenience of using such architecture for particular applications. With respect to the second factor (type of data used for the training of ANNs), the probabilistic dataset we propose requires finding a fast approximation to the optimal solution. We also propose a methodology for finding such a nearly-optimal network and for generating the probabilistic dataset based on a non-uniform probability distribution. In both case studies...
we presented here, the ANNs trained with a probabilistic dataset outperformed the ANNs trained with a random dataset, but the advantages of using a probabilistic dataset become more evident when ANNs trained with them were applied to the large network presented in the second case study. This study showed that the third factor (the network’s size) has a negative impact in the accuracy of ANNs. The results presented here showed that increasing the number of decision variables (and as a consequence the number of input neurons in the ANNs) has a major impact on the accuracy of ANNs. Similarly, the results presented for the first case study suggested that a probabilistic dataset could be a better option than a random dataset. However, a conclusive confirmation of this statement was only possible after the second Case Study was analyzed. Therefore, we suggest that future enhancements of ANNs applied to the optimal design of a WDS include a realistic WDS, like the one presented here.

Acknowledgements

This material is based in part upon work supported by the National Science Foundation under Grant No. 083590. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.
References


