

Research Article

On the Complexities of the Design of Water Distribution Networks

Joaquín Izquierdo,¹ Idel Montalvo,² Rafael Pérez-García,¹
and Agustín Matías³

¹ FluIng-IMM, Universitat Politècnica de València, Camino Vera s/n, 46022 Valencia, Spain

² 3S Consult Büro Karlsruhe, Albtalstrasse 13D, 76137 Karlsruhe, Germany

³ Universidad de Extremadura, Avda. Universidad, s/n, 10071 Cáceres, Spain

Correspondence should be addressed to Joaquín Izquierdo, jizquier@upv.es

Received 10 July 2011; Accepted 13 September 2011

Academic Editor: Zidong Wang

Copyright © 2012 Joaquín Izquierdo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Water supply is one of the most recognizable and important public services contributing to quality of life. Water distribution networks (WDNs) are extremely complex assets. A number of complex tasks, such as design, planning, operation, maintenance, and management, are inherently associated with such networks. In this paper, we focus on the design of a WDN, which is a wide and open problem in hydraulic engineering. This problem is a large-scale combinatorial, nonlinear, nonconvex, multiobjective optimization problem, involving various types of decision variables and many complex implicit constraints. To handle this problem, we provide a synergetic association between swarm intelligence and multiagent systems where human interaction is also enabled. This results in a powerful collaborative system for finding solutions to such a complex hydraulic engineering problem. All the ingredients have been integrated into a software tool that has also been shown to efficiently solve problems from other engineering fields.

1. Introduction

Water distribution networks (WDNs) are important and dynamic systems. Most people are unaware of their importance, despite the fact that they routinely open the water tap every day. The existence of WDNs is generally ignored, except when disruption occurs. Politicians tend to neglect them because they are buried assets. Nevertheless, WDNs provide citizens with an essential public service. They supply drinking water, which is a crucial requirement for the normal development of most basic activities of life, such as feeding and hygiene [1].

For most of the people, perhaps with the exception of those directly involved in their management, WDNs are simply static infrastructures. However, WDNs are dynamic

and living beings. They are born, when they are designed and built; they grow, as a consequence of increasing urban development and the appearance of new demands; they age and deteriorate, since they suffer a number of operational and environmental conditions that cause progressive and insidious deterioration; they need care, preventive care but also sometimes surgery; they are expected to work properly, despite the great amount of uncertainty involved and defective information about the state and operation of these systems due to their geographical dispersion and the fact that they are hidden assets; they have to meet basic requirements even under adverse circumstances, hence they must show resilience; and so on.

Our aim is to achieve a quality long-lasting life for WDNs. Nevertheless, it is a fact that with the passing of time these systems become gradually, but substantially, impaired. Some of the reasons include increasing loss of pressure triggered by increasing roughness of the inner pipes; breakage or cracking of pipes provoked by corrosion and mechanical or thermal charges; loss of water (leaks) and pathogen intrusion due to pipe breaks and cracks, with their corresponding economic loss, third party damage, and risk of contamination.

Several complex tasks, such as design, planning, operation, maintenance, and management, are inherently associated with WDNs. These tasks are not simple at all and require considerable investment. Efficiency and reduction of costs have been compelling reasons for practitioners to progressively move away from manual design based on experience and support the development of suitable automatic or semiautomatic tools. Support, of course, is increasingly multidisciplinary and receives contributions from various scientific areas. Mathematics is one of the areas contributing most effectively with suitable and efficient methods and tools.

We focus on the design of a WDN, a wide and open problem in hydraulic engineering that involves the addition of new elements in a system; the rehabilitation or replacement of existing elements; decision making on operation; reliability and protection of the system; among other actions. Designs are necessary to carry out new configurations or enlarge and improve existing configurations to meet new conditions. The design of a WDN involves finding acceptable trade-offs among various conflicting objectives: finding the lowest costs for layout and sizing using new components and rehabilitating, reusing, or substituting existing components; creating a working system configuration that fulfils all water demands, including water quality; adhering to the design constraints; guaranteeing a certain degree of reliability for the system [2–4].

The formulation of the optimal WDN design problem has traditionally been influenced by the available mathematical support for solving the problem; formulations have been adapted (restricted and/or simplified) to the available mathematical techniques [5]. However, in its more general setting, the optimal design of a large WDN is a formidable problem because of the very high computational complexity involved if it has to be solved within a reasonable time framework. There are various explanations for this complexity. One reason is the huge number of expensive hydraulic analyses that must be performed during the process. In addition, WDN optimal design is a nonlinear, nonconvex problem that cannot be formulated in just one way and has various objective functions because of the plurality of situations that can be found and the different aims that each situation may involve. Last but not least, even for the simplest cases and the smallest of WDNs, design can easily become an NP-hard problem.

In this paper, we provide a detailed description of the various objectives involved in the WDN optimal design problem. We argue that classical optimization techniques are unable to satisfactorily solve these problems and describe an evolutionary, multiagent approach

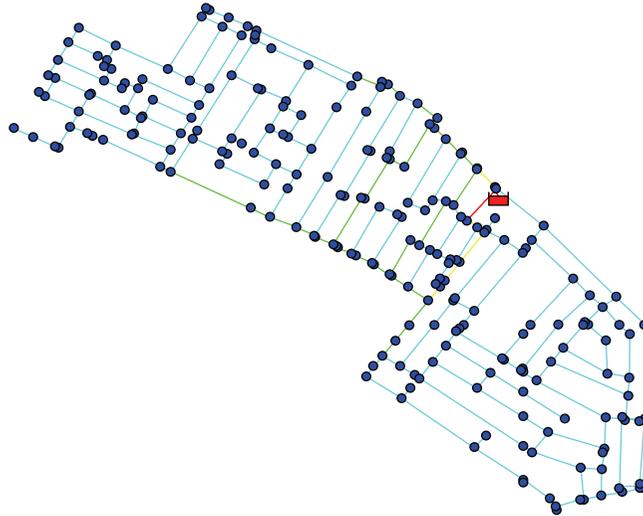


Figure 1: A WDN.

to tackle this task. We then provide the solution for some real-world water distribution networks.

2. The WDN Optimal Design Problem

The layout of a network is usually conditioned by urban plans and/or various other reasons. Street positions, location of sources, main consumption points, and so forth are perfectly known before starting the design. Therefore, design does not generally consider the layout and is usually restricted to sizing the necessary network components. Figure 1 presents a network fed by one tank (small red box), with 294 pipes (lines) amounting to 18.34 km and 240 nodes (blue points). This network has already been designed, and the different colors (light blue, green, yellow, and red) represent different pipe diameters.

Various objectives may be considered in the WDN optimal design problem. In this section, we describe these objectives, namely, cost of components; adherence to hydraulic constraints; satisfaction of water demand quality; resilience of the system during stressed conditions.

2.1. Cost of Components

Apart from the basic variables of the problem, which are the diameters of the new pipes, additional variables that depend on the design characteristics of the system may be required: storage volume, pump head, type of rehabilitation to be carried out for various parts of the network, and so forth. The estimation of individual costs will always depend on these variables. The correct approach to assess the costs for each element is important when defining the objective function, which has to be fully adapted to the problem under consideration in terms of design, enlargement, rehabilitation, operational design, and so forth. For example, in the network shown in Figure 1, some of the areas may be already built,

while others must be newly designed. For the already existing pipes, the objective may consist in one of various actions: rehabilitation (with several available alternatives with associated costs), replacement, or simply duplication.

In addition, it is important that the objective function reflects with utmost reliability the total cost of the system during its lifetime [6].

Many authors have used, in their optimization, an objective function that only considers the cost of pipelines, new and/or additional, duplicated pipelines, while others have taken into account other various costs [7, 8]. For the sake of simplicity, we only describe in detail here the cost related to the L pipes of the network, which includes the cost of new pipes, L_{new} , and the cost of rehabilitated pipes, L_{reh} . This function, besides contributing most of the total cost, exhibits the characteristics we are interested in underlining. Costs corresponding to other elements (tanks, pumps, valves, etc.), which are typically nonlinear, are just new terms to be added within this fitness function (see [7], for example).

The cost of the pipes is expressed as

$$C(D) = \sum_{i=1}^L c(D_i) \cdot l_i, \quad (2.1)$$

and the sum extended to all ($L = L_{\text{new}} + L_{\text{reh}}$) individual pipes. $D = (D_1, \dots, D_L)^t$ is the vector of the pipe diameters. The costs per meter, depending on the diameter of pipe i , D_i , are given by $c(D_i)$; l_i is the i th pipe length. Note that D_i is chosen from a discrete set of commercially available diameters, and $c(\cdot)$ is a nonlinear function of diameter.

Typically, the unitary cost of a new pipe, including purchase, transport, and installation costs, takes the form

$$c(D) = (T_1 + T_2 D^\alpha), \quad (2.2)$$

where T_1 , T_2 , and α are characteristic constants of the pipes (information provided by pipe manufacturers, market costs of construction works, etc.). Typically, α takes values different from 1. Maintenance costs are usually considered as a fraction of the installation costs.

There are various rehabilitation options: no rehabilitation, relining, duplication, and replacement are the most usual. Relining costs may be evaluated by the nonlinear function

$$c(D) = T_4 D^\beta, \quad (2.3)$$

with T_4 and β , other characteristic constants. Duplication is equivalent to the installation of a new pipe, and replacement involves, in addition, the cost of removing the old pipe.

2.1.1. Working Conditions or Scenarios

The working conditions of a WDN depend on the values adopted by two types of variables, namely, demand models and roughness coefficient values. Typically, independent random variables are used to model both types of variables. Under the assumption that design is made to work for N_{dm} demand models and N_{rc} sets of roughness coefficient values, the design is performed for $N_{\text{wc}} = N_{\text{dm}} \cdot N_{\text{rc}}$ working conditions. Each of these conditions

has individual probabilities, P_{wc}^k , $k = 1, \dots, N_{wc}$, given by the product of the corresponding probabilities regarding demand models and roughness values. In addition, these probabilities verify

$$\sum_k P_{wc}^k = 1. \quad (2.4)$$

In the case of inclusion of the operational costs of the network along a certain temporal horizon, the necessary amortization rates must be considered. In this case, the objective cost function, omitting for simplicity the independent variables, may be represented by

$$C_{Net} = \sum_k \left[P_{wc}^k (a_{pipe} \cdot C_{pipe} + a_{pump} \cdot C_{pump} + a_{valv} \cdot C_{valv} + a_{tank} \cdot C_{tank} + C_{Oper}) \right]. \quad (2.5)$$

In this case, values a_{xxx} correspond to amortization rates, C_{pipe} is given by expression (2.1), and the costs of pumps, tanks, valves, and operation (which have not been detailed in this paper, as stated) are also considered. Observe that, in general, C_{Net} is a nonlinear, partially stochastic function depending on continuous, discrete, and binary variables.

2.2. Hydraulic Constraints

When modeling physical processes where the underlying functions are known, the deterministic equations can be solved to forecast the model output to a certain degree of accuracy. In hydraulic modeling, different governing laws, [9], can provide a very accurate description of the process provided that the initial and/or boundary conditions and the forcing terms are precisely defined.

Analyzing pressurized water systems is a mathematically complex task that hydraulic engineers must face, especially for the large systems found in even medium-sized cities, since it involves solving many nonlinear simultaneous equations. Several formulations are available (see, for example, [9]). One formulation considers the $N - 1$ continuity equations, which are linear, plus the L energy equations, typically nonlinear

$$\sum_{j \in N_i} q_{ij} = Q_i, \quad i = 1, \dots, N - 1, \quad (2.6)$$

$$H_{k1} - H_{k2} = R_k q_k |q_k|, \quad k = 1, \dots, L.$$

N is the number of demand junctions, and L is the number of lines in the system. N_i is the number of nodes directly connected to node i ; Q_i is the demand associated to node i ; $k1$ and $k2$ represent the end nodes of line k , which carries an unknown flowrate q_k and is characterized by its resistance R_k , which depends on the diameter D_k and on q_k through the Reynolds number (the nonlinearity of the energy equations arises not only from the quadratic term, but also from the function R_k). H_{k1} and H_{k2} , piezometric heads at nodes $k1$ and $k2$, are

unknown for consumption nodes and are given for fixed head nodes. The complete set of equations may be written by using block matrix notation as

$$\begin{pmatrix} A_{11}(q) & A_{12} \\ A_{12}^t & 0 \end{pmatrix} \begin{pmatrix} q \\ H \end{pmatrix} = \begin{pmatrix} -A_{10}H_f \\ Q \end{pmatrix}, \quad (2.7)$$

where A_{12} is the connectivity matrix describing the way demand nodes are connected through the lines. Its size is $L \times N_p$, N_p being the number of demand nodes; q is the vector of the flowrates through the lines; H is the vector of unknown heads at demand nodes; A_{10} describes the way fixed head nodes are connected through the lines and is an $L \times N_f$ matrix, N_f being the number of fixed head nodes with known head H_f ; Q is the N_p -dimensional vector of demands. Finally, $A_{11}(q)$ is an $L \times L$ diagonal matrix, with elements

$$a_{ii} = R_i q_i + B_i + \frac{A_i}{q_i}, \quad (2.8)$$

with $R_i = R_i(D_i, q_i)$ being the line resistance and A_i, B_i coefficients characterizing a potential pump. System (2.7) is a nonlinear problem, whose solution is the state vector $x = (q^t, H^t)^t$ (flowrates through the lines and heads at the demand nodes) of the system.

Since most water systems involve a huge number of equations and unknowns, the system (2.7) is usually solved using some gradient-like technique. Various tools to analyze water networks using gradient-like techniques have been developed in the past. Among them, EPANET2, [10], is used in a generalized way.

To be integrated in the algorithm later described, we have modified the EPANET Toolkit to support pressure-driven demands as described in [11]; the idea of pressure-driven demands has also been considered in other works [12, 13]. The demand at a certain node is formulated for pressure (piezometric head) values, H_{real} , between the minimum pressure allowed, H_{inf} , and the required pressure, H_{req} , with $0 < H_{\text{inf}} < H_{\text{req}}$, at the node by

$$\frac{Q_{\text{real}}}{D_{\text{req}}} = \left(\frac{H_{\text{real}} - H_{\text{inf}}}{H_{\text{req}} - H_{\text{inf}}} \right)^{0.5}, \quad \text{if } H_{\text{req}} \leq H_{\text{real}} \leq H_{\text{inf}}, \quad (2.9)$$

a function of Q_{real} and D_{req} representing the real flow delivered and the demand requested at the node, respectively. Along with this equation, two other conditions complete the definition of the demand at the node

$$\begin{aligned} \text{(i) if } H_{\text{real}} \geq H_{\text{req}}, \quad \text{then } Q_{\text{real}} &= D_{\text{req}}, \\ \text{(ii) if } H_{\text{real}} \leq H_{\text{inf}}, \quad \text{then } Q_{\text{real}} &= 0. \end{aligned} \quad (2.10)$$

The integration of such software to run different analyses or simulations for potential solutions of the problem is performed during the optimization process that is developed within the evolutionary algorithms [14–16], such as the algorithm presented in this paper.

2.3. Satisfaction of Demand Quality

WDN design is typically performed subject to several performance constraints in order to achieve an adequate service level. The most used constraint requires a certain minimum pressure level at each node of the system. Other constraints may include maximum pipe flow velocities and minimum concentrations of chlorine, for example. For many years, nodal pressure constraints have been considered as strong constraints in the sense that they should be strictly satisfied. Nevertheless, the possibility of violating by a small degree some of these constraints opens the door to various strategies for adopting suboptimal designs or soft solutions that may be more convenient from other (global or political) perspectives. This fact has been openly favored by multiobjective approaches such as the one we present in this paper.

In many studies, these constraints have been included as penalty terms in the cost function. However, in this paper, we consider the satisfaction of demand quality as a new objective that must be fulfilled.

There are various ways of expressing lack of compliance with pressure, velocity, disinfectant, and so forth conditions. For example, an objective function considering nodal and velocity constraints given by minimum values of node pressures and pipe velocities may be given by

$$P = \alpha \sum_{j=1}^N H(p_{\min} - p_j) \cdot (p_{\min} - p_j) + \beta \sum_{i=1}^L H(v_{\min} - v_i) \cdot (v_{\min} - v_i), \quad (2.11)$$

where all the functions involved depend on the D , the vector of diameters, through the hydraulic model presented in the previous section.

Here, N is the number of demand nodes in the network, and L is the number of pipes. For nodes with pressures greater than this minimal value, the associated individual terms vanish, and the Heaviside step function H is used in this explicit expression. The same argument applies to pipe velocities (note that absolute values for velocities are considered since flow may occur in any direction). Parameters α and β help normalize the importance of the different scales between pressure and velocity, and this enables a more meaningful aggregation of different types of constraint violation and can also be used to balance the importance of one over the other. Extensions of (2.11) may be provided to consider maximum bounds for both variables. It is also straightforward to extend (2.11) to consider additional objectives, such as limiting the level of chlorine in each pipe in the case of water quality optimization. This expression is also a function of the selected pipe diameters through the hydraulic model presented in the next subsection.

2.4. Reliability and Tolerance

WDNs have almost always been designed with loops so as to provide alternative paths from the source to every network node or junction. This reduces the number of affected consumers when a pipe is withdrawn from service for various reasons. Under normal operating conditions, there is no need for loops, meaning that a looped network is redundant. Redundancy is the capacity of the network to distribute water to users using alternative routes. Redundancy is only needed to maintain service, reduce deficit, and minimize the number of affected consumers when a pipe is withdrawn from service. Redundancy includes

two important concepts: firstly, the connectivity necessary to provide alternative flow paths to each node; secondly, the provision of an adequate flow capacity (diameter) for those paths [17].

The concept of redundancy is closely related to reliability [18–20]. The concept of reliability was introduced to quantitatively measure the possibility of maintaining an adequate service for a given period.

The reliability calculation in looped water supply networks is formulated in the literature as a function of the causes affecting consumer demand. The causes usually considered include real demand exceeding the design demand (e.g., a fire demand); growth of population served by the network; pipe aging; pipe failure.

An explicit formulation of all these causes in probabilistic terms and their further integration implies considerable mathematical and algorithmic complexity. Thus, although numerous WDN reliability quantification schemes exist [17, 21], most are computationally expensive.

This paper considers a simple reliability formulation as in [11, 22, 23] which only considers pipe failures. It is assumed that a pipe temporarily withdrawn from service can be isolated, and so only those consumers connected to that pipe are affected. Only one pipe failure at a time is considered in the formulation of reliability. This is supported by the well-known fact that the probability of simultaneous failure by two or more pipes is extremely small [11, 19, 20, 24–29].

Accordingly, it is accepted that the probability of simultaneous pipe failures is practically zero. In this case, the probability pf_0 of the whole network working without failure is

$$pf_0 = 1 - \sum_{k=1}^L pf_k, \quad (2.12)$$

where k is a pipe counter; L is the total number of pipes in the network; pf_k is the failure probability of pipe k .

The value of pf_k can be obtained from empirical formulae as a function of pipe diameter and length [20, 27, 28, 30, 31].

Considering an average time for the duration of pipe failure, reliability R is defined as

$$R = \frac{1}{q^{\text{req}}} \left(q^{\text{nf}} pf_0 + \sum_{k=1}^L q^k pf_k \right), \quad (2.13)$$

where q^{req} is the total required demand by the network (the sum of all nodal demands); q^{nf} is the total flow delivered to the network when there are no failures; q^k is the total flow delivered to the network when pipe k fails. Again, R is a function of D , the vector of diameters, through the hydraulic model.

After this definition, it can be seen that reliability R represents the expected fraction of q^{req} that can be maintained for a certain time horizon, providing the network properties used for this reliability calculation are maintained.

By calling $r_0 = q^{nf}/q^{req}$ and $r_k = q^k/q^{req}$, as follows(2.13) can be written:

$$R = r_0 p f_0 + \sum_{k=1}^L r_k p f_k. \quad (2.14)$$

As WDNs should behave satisfactorily under normal conditions when there are no failures ($r_0 = 1$), it is worthwhile making a separate and specific analysis of their behavior under only failure states. Accordingly, the concept of *tolerance* to failure T has been introduced [22, 23] using the expression

$$T = \frac{R - r_0 p f_0}{1 - p f_0} = \frac{\sum_{k=1}^L r_k p f_k}{\sum_{k=1}^L p f_k}. \quad (2.15)$$

In this expression, the variables are related to the whole network, and T is a function of D . However, tolerance can also be formulated for each individual node if desired.

This tolerance to failure represents the expected q^{req} fraction that the network supplies as an average when it is in a state of failure. In other words, this index answers the question of how well the network behaves, on average, when a pipe is removed from service.

From (2.15), a very important conclusion can be derived: the value of tolerance is not influenced by the value of r_0 .

Kalungi and Tanyimboh [23] showed that despite the fact that tolerance is not an explicit measure of redundancy, the tolerance index is a good measure of the impact of redundancy. Moreover, tolerance seems to give an adequate inverse measure of network vulnerability, or of the vulnerability of the whole system if other components are included in the calculation. It is an inverse measure, because the greater the tolerance, the lesser the vulnerability.

In short, the reliability concept, as defined above, can be regarded as simply a measure of the behavior of the network under normal conditions: meaning that reliability is not a good measure of behavior under failure situations. This is because the term $r_0 p f_0$ from (2.15) is absolutely predominant. Moreover, tolerance T refers only to the time during which the network is in a failure state. The main aim in looping the network is to reduce the consequences of failures. Therefore, we will use both reliability and tolerance objectives in the optimal design of WDNs.

An additional factor considered by various authors (see, for example, [17]) is that both reliability and tolerance assessment depends on the real demand expected to be required at each node. Demands are considered not as fixed values but as random variables, and a fixed design demand will not be established a priori. However, it is usual to estimate design demands as a fixed value that is the result of multiplying an average demand (usually an estimated value of liters per person per day) by some coefficient reflecting expected peaks of demand.

3. Combining Swarm Optimization and Multiagent Paradigm for Multiobjective WDN Design

Given the complexity of the problem described above, we now develop suitable tools to handle the problem. Our approach is a synergetic combination of swarm intelligence

principles together with multiagent system properties aimed at solving the above multiobjective problems. Let us first briefly introduce the necessary details around multiobjective optimization.

In multiobjective optimization, the goal is to find the vector, X , in the decision or search space $S \subseteq \mathbb{R}^d$, of decision variables, x_1, \dots, x_d , that satisfy a set of constraints, and optimizes a vector function, $F(x) = (f_1(x), \dots, f_m(x))^t$ in the objective space $F(S) \subset \mathbb{R}^m$, with m components representing the various objective functions considered. As a consequence, a multiobjective optimization problem may be formulated as follows

$$\text{optimize } F(x) = (f_1(x), \dots, f_m(x))^t \quad (3.1)$$

subject to

$$\begin{aligned} g_i(x) &> 0, \quad i = 1, 2, \dots, k, \\ h_j(x) &= 0, \quad j = 1, 2, \dots, l, \end{aligned} \quad (3.2)$$

where k and l are inequality and equality constraints, respectively.

Both the decision and the objective spaces are multidimensional, and a change in the decision variables producing a positive increment in one of the components of F often simultaneously causes worse values in other components of F . If the objectives are conflicting in this way, the goal is to find, from all the sets of solutions satisfying the constraints, the set of solutions that yield optimal values with respect to all the objective functions. This set is called the Pareto optimal solution set, $P \subset S$, and its image $F(P)$ in the objective space is called the Pareto front [32]. This set represents a cost-benefit trade-off among the considered objectives and enables informed decision making.

Each solution in the Pareto optimal set is optimal because improvements in one of the components are not possible without impairing at least one of the other components. Two solutions are compared based on the concept of *dominance*. The concept of dominance (for a minimization problem) is concisely defined by stating that solution X dominates another solution Y , and we write $X \leq Y$ if $X \neq Y$ and X is not worse than Y for any of the objectives, that is,

$$X \text{ dominates } Y \text{ if } \{f_i(X) \leq f_i(Y), i = 1, \dots, m\}, \{\exists j, 1 \leq j \leq m : f_j(X) < f_j(Y)\}. \quad (3.3)$$

Two solutions are termed indifferent or incomparable if neither dominates the other. In general, the goal of a multiobjective optimization algorithm is to identify P and its image $F(P)$. However, finding solutions in P may be too difficult, or the Pareto optimal set may consist of a prohibitory large number of decision alternatives. In fact, what is frequently sought is an approximation of the global Pareto-optimal set of design solutions.

The design of a WDN is a large-scale combinatorial, nonlinear, multiobjective optimization problem, involving various types of decision variables and many complex implicit constraints, such as the hydraulic constraints already mentioned. There is no single search algorithm for solving such real-world optimization problems without compromising solution accuracy, computational efficiency, and problem completeness.

Classical methods of optimization involve the use of gradients or higher-order derivatives of the fitness function. But these methods are not well suited for many real-world problems since they are unable to process inaccurate, noisy, discrete, and complex data. Robust methods of optimization are often required to generate suitable results. Several works (e.g., [33–35]) have shown that evolutionary algorithms and, in particular, genetic algorithms are suitable for handling this type of problem.

Many researchers have shifted direction and embarked on the implementation of various evolutionary algorithms: genetic algorithms, ant colony optimization, particle swarm optimization, simulated annealing, shuffled complex evolution, harmony search, and memetic algorithms, among many others. These derivative-free global search algorithms have been shown to obtain better solutions for large network design problems. Recent examples of the use of evolutionary algorithms for multiobjective design of WDN include [36–38].

The advantages of the growing use of evolutionary algorithms in the optimal design of WDN include [5]

- (1) evolutionary algorithms can deal with problems in a discrete manner, which unlike other optimization methods, enables the use of naturally discrete variables and the use of the binary variables in yes/no decisions so frequently in many real-world problems;
- (2) evolutionary algorithms work with only the information of the objective function, and this prevents complications associated with the determination of the derivatives and other auxiliary information;
- (3) evolutionary algorithms are generic optimization procedures and can directly adapt to any objective function, even if it is not described by closed expressions, but by whole, complex procedures;
- (4) because evolutionary algorithms work with a population of solutions, various optimal solutions can be obtained, or many solutions can be obtained with values close to the optimal objective function, and this can be of great value from an engineering point of view;
- (5) an analysis of systems with various loading conditions or forcing terms can be performed within the optimal design process.

A particle swarm optimization-based environment has been developed by the authors that mimics the judgment of an engineer. It was built by using various prior features and improvements regarding swarm intelligence, multiagent systems, and the necessary adaptation to multiobjective performance, including human interaction.

3.1. Swarm Intelligence Approach

The first feature derives from the philosophy behind PSO (particle swarm optimization) [39]. It consists of a variant of the standard PSO that can deal with various types of variables [40], includes a mechanism for increased diversity [15, 41], and enables the self-management of the parameters involved so that engineers are spared the task of parameter selection and fine-tuning [16]. A concise description follows.

A swarm of M particles is initially randomly generated. A particle, X , is represented by its location in a d -dimensional subset, $S \subset \mathbb{R}^d$ (search space). Any set of values

of the d variables, determining the particle location, represents a candidate solution for the optimization problem. The optimal solution is then searched for by iteration. The performance of each particle is measured using one or more fitness functions, according to the problem in hand. During the process, a particle X is associated with three vectors;

- (i) current position, $X = (x_1 \dots x_d)$;
- (ii) best position, $Y = (y_1 \dots y_d)$, reached in previous cycles; and
- (iii) flight velocity $V = (v_1 \dots v_d)$, which makes it evolve.

The particle which is in the best position, Y^* , is identified in every iteration.

In each generation, the velocity of each particle is updated as in (3.4) based on its recent trajectory, its best encountered position, the best position encountered by any particle, and a number of parameters as follows: ω is a factor of inertia suggested in [42] that controls the impact of the velocity history on the new velocity; c_1 and c_2 are two positive acceleration constants, called the cognitive and social parameters, respectively,

$$V \leftarrow \omega V + c_1 R_1(Y - X) + c_2 R_2(Y^* - X), \quad (3.4)$$

where R_1 and R_2 are $d \times d$ diagonal matrices with their in-diagonal elements randomly distributed within the interval $[0, 1]$.

For discrete variables, we use

$$V \leftarrow \text{fix}(\omega V + c_1 R_1(Y - X) + c_2 R_2(Y^* - X)), \quad (3.5)$$

where $\text{fix}(\cdot)$ is a function that takes the integer part of its argument.

Expressions (3.4) and (3.5) are used to calculate the particle's new velocity, a determination that takes into consideration three main terms: the particle's previous velocity; the distance of the particle's current position to its own best position; the distance of the particle's current position to the swarm's best experience (position of the best particle).

In each dimension, particle velocities are clamped to minimum and maximum velocities, which are user-defined parameters,

$$V_{\min} \leq V_j \leq V_{\max}, \quad (3.6)$$

in order to control excessive roaming by particles outside the search space. These very important parameters are problem dependent. They determine the resolution with which regions between the present position and the target (best so far) positions are searched. If velocities are too great, particles might fly through good solutions. On the other hand, if they are too slow then, particles may not explore sufficiently beyond locally good regions, becoming easily trapped in local optima and unable to move far enough to reach a better position in the problem space. Usually, V_{\min} is taken as $-V_{\max}$.

There is, however, a singular aspect regarding velocity bounds that must be taken into consideration so that the algorithm can treat both continuous and discrete variables in a balanced way. In [40], it was found that using different velocity limits for discrete and continuous variables produces better results.

Finally, the position of each particle is updated every generation. This is performed by adding the velocity vector to the position vector,

$$X \leftarrow X + V. \quad (3.7)$$

Thus, each particle or potential solution moves to a new position according to expression (3.7).

The main drawback of PSO is the difficulty in maintaining acceptable levels of population diversity while balancing local and global searches [43]; as a result, suboptimal solutions are prematurely obtained [44]. Some evolutionary techniques maintain population diversity by using some more or less sophisticated operators or parameters. Several other mechanisms for forcing diversity in PSO can be found in the literature [45–47]. In general, the random character that is typical of evolutionary algorithms adds a degree of diversity to the manipulated populations. Nevertheless, in PSO, those random components are unable to add sufficient diversity.

Frequent collisions of birds in the search space, especially with the leader, can be detected—as shown in [15]. This causes the effective size of the population to decrease and the algorithm's effectiveness to be consequently impaired. The study in [41] introduces a PSO derivative in which a few of the best birds are selected to check collisions, and colliding birds are randomly regenerated after collision. This random re-generation of the many birds that collide with the best birds has been shown to avoid premature convergence because it prevents clone populations from dominating the search. The inclusion of this procedure into PSO greatly increases diversity, as well as improving convergence characteristics and the quality of the final solutions.

The role of inertia, ω , in (3.4) and (3.5) is considered critical for the convergence behavior of the PSO algorithm. Although inertia was constant in the early stages of the algorithm, it is currently allowed to vary from one cycle to the next. As inertia facilitates the balancing of global and local searches, it has been suggested that ω could be allowed to adaptively decrease linearly with time, usually in a way that initially emphasizes global search and then, with each cycle of the iteration, increasingly prioritizes local searches [48]. A significant improvement in the performance of PSO, with decreasing inertia weight across the generations, is achieved by using the proposal in [49],

$$\omega = 0.5 + \frac{1}{2(\ln(k) + 1)}, \quad (3.8)$$

where k is the iteration counter.

In the variant, we propose that, the acceleration coefficients and the clamping velocities are neither set to a constant value, as in standard PSO, nor set as a time-varying function, as in adaptive PSO variants [50]. Instead, they are incorporated into the optimization problem [16]. Each particle is allowed to self-adaptively set its own parameters by using the same process used by PSO—and given by expressions (3.4) or (3.5), and (3.7). These three parameters are considered as three new variables that are incorporated into position vectors X . In general, if d is the dimension of the problem, and p is the number of self-adapting parameters, the new position vector for X will be

$$X = (x_1, \dots, x_d, x_{d+1}, \dots, x_{d+p}). \quad (3.9)$$

Clearly, these new variables do not enter the fitness function, but rather they are manipulated by using the same mixed individual-social learning paradigm used in PSO. Also, V and Y , which give the velocity and thus-far best position for the particle, increase their dimension, correspondingly.

By using expressions (3.4) or (3.5), and (3.7), each particle is additionally endowed with the ability to self-adjust its parameters by taking into account the parameters it had at its best position in the past, as well as the parameters of the leader, which facilitated this best particle's move to its privileged position. As a consequence, particles use their cognition of individual thinking and social cooperation to improve their positions, as well as improving the way they better their position by accommodating themselves to the best-known conditions, namely, their conditions and their leader's conditions when they achieved the thus-far best position.

Although the authors have applied this algorithm mainly to WDN design, it has proven very efficient in solving optimization problems in other fields [51–54].

3.2. Multiagent Paradigm Adoption

The emergent behavior of a PSO swarm is strongly reminiscent of the philosophy behind the multiagent (MA) paradigm [55, 56]. In an MA system each agent has a limited capacity and/or incomplete information to resolve a problem, and, therefore, has a limited view of the solution. There is no overall control of the system; values are decentralized, and the computation is asynchronous [55]. Each agent acting alone cannot solve the problem in all its entirety, but a group of agents, with the coexistence of different views, is better able to find a solution by interacting together. This idea can be clearly extrapolated to the case of multiobjective optimization, since the result of the many interactions occurring within an MA, as explained above, is an improved performance.

For the optimal design of WDNs, an MA system offers considerable added value because of the introduction of several agents with different visions of the evaluation of solutions for the same problem, so enabling a multiobjective optimization that is qualitatively much closer to reality. From a practical standpoint, the development of a multiobjective optimization process enables the combination of economic, engineering, and policy viewpoints when searching for a solution.

Taking into account the desirability of solving the optimal design of WDNs with a multiobjective approach and the benefits offered by MA systems, a departure from the standard behavior of particles in PSO must be performed. In addition to using the concept of dominance, various other aspects must also be restated.

3.3. Adaptation to Multiobjective Performance

Firstly, the concept of leadership in a swarm must be redefined. The most natural option is to select as leader the closest particle to the so-called *utopia point* in the objective space. The utopia point is defined as the point in the objective space whose components give the best values for every objective. The *utopia point* is an unknown point since the best value for every objective is something unknown at the start (and perhaps during the whole process). Accordingly, we use a dynamic approximation of this utopia point, termed *singular point*, which is updated with the best values found so far during the evolution of the algorithm

[38]. Even though this idea resembles the concept of *reference point* [57–59], it is simpler while effective.

Secondly, because each objective may be expressed in different units, it is necessary to make some regularization for evaluating distances in the objective space. Once a regularization mechanism has been enforced, to establish the distance between any two objective vectors, the Euclidean distance between them is calculated. Note that the worst and best objective values are not usually known a priori; they are updated while the solution space is explored.

Thirdly, arguably, the most interesting solutions are located near the singular point and not too far from the peripheral areas of the Pareto front. Therefore, instead of seeking a complete and detailed Pareto front, we may be more interested in precise details around the singular point. Nevertheless, situations can occur when unbalanced Pareto fronts develop with respect to the singular point. Consequently, poorly detailed sections on the Pareto front may appear to be worth exploring. It seems plausible that problem complexity is the cause of this asymmetry in many real-world, multiobjective optimization problems.

It is not easy to find a general heuristic rule for deciding which parts of the Pareto front should be more closely represented and how much detail the representation of the Pareto front should contain. Various ways of favoring the completeness of a Pareto front may be devised. We describe one possible approach based on dynamic population increases to raise the Pareto front density [60, 61] and another approach based on human-computer interaction to complete poorly represented areas of the Pareto front.

3.3.1. Increasing the Density of the Pareto Front

In the first approach, during the search process, swarms can increase their population when needed in order to better define the Pareto front; a particle whose solution already belongs to the Pareto front may, on its evolution, find another solution belonging to the front. In this situation, a new clone of the particle is placed where the new solution is found, thus increasing the density of particles on the Pareto front. Greater densities on the Pareto front must be restricted to the case where the new clone has at least one of its neighbors located further away than some minimal permissible distance in the objective space. It must be noted that two particles are considered to be neighbors when no other particle is located between them for at least one of the objectives considered in the problem.

3.3.2. Human-Computer Interaction

In the second approach, users are allowed to add new swarms for searching in the desired region of the objective space. The concept of a singular point is now extended to any desired point in the objective space for particles to search around.

Decisions are strongly dependent on the people solving the problem and on the problem itself. The user can specify additional points where the algorithm should focus the search and specify how much detail a region should contain. This must be achieved in real time during the execution of the algorithm. Once a new singular point is added, a new swarm is created with the same characteristics as the first created swarm. Swarms will run in parallel, but they share (and can modify) the information related to the Pareto set. Particles from any swarm can be added to the Pareto set. If the user changes the fixed values for a singular point, then the corresponding swarm selects a new leader considering the location of the new singular point.

Human interaction with the algorithm in real time also enables the incorporation of human behavior, so that the human becomes another member of the swarm by proposing new candidate solutions. Eventually, such a solution can be incorporated into the Pareto front or lead the behavior of a group of particles. User solutions will always be evaluated in the first swarm created. If a particle is being evaluated, then the user request waits until the evaluation of the particle is finished. If a solution proposed by the user is being evaluated, then any particle belonging to the first swarm should wait for evaluation. Once any solution is evaluated, the algorithm checks whether it could be incorporated in the Pareto front. Synchronization is effected among all the swarms in order to open access for managing the Pareto front. Proposed solutions could even become leaders of the swarm(s) if they are good enough. At this point, human behavior begins to have a proactive role during the evolution of the algorithm.

The participation of several human agents with different perspectives on a problem is very close to what happens in the practice of engineering decision making, where politicians, economists, engineers, and environmental specialists are involved in final decisions. The idea of incorporating user experience into the search process is a step forward in the development of computer-aided design.

The combination of various swarms within the same algorithm is efficient because it conducts a neighborhood search in which each of the swarms specializes, and the best improvement step in terms of Pareto optimality is followed to yield a new solution. The practice of incorporating different search mechanisms also reduces the probability of the search becoming trapped in local optima.

3.4. The Algorithm

In the approach presented, as explained above, new particles are used that are based on the behavior of particles in PSO. Swarms running in parallel may be distributed in different computers, and it must be ensured that the swarms can communicate amongst themselves—a peer-to-peer scenario would be a good choice for this task. The steps of the algorithm for every swarm may be summarized in Pseudocode 1.

The algorithm and its connection with EPANET2 (modified with the pressure-driven demand feature explained above) were implemented in a software program called WaterIng (<http://www.ingeniousware.net/>) [62], which was developed for water distribution system design and analysis. WaterIng is in constant development and may be downloaded from its website; the installation includes a file with network data as an example. A first step guide is also available to learn the main concepts of how to design a water distribution system using the software.

4. Case Studies

This software has been used to perform the design of the case studies shown below. The multiobjective model implemented by this software has shown robustness and good explanatory outcomes. Decision makers are provided with a set of informed solutions to select the best design with regard, for example, to available resources and/or other criteria.

Some parameters of the algorithm were established a priori for running the case studies; the initial population size was set equal to 20. The inertia weight was calculated using (3.8). Finally, fine-tuning the other parameters is performed by using the self-adaptive

- (1) Set up parameters and initialize the number of iterations to zero.
- (2) Generate a random population of M particles: $\{X_i(k)\}_{i=1}^M$
- (3) Evaluate the fitness of the particles and set the local best location for each particle equal to its current location.
- (4) Form the Pareto front and make a list of particles belonging to the front.
- (5) Build the singular point.
- (6) Find the closest particle to the singular point and establish it as swarm leader.
- (7) While not in termination-condition, do the following:
 - (a) Execute from $i = 1$ to number of particles.
 - Start
 - (i) Change the position of the particle:
 - Determine the inertia parameter $\omega(k)$, according to (3.8).
 - Calculate the new velocity, $V_i(k + 1)$, for particle i according to (3.4) or (3.5).
 - Set a new position, $X_i(k + 1)$, for particle i according to (3.7).
 - (ii) Calculate the new fitness function vector for particle i in its new position.
 - (iii) If the new fitness function vector for particle i dominates the fitness function vector that the particle had before moving to the new position; then set the new position as the best position currently found by particle i .
 - (iv) If particle i is in the list of particles belonging to the Pareto front then:
 - if the new fitness function vector may also be a point on the Pareto front and this new position has at least one of its neighbors located further than the minimal permissible distance from any of the objectives, then add a new particle j (a clone of i) with P_k and P_{kbest} located at the current position of i ;
 - else
 - try to add (if possible) the particle i (at its new position) to the Pareto front; if the particle is added, remove from the list any dominated solution; dominated clones are eliminated from the swarm.
 - (v) If particle i is closer to the singular point than any other particle in the swarm, then set particle i as the leader of the swarm with regard to the singular point.
 - (vi) If particle i is not currently the leader of the swarm, but coincides in position with the leader, then re-generate particle i randomly.
 - End
 - (b) Increase the iteration number.
- (8) Show the Pareto front and related results.

PSEUDOCODE 1

techniques described above. As a termination condition, we ran the algorithm until 600 iterations were completed without improvement. However, the figures presented in this work were taken during the first moments and not after reaching the number of iterations without improvements. An improvement is understood as any positive change in the approximated Pareto front that the algorithm obtains. It must be noted that even if the algorithm reaches its own termination condition, it could still be receiving requests from users or other swarms running in parallel; each swarm can, in addition, restart the search by itself when an update in its Pareto front is needed after the interaction with a user or another swarm.

4.1. Hanoi Network

This is a well-known problem, one of the most explored systems in the research literature, and it has been included in this paper for comparison purposes. In this problem, the

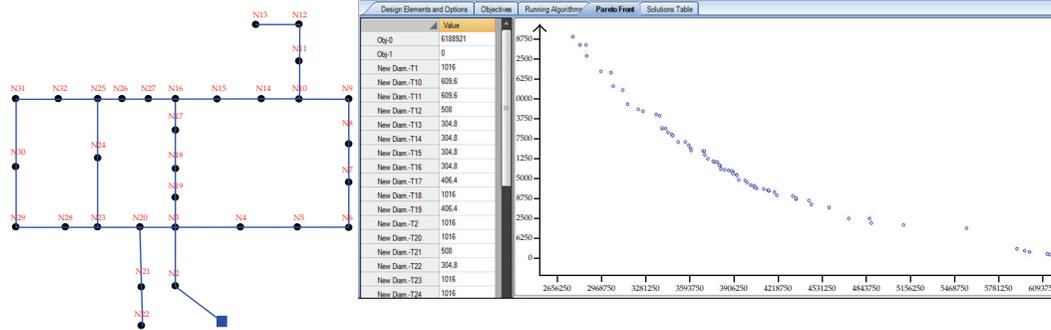


Figure 2: Hanoi network. Approximated Pareto front regarding investment and lack of pressure.

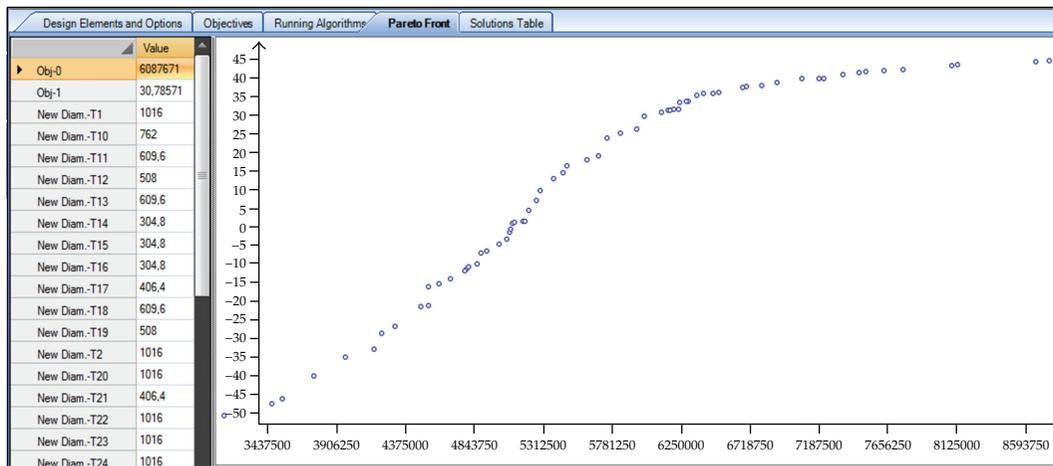


Figure 3: Approximated Pareto front regarding investment and minimum pressure in the network.

engineer seeks the minimum investment cost for a water network, constrained to have at least a minimum pressure value at demand nodes. This constraint was turned into an objective for considering a multiobjective approach: to find solutions minimizing the investment and minimizing the lack of pressure at demand nodes. More information about the original problem can be found in [63]. Figure 2 represents the network layout and an approximated Pareto front considering as objectives the minimization of investment costs and the minimization of the lack of pressure. The estimation of the lack of pressure was made without considering any failure condition in the network.

This problem was also run considering as objectives the minimization of the investment costs and the maximization of the minimum expected pressure in the network (see Figure 3).

All approximated Pareto fronts were obtained in less than 25 seconds (in a Pentium core2duo, 2 GB RAM). Although images were taken when the algorithm was still running (without reaching the termination condition), some similarities with results obtained in other works may be identified (see [7, 14–16, 64, 65] among others). From Figure 2, it can be seen that a solution without lack of pressure is found around 6.2×10^6 \$. After a zoom in the Pareto front it can be seen that the algorithm found a solution without lack of pressure and with a

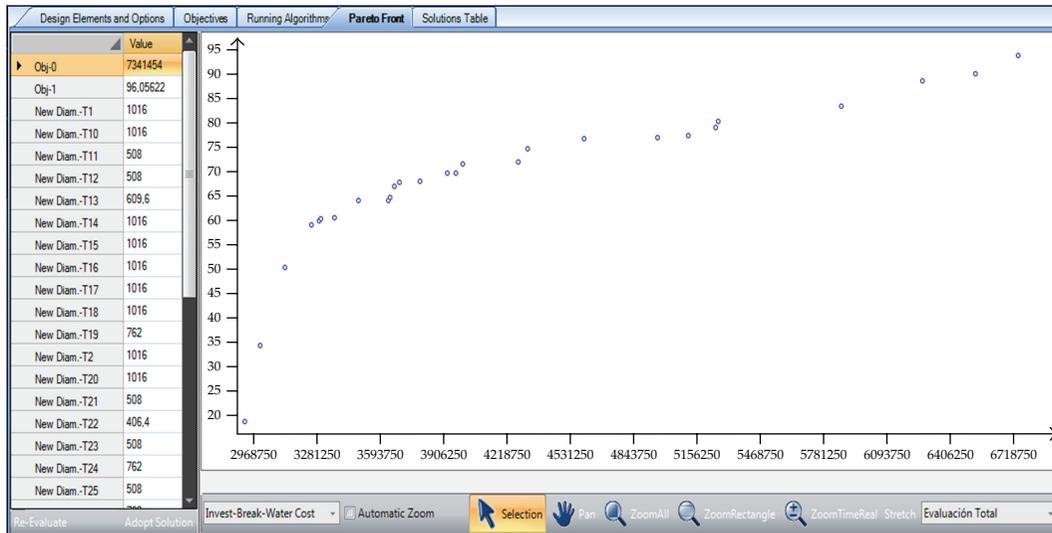


Figure 4: Approximated Pareto front regarding investment and reliability.

cost of 6.12×10^6 \$. This result is very close to the best results obtained by other authors for this problem under the same conditions.

Finally, in Figure 4, an approximated Pareto front considering as objectives the investment costs and the reliability of the network is represented.

4.2. Real Case I: Modified Sector of Lima, Peru

The second case study is a real-world design with three objectives: minimizing the investment cost, minimizing the lack of pressure at demand nodes, and minimizing additional costs caused by reliability problems. This case is a modification of a network designed in collaboration with Wasser SL, a company located at Madrid, Spain.

Results obtained in this design showed a good tolerance to failure conditions. Consideration of tolerance made it possible to find a solution (Figure 5, left) that could distribute all the required water with a pressure satisfying the requirement, even if failure conditions occur in any pipe. In contrast, in Figure 5 (centre and right) problems for the network designed without considering any tolerance to failure condition are presented. Red points indicate a pressure lower than the minimum requirement if the pipe marked with an arrows fails.

Figure 6 represents the solutions initiating the Pareto front. The best solution found for this project when considering a good performance under failure was just 3% more expensive than any solution obtained without consideration of tolerance.

4.3. Real Case II: Modified San José Town

This case was taken from studies developed in San José de las Lajas, Havana, Cuba. A modified variant of the network of this town (see Figure 7) was used for testing the algorithm.

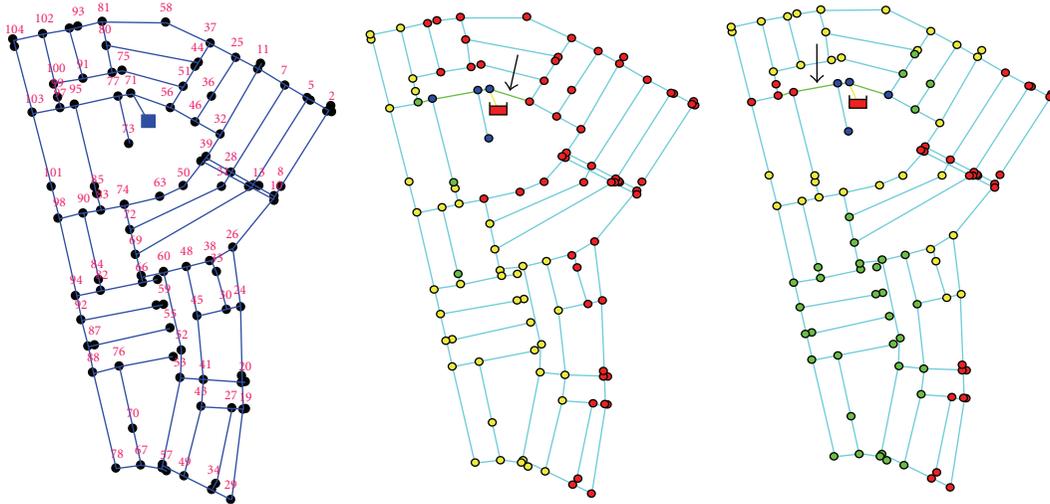


Figure 5: Lima sector, a modified real case example.

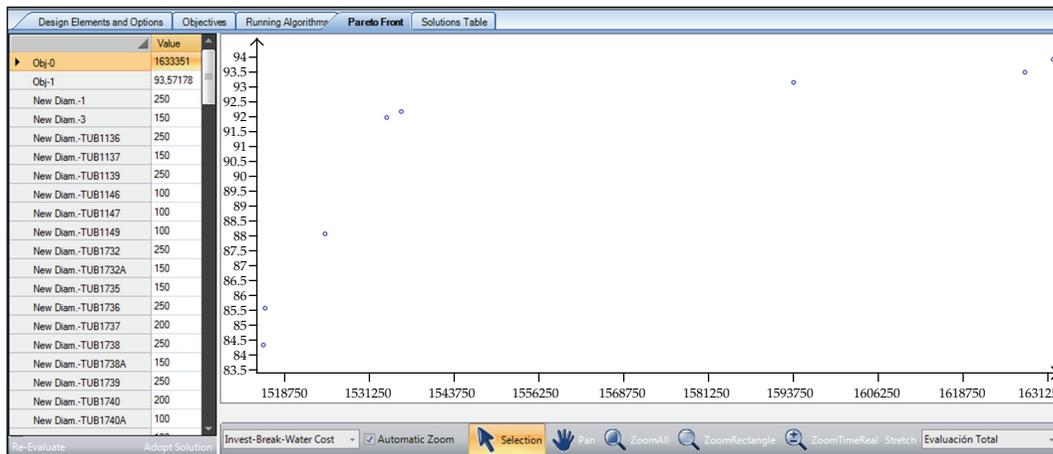


Figure 6: Approximated Pareto front regarding cost and tolerance to failure condition.

In this network, a significant investment is necessary for increasing the minimum pressure at demand nodes. The layout and configuration of the network do not contribute to good pressure distribution throughout the network. This system is currently working without distributing water 24 hours a day; various zones have been determined that receive water at specific moments of the day. Even if this system was designed anew, with the presented configuration and the proposed commercial pipes, the network would have pressure problems at various points. The addition of another tank and different layouts were proposed for analysis in future works. Figure 8 provides relevant information for sound decision making as it shows an approximated relation between investment cost and minimum pressure in the network.

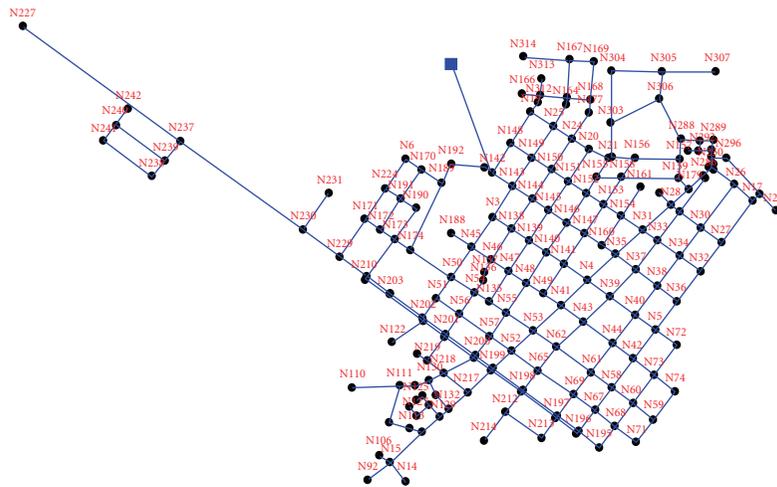


Figure 7: San José town, a modified real case example.

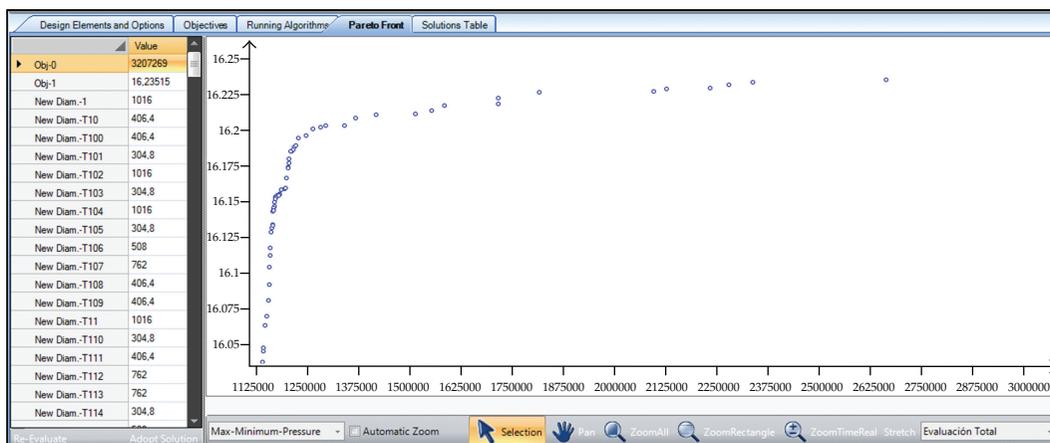


Figure 8: San José town. Approximated Pareto front regarding investment and minimum pressure in the network.

5. Conclusions

In this paper, one of the aspects that reflects the huge complexity of WDNs, namely, the design of these infrastructures, is addressed. We have described the various design elements that render the problem into a large-scale combinatorial, nonlinear, nonconvex, multiobjective optimization problem, involving various types of decision variables and many complex implicit constraints. We then argued that this type of problem is not accessible for classical optimization techniques and discussed the great interest and impact produced by the many evolutionary algorithms.

Based on PSO, we have developed the necessary tools to build an MA algorithm that has proven to be efficient for solving the stated problem. We described some features that substantially improve the searching ability of PSO, paying special attention to two aspects: increase in diversity and self-management of parameters. We then provided the necessary

adaptations to deal with multiobjective problems. We proposed the possibility that various swarms could explore different areas of the Pareto front. And finally, we have integrated human interaction into the process. This offers a special platform for finding solutions in a human-computer cooperative framework. Integrating the search capacity of algorithms with the ability of specialists to redirect the search towards specific points of interest—based on their experience in solving problems—results in a powerful collaborative system for finding solutions to complex engineering problems. Agents can profit from the creativity and ideas of human experts to improve their own solutions; and in turn, human experts can profit from the speed and search capabilities of artificial agents to explore broader solution spaces.

WaterIng is just a first step for engineers working in WDN decision making. The development of the software has not stopped with the contents of this paper; it continues to meet new challenges. Extensions to this work are possible for solving more complex problems, especially in the field of hydraulic engineering. The ideas presented in this paper can also be applied to other engineering fields or, in general, to problems where the search for optimal solutions needs the use of a computer.

One of the aspects we have not explicitly addressed in this paper is the partial stochastic character of the considered problem due to demand variations and the probability, or even the possibility from a fuzzy theory point of view, of the evolution (due to aging) of the roughness within the pipes. Future research topics could explore this aspect, perhaps using recently developed ideas regarding stochastic complex networks, as in [66–69].

Acknowledgments

This work has been developed with the support of the project IDAWAS, DPI2009-11591, of the Dirección General de Investigación of the Ministerio de Educación y Ciencia, and ACOMP/2010/146 of the Conselleria d'Educació of the Generalitat Valenciana. The first author is also indebted to the Universitat Politècnica de València for the sabbatical leave granted during the first semester of 2011. The use of English in this paper was revised by John Rawlins.

References

- [1] J. Izquierdo and R. Pérez-García, Eds., *Applications of Intelligent Data Analysis in Urban Supply Systems*, Universitat Politècnica de València, Valencia, Spain, 2011.
- [2] I. C. Goulter and A. V. Colas, "Quantitative approaches to reliability assessment in pipe networks," *Journal of Transportation Engineering*, vol. 112, no. 3, pp. 287–301, 1986.
- [3] I. C. Goulter and F. Bouchart, "Reliability-constrained pipe network model," *Journal of Hydraulic Engineering—ASCE*, vol. 116, no. 2, pp. 211–229, 1990.
- [4] T. M. Walski, Ed., *Advanced Water Distribution Modeling and Management*, Haestad Press, Waterbury, Conn, USA, 2003.
- [5] I. Montalvo, *Diseño óptimo de sistemas de distribución de agua mediante agent swarm optimization*, Ph.D. thesis, Universitat Politècnica de València, Valencia, Spain, 2011.
- [6] Y. Kleiner, B. J. Adams, and J. S. Rogers, "Water distribution network renewal planning," *Journal of Computing in Civil Engineering*, vol. 15, no. 1, pp. 15–26, 2001.
- [7] A. S. Matías, *Diseño de redes de distribución de agua contemplando la fiabilidad, mediante algoritmos genéticos*, Ph.D. thesis, Universidad Politécnica de Valencia, Valencia, Spain, 2003.
- [8] G. C. Dandy and M. O. Engelhardt, "Multi-objective trade-offs between cost and reliability in the replacement of water mains," *Journal of Water Resources Planning and Management*, vol. 132, no. 2, pp. 79–88, 2006.

- [9] J. Izquierdo, R. Pérez, and P. L. Iglesias, "Mathematical models and methods in the water industry," *Mathematical and Computer Modelling*, vol. 39, no. 11-12, pp. 1353–1374, 2004.
- [10] L. A. Rossman, *EPANET 2 User's Manual*, Environmental Protection Agency, Cincinnati, Ind, USA, 2000.
- [11] C. Xu and I. C. Goulter, "Simulation-based optimal design of reliable water distribution networks," in *Proceedings of the 3rd International Conference on Modeling and Simulation*, Victoria University of Technology, Melbourne, Australia, 1997.
- [12] Z. Y. Wu, R. H. Wang, T. Walski, S. Y. Yang, D. Bowdler, and C. C. Baggett, "Efficient pressure dependent demand model for large water distribution system analysis," in *Proceedings of the 8th Annual Water Distribution System Analysis Symposium*, Cincinnati, Ohio, USA, 2006.
- [13] O. Giustolisi, D. Savic, and Z. Kapelan, "Pressure-driven demand and leakage simulation for water distribution networks," *Journal of Hydraulic Engineering—ASCE*, vol. 134, no. 5, pp. 626–635, 2008.
- [14] I. Montalvo, J. Izquierdo, R. Pérez, and M. M. Tung, "Particle swarm optimization applied to the design of water supply systems," *Computers & Mathematics with Applications*, vol. 56, no. 3, pp. 769–776, 2008.
- [15] I. Montalvo, J. Izquierdo, R. Pérez, and P. L. Iglesias, "A diversity-enriched variant of discrete PSO applied to the design of water distribution networks," *Engineering Optimization*, vol. 40, no. 7, pp. 655–668, 2008.
- [16] I. Montalvo, J. Izquierdo, R. Pérez, and M. Herrera, "Improved performance of PSO with self-adaptive parameters for computing the optimal design of water supply systems," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 5, pp. 727–735, 2010.
- [17] J. B. Martínez, "Cost and reliability comparison between branched and looped water supply networks," *Journal of Hydroinformatics—IWA*, vol. 12, no. 2, pp. 150–160, 2010.
- [18] I. C. Goulter, "Systems analysis in water-distribution network design: from theory to practice," *Journal of Water Resources Planning and Management—ASCE*, vol. 118, no. 3, pp. 238–248, 1992.
- [19] H. Park and J. Liebman, "Redundancy-constrained minimum-cost design of water-distribution nets," *Journal of Water Resources Planning and Management—ASCE*, vol. 119, no. 1, pp. 83–98, 1993.
- [20] D. Khomsi, G. A. Walters, A. R. D. Thorley, and D. Ouazar, "Reliability tester for water-distribution networks," *Journal of Computing in Civil Engineering—ASCE*, vol. 10, no. 1, pp. 10–19, 1996.
- [21] I. C. Goulter, T. M. Walski, L. W. Mays, A. B. A. Sekarya, R. Bouchart, and Y. K. Tung, "Reliability analysis for design," in *Water Distribution Systems Handbook*, L. W. Mays, Ed., McGraw-Hill, New York, NY, USA, 2000.
- [22] T. T. Tanyimboh, M. Tabesh, and R. Burrows, "Appraisal of source head methods for calculating reliability of water distribution networks," *Journal of Water Resources Planning and Management—ASCE*, vol. 127, no. 4, pp. 206–213, 2001.
- [23] P. Kalungi and T. T. Tanyimboh, "Redundancy model for water distribution systems," *Reliability Engineering and System Safety*, vol. 82, no. 3, pp. 275–286, 2003.
- [24] D. R. Morgan and I. C. Goulter, "Optimal urban water distribution design," *Water Resources Research*, vol. 21, no. 5, pp. 642–652, 1985.
- [25] G. A. Walters and J. Knezevic, "Discussion of "Reliability based optimization model for water distribution systems" by Su, Y., Mays, L. W., Duan, N., and Lansey, K.," *Journal of Hydraulic Engineering—ASCE*, vol. 115, no. 8, pp. 1157–1158, 1989.
- [26] G. V. Loganathan, M. P. Shah, and H. P. Sherali, "A two-phase network design heuristic for minimum cost water distribution systems under a reliability constraint," *Engineering Optimization*, vol. 15, no. 4, pp. 311–336, 1990.
- [27] F. Bouchart and I. Goulter, "Reliability improvements in design of water distribution networks recognizing valve location," *Water Resources Research*, vol. 27, no. 12, pp. 3029–3040, 1991.
- [28] R. Gupta and R. Bhave, "Reliability analysis of water-distribution systems," *Journal of Environmental Engineering*, vol. 120, no. 2, pp. 447–460, 1994.
- [29] C. Xu and I. C. Goulter, "Reliability-based optimal design of water distribution networks," *Journal of Water Resources Planning and Management—ASCE*, vol. 125, no. 6, pp. 352–362, 1999.
- [30] Y. Su, L. W. Mays, N. Duan, and K. E. Lansey, "Reliability based optimization model for water distribution systems," *Journal of Hydraulic Engineering—ASCE*, vol. 113, no. 12, pp. 1539–1556, 1987.
- [31] M. J. Cullinane, K. E. Lansey, and L. W. Mays, "Optimization-availability-based design of water-distribution networks," *Journal of Hydraulic Engineering—ASCE*, vol. 118, no. 3, pp. 420–441, 1992.
- [32] V. Pareto, *Cours d'Economie Politique*, Université de Lausanne, Lausanne, Switzerland, 1896.
- [33] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Boston, Mass, USA, 1989.

- [34] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, New York, NY, USA, 2001.
- [35] D. A. Savic, "Single-objective vs. multiobjective optimisation for integrated decision support integrated assessment and decision support," in *Proceedings of the 1st Biennial Meeting of the International Environmental Modeling and Software Society*, A. E. Rizzoli and A. J. Jakeman, Eds., vol. 1, pp. 7–12, University of Lugano, Lugano, Switzerland, 2002.
- [36] L. S. Vamvakieridou-Lyroudia, G. A. Walters, and D. A. Savic, "Fuzzy multiobjective optimization of water distribution networks," *Journal of Water Resources Planning and Management—ASCE*, vol. 131, no. 6, pp. 467–476, 2005.
- [37] G. C. Dandy and M. O. Engelhardt, "Multi-objective trade-offs between cost and reliability in the replacement of water mains," *Journal of Water Resources Planning and Management—ASCE*, vol. 132, no. 2, pp. 79–88, 2006.
- [38] I. Montalvo, J. Izquierdo, S. Schwarze, and R. Pérez-García, "Multi-objective particle swarm optimization applied to water distribution systems design: an approach with human interaction," *Mathematical and Computer Modelling*, vol. 52, no. 7-8, pp. 1219–1227, 2010.
- [39] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Piscataway, NJ, USA, 1995.
- [40] J. Izquierdo, I. Montalvo, R. Pérez, and V. S. Fuertes, "Design optimization of wastewater collection networks by PSO," *Computers & Mathematics with Applications*, vol. 56, no. 3, pp. 777–784, 2008.
- [41] M. Herrera, J. Izquierdo, I. Montalvo, and R. Pérez-García, "Injecting diversity into particle swarm optimization. Application to water distribution system design," *Advances in Computer Science and Engineering*, vol. 6, no. 2, pp. 159–179, 2011.
- [42] Y. F. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 69–73, Anchorage, Alaska, USA, 1998.
- [43] C. A. C. Coello, G. B. Lamont, and D. A. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, New York, NY, USA, 2007.
- [44] Y. Dong, J. Tang, B. Xu, and D. Wang, "An application of swarm optimization to nonlinear programming," *Computers & Mathematics with Applications*, vol. 49, no. 11-12, pp. 1655–1668, 2005.
- [45] P. J. Angeline, "Using selection to improve particle swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 84–89, Anchorage, Alaska, USA, 1998.
- [46] M. Løvbjerg, T. K. Rasmussen, and T. Krink, "Hybrid particle swarm optimiser with breeding and subpopulations," in *Proceedings of the Genetic and Evolutionary Computation Conference*, San Francisco, Calif, USA, 2001.
- [47] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas, and M. N. Vrahatis, "Stretching technique for obtaining global minimizers through particle swarm optimization," in *Proceedings of the Workshop on Particle Swarm Optimization*, Indianapolis, Ind, USA, 2001.
- [48] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, Washington, DC, USA, 1999.
- [49] Y. X. Jin, H. Z. Cheng, J. I. Yan, and L. Zhang, "New discrete method for particle swarm optimization and its application in transmission network expansion planning," *Electric Power Systems Research*, vol. 77, no. 3-4, pp. 227–233, 2007.
- [50] M. S. Arumugam and M. V. C. Rao, "On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems," *Applied Soft Computing*, vol. 8, no. 8, pp. 324–336, 2008.
- [51] J. Izquierdo, R. Minciardi, I. Montalvo, M. Robba, and M. Tavera, "Particle Swarm Optimization for the biomass supply chain strategic planning," in *Proceedings of the 4th Biennial Meeting, iEMSs 2008: International Congress on Environmental Modelling and Software*, pp. 1272–1280, Barcelona, Spain, 2008.
- [52] J. L. Díaz, M. Herrera, J. Izquierdo, I. Montalvo, and R. Pérez-García, "A particle swarm optimization derivative applied to cluster analysis," in *Proceedings of the 4th Biennial Meeting, iEMSs 2008: International Congress on Environmental Modelling and Software*, Barcelona, Spain, 2008.
- [53] J. Izquierdo, I. Montalvo, R. Pérez, and V. S. Fuertes, "Forecasting pedestrian evacuation times by using swarm intelligence," *Physica A*, vol. 388, no. 7, pp. 1213–1220, 2009.
- [54] M. Herrera, J. Izquierdo, I. Montalvo, J. García-Armengol, and J. V. Roig, "Identification of surgical practice patterns using evolutionary cluster analysis," *Mathematical and Computer Modelling*, vol. 50, no. 5-6, pp. 705–712, 2009.
- [55] K. P. Sycara, "Multiagent systems," *American Association for Artificial Intelligence—AI Magazine*, vol. 19, no. 2, pp. 79–92, 1998.

- [56] M. Wooldridge, *An Introduction to Multiagent Systems*, John Wiley & Sons, New York, NY, USA, 2002.
- [57] A. P. Wierzbicki, "The use of reference objectives in multiobjective optimization," in *Multiple Criteria Decision Making Theory and Application*, T. Gal and G. Fandel, Eds., Springer-Verlag, Berlin, Germany, 1980.
- [58] K. Deb, J. Sundar, N. U. B. Rao, and S. Chaudhuri, "Reference point based multiobjective optimization using evolutionary algorithms," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, 2006.
- [59] J. Molina, L. V. Santana, A. G. Hernández-Díaz, C. A. Coello Coello, and R. Caballero, "G-dominance: reference point based dominance for multiobjective metaheuristics," *European Journal of Operational Research*, vol. 197, no. 2, pp. 685–692, 2009.
- [60] M. Reyes-Sierra and C. A. Coello Coello, "Multi-objective particle swarm optimizers: a survey of the state-of-the-art," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 287–308, 2006.
- [61] G. Dupont, S. Adam, Y. Lecourtier, and B. Grilheres, "Multi objective particle swarm optimization using enhanced dominance and guide selection," *International Journal of Computational Intelligence Research*, vol. 4, no. 2, pp. 145–158, 2008.
- [62] I. Montalvo, J. B. Martínez-Rodríguez, J. Izquierdo, and R. Pérez-García, "Water distribution system design using agent swarm optimization," in *Proceedings of the 12th Annual Water Distribution Systems Analysis conference (WDSA '10)*, Tucson, Ariz, USA, 2010.
- [63] O. Fujiwara and B. Khang, "A two-phase decomposition method for optimal design of looped water distribution networks," *Water Resources Research*, vol. 26, no. 4, pp. 539–549, 1990.
- [64] D. A. Savic and G. A. Walters, "Genetic algorithms for least-cost design of water distribution networks," *Journal of Water Resources Planning and Management—ASCE*, vol. 123, no. 2, pp. 67–77, 1997.
- [65] A. C. Zecchin, A. R. Simpson, H. R. Maier, and J. B. Nixon, "Parametric study for an ant algorithm applied to water distribution system optimization," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 2, pp. 175–191, 2005.
- [66] Y. Liu, Z. Wang, J. Liang, and X. Liu, "Stability and synchronization of discrete-time Markovian jumping neural networks with mixed mode-dependent time delays," *IEEE Transactions on Neural Networks*, vol. 20, no. 7, pp. 1102–1116, 2009.
- [67] J. Liang, Z. Wang, and X. Liu, "State estimation for coupled uncertain stochastic networks with missing measurements and time-varying delays: The discrete-time case," *IEEE Transactions on Neural Networks*, vol. 20, no. 5, pp. 781–793, 2009.
- [68] Z. Wang, Y. Wang, and Y. Liu, "Global synchronization for discrete-time stochastic complex networks with randomly occurred nonlinearities and mixed time delays," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 11–25, 2010.
- [69] B. Shen, Z. Wang, and X. Liu, "Bounded H-infinity synchronization and state estimation for discrete time-varying stochastic complex networks over a finite-horizon," *IEEE Transactions on Neural Networks*, vol. 22, no. 1, pp. 145–157, 2011.