Water **2015**, *7*, 6496-6515; doi:10.3390/w7116496

OPEN ACCESS

water

ISSN 2073-4441

www.mdpi.com/journal/water

Article

Optimal Sensor Placement for Leak Location in Water Distribution Networks using Evolutionary Algorithms

Myrna V. Casillas [1,*], Luis E. Garza-Castañón [1] and Vicenç Puig [2]

[1] Supervision and Advanced Control Chair, Tecnológico de Monterrey, Campus Monterrey, Av. Eugenio Garza Sada 2501, 64849 Monterrey, Mexico; E-Mail: legarza@itesm.mx

[2] Advanced Control Systems Research Group, Technical University of Catalonia, Rambla Sant Nebridi 10, 08222 Terrassa, Spain; E-Mail: vicenc.puig@upc.edu

* Author to whom correspondence should be addressed; E-Mail: violeta.casillas@gmail.com; Tel.: +52-81-8358-2000 (ext. 5486); Fax: +52-81-8328-4077.

Academic Editor: Miklas Scholz

Received: 20 May 2015 / Accepted: 26 October 2015 / Published: 16 November 2015

Abstract: In this paper, a sensor placement approach to improve the leak location in water distribution networks is proposed when the leak signature space (LSS) method is used. The sensor placement problem is formulated as an integer optimization problem where the criterion to be minimized is the number of overlapping signature domains computed from the original LSS representation. First, a semi-exhaustive search approach based on a lazy evaluation mechanism ensures optimal placement in the case of low complexity scenarios. For more complex cases, a stochastic optimization process is proposed, based on either the genetic algorithms (GAs) or particle swarm optimization (PSO). Experiments on two different networks are used to evaluate the performance of the resolution methods, as well as the efficiency achieved in the leak location when using the sensor placement results.

Keywords: water distribution networks; leak isolation; sensor placement; leak signature space

1. Introduction

As reported by the International Water Association (IWA), water loss reduction with the sustainability of the source and supply and the cost effectiveness of system operations has emerged as a world-wide issue [1]. Currently, in a world struggling to satisfy the water demands of a growing population, leaks are still estimated to account for up to 30% of the total amount of extracted water [2]. To face these challenges, encouraging new technologies arose during the last few decades [3], achieving higher levels of efficiency and that came together with novel methods for leakage management [4]. One important issue when addressing the problem of water loss is the design of original methods for leak location in water distribution networks (WDNs). In this field, major advances have been made during the last few years, most of them regarding methods based on transient states; [5,6] offer a review of transient-based leak detection methods that summarizes current and past contributions, serving as a basis for the development of new techniques. On the other hand, leak detection and isolation techniques using stationary models have also been studied, starting with the seminal paper of [7], which formulates the problem as a least-squares optimization problem. However, the stationary representation of the WDN leads to a non-linear non-explicit model, and the estimation of the associated parameters is not an easy task. Alternatively, in [8], a method based on pressure measurements and leak sensitivity analysis is proposed. This method relies on the analysis of the residuals on-line, *i.e.*, the difference between the measurements and their estimation using the network hydraulic model, and the comparison to given thresholds that takes into account the model uncertainty and the noise. When some of the residuals violate the threshold, they are correlated against the leak sensitivity matrix in order to discover which possible leak is present. Although this approach presents satisfactory results under ideal conditions, its performance decreases in the presence of nodal demand uncertainty and noise in the measurements. This method has recently been applied to a real fault scenario [9], and an improvement of this approach has been developed [10,11], where an extended time horizon analysis of pressure measurements is considered and a comparison between different performance metrics is achieved.

Despite these first results, the performance obtained until now is still far from allowing the detection of WDN leaks with only a few sensors in a robust and fast way. One limitation that we are trying to overcome comes from the fact that the performance achieved is dependent on the location of the sensors installed in the network and the nominal leak size needed to be assumed in the sensor placement process. Actually, leak location and sensor placement approaches should be considered altogether, since the best placement depends on the method that is used to locate the potential leaks and the efficiency of the leak location depends on the sensor placement.

At the same time, the development of a sensor placement strategy has been an extensive subject of research in the general context of model-based fault detection and isolation (FDI) [12], where the approach may depend on the type of isolability criteria [13,14]. These methods do not apply directly to the case of water distribution networks because of the non-explicit non-linear nature of their model. Thus, an *ad hoc* sensor placement strategy should be developed.

In the case of a WDN where the number of sensors that can be installed is limited due to budget constraints, the choice of the sensor placement for leak location aims at maximizing the leak localization, that is the capability to locate the node where the leak is occurring. Most of the previous works have addressed the sensor placement problem regarding contamination monitoring or model/demand calibration. For example, in [15,16], the problem of deploying sensors in a large water distribution network is considered in order to detect the malicious introduction of contaminants; or, in [17,18], the problem of sensor placement for model and demand calibration, respectively, is addressed. On the other hand, less work has been carried out regarding sensor placement for leak location. In [19], an isolability index is defined and used to place the sensors in order to maximize the number of isolable node pairs. Closer to this work, [8] used thresholds on the differences of pressures measured to obtain binary matrices that were used to translate the sensor placement problem into an integer programming problem. Recently, a sensor placement method that uses an entropy-based approach has been proposed [20]. The approach is based on the maximization of the total entropy defined as the ratio of sensing length over pipe length. On the other hand, in [21], a new approach for sensor placement for leak location in WDN is proposed. This approach is used in combination with the projection-based location scheme proposed in [10,11]. Here, the sensor placement problem is formulated as an integer optimization problem where the optimization criterion concerns the minimization of the number of non-isolable leaks according to the isolability criteria introduced.

This paper introduces an optimization-based sensor placement approach to locate leaks in WDNs using the leak signature space (LSS) method [22]. The LSS method is based on transforming the residual vector, obtained from comparing the pressure measurements with their estimation using a hydraulic model, into an original representation where a specific signature is associated with each leak localization, which does not depend on the leak magnitude [22]. The sensor placement problem is formulated as an integer optimization problem where the criterion to be minimized is the number of overlapping signature domains computed from the LSS representation. First, a semi-exhaustive search approach based on a lazy evaluation mechanism that ensures optimal placement in the case of small-sized networks is proposed. For larger networks, a stochastic optimization process is proposed, based on either the genetic algorithms (GAs) or particle swarm optimization (PSO). Experiments on two different networks are used to evaluate the performance of the resolution methods, as well as the efficiency achieved in the leak location when using the sensor placement results.

The paper is organized as follows: Section 2 introduces the LSS method that is proposed for the leak location. In Section 3, the sensor placement optimization problem is formulated, while Section 4 shows the methods proposed to solve it. Section 5 evaluates the performance of the approach on two WDNs. Finally, Section 6 summarizes the contribution and indicates points that deserve further research.

2. LSS-Based Leak Location Method

This section proposes a summary of the LSS method already presented in [22], which is used in our approach as the basis to perform the sensor placement for leak localization. The main idea is to use a transformation that allows one to characterize node leaks by their signature in a given space, independently of the associated leak magnitude.

2.1. Leak Magnitude Linear Dependency Approximation

Let us assume that the behavior of the WDN follows the models described by [23]. Then, we may define the vectors of pressure in the junction nodes, the pressure in reservoirs, the flows through the pipes and the demands as:

$$\begin{aligned}\mathbf{p} &= (p_1, \cdots, p_m)^{\mathbf{T}} \\ \mathbf{p}^* &= (p_1^*, \cdots, p_u^*)^{\mathbf{T}} \\ \mathbf{q} &= (q_1, \cdots, q_f)^{\mathbf{T}} \\ \mathbf{d} &= (d_1, \cdots, d_m)^{\mathbf{T}}\end{aligned} \tag{1}$$

with u corresponding to the number of reservoirs supplying the WDN. Following the representation proposed in [23], the water network model can be solved numerically using a Newton–Raphson iterative scheme where the iteration $k+1$ is given by the following set of equations:

$$\begin{aligned}\mathbf{q}^{k+1} &= (\mathbf{I} - \mathbf{N}^{-1})\mathbf{q}^k - \mathbf{N}^{-1}\mathbf{A}_{11}^{-1}(\mathbf{q}^k)(\mathbf{A}_{12}\mathbf{p}^k + \mathbf{A}_{10}\mathbf{p}^*) \\ \mathbf{p}^{k+1} &= -(\mathbf{A}_{21}\mathbf{N}^{-1}\mathbf{A}_{11}^{-1}(\mathbf{q}^k)\mathbf{A}_{12})^{-1} \cdot (\mathbf{A}_{21}\mathbf{N}^{-1}(\mathbf{q}^k + \mathbf{A}_{11}^{-1}(\mathbf{q}^k)\mathbf{A}_{10}\mathbf{p}^*) + (\mathbf{d} - \mathbf{A}_{21}\mathbf{q}^k))\end{aligned} \tag{2}$$

where $\mathbf{N}$ is a diagonal matrix, such that $\mathbf{N} = \text{diag}(\gamma_i)$, $i \in [1, \cdots, f]$, $\mathbf{A}_{12} = \mathbf{A}_{21}^T$, $\mathbf{A}_{10} = \mathbf{A}_{01}^T$ with $\mathbf{A}_{21}$, $\mathbf{A}_{01}$ being the incidence matrices, $\mathbf{A}_{11}(\mathbf{q}) = \text{diag}(c_i|q_i|^{\gamma_i})$, $|q_i|$ is the absolute value of the flow q_i, c_i is a constant parameter, which depends on the diameter, the roughness and the length of the pipe, and γ_f is the flow exponent parameter. Let us remark that such a resolution approach is commonly employed in numerical hydraulic simulators. This is in particular the pilot case in the EPANET simulator [24] where large WDNs can be efficiently simulated.

Besides, and as stated in [22], the leak can be modeled as an extra demand:

$$\mathbf{d}' = \mathbf{d} + \mathbf{\Delta d} \tag{3}$$

where $\mathbf{\Delta d}$ is an m-dimensional vector with zeros everywhere, except at the node index where the leak occurs. Defining the residual $\mathbf{r}$ as the difference between the nominal pressure obtained from the simulation of the model without leaks and the measured pressure in the network in the case of a leak (*cf.* [8]), we have:

$$\mathbf{r} = \mathbf{p} - \mathbf{p}' \approx (\mathbf{A}_{21}\mathbf{N}^{-1}\mathbf{A}_{11}^{-1}(\mathbf{q})\mathbf{A}_{12})^{-1}\mathbf{\Delta d} = \mathbf{S} \cdot \mathbf{\Delta d} \tag{4}$$

As one can see, this approximation establishes a proportional relation between the residual (and consequently, with the pressure measurement) and the leak through a matrix $\mathbf{S}$, known as the leak sensitivity matrix. In practice, this matrix is obtained not by using Equation (4), but by means of simulating leaks in every node of the WDN using a hydraulic simulator.

The pressure residual in a given WDN can be represented by a point in the m-dimensional residual space. In the case of a leak, Equation (4) indicates that the position of this point is located on a line passing through the origin and whose orientation depends on the node where the leak occurs. Moreover, the position of the point on this line depends on the leak magnitude. However, in practice, pressure measurements are accessible only in a limited number of nodes n that correspond to locations where the sensors are placed. Fortunately, the n-dimensional space of the sensors is a subspace (a projection) of the m-dimensional space of the pressure measurements. Thus, the linear dependency Equation (4) is also valid in this projected space.

2.2. Leak Signature Space

Exploiting the linear dependency revealed in Equation (4), two pairs of residual vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ corresponding to leaks at the same node j, but with different magnitudes are related by:

$$\mathbf{r}_2^j = \alpha \mathbf{r}_1^j \tag{5}$$

with α proportional to the leak magnitude. Hence, each leak location can be characterized by a straight line in the n-dimensional residual space. From these lines, it is possible to characterize leak locations by projecting the lines representing the residuals onto a hyperplane of the n-dimensional residual space. For simplicity, let us assume for now that the last coordinate of the residual vector is chosen to form the hyperplane, such that for a given residual $\mathbf{r}^j = [r_1^j, \cdots, r_n^j]^\mathbf{T}$, the projection vector $\bar{\mathbf{r}}^j$ is computed as:

$$\bar{\mathbf{r}}^j = [\frac{r_1^j}{r_n^j}, \cdots, \frac{r_{n-1}^j}{r_n^j}, 1]^\mathbf{T} \tag{6}$$

Thus, there is a unique expression of such a projection vector for a linear representation of the residuals. Consequently, it is possible to associate with the leak in node j, and independently of its magnitude, a unique point $\tilde{\mathbf{r}}^j = [\bar{r}_1^j, \cdots, \bar{r}_{n-1}^j]$ in this $(n-1)$-dimensional hyperplane, called the leak signature space (LSS).

For practical applications, it is necessary to take into account sensor and demand uncertainty. Therefore, for a network involving n sensors, s different leak magnitudes are simulated to represent a leak associated with a given node j. Then, the final leak signature is obtained as the barycenter of the s partial signatures ${}^k\tilde{\mathbf{r}}^j$ built from the different leak magnitudes:

$$\tilde{\mathbf{r}}^j = \left[\frac{1}{s} \sum_{k=1}^{s} \frac{{}^k r_1^j}{{}^k r_n^j}, \cdots, \frac{1}{s} \sum_{k=1}^{s} \frac{{}^k r_{n-1}^j}{{}^k r_n^j} \right] \tag{7}$$

Following such a scheme, m leak signatures $\tilde{\mathbf{r}}^j$ can be built characterizing each of the possible leak locations. Such signatures can later be exploited to identify leak locations. Notice also that the dimension of the hyperplane in the LSS increases with the number of sensors present, also increasing the chances to discriminate the different leak signatures. Coming together with the leak signatures, a domain of influence for each leak node can be determined as the (n-1)-dimensional sphere. Two overlapping domains mean that two signatures are very similar. These domains will be used later on to evaluate the risks to select an incorrect location instead of the correct one. The radius for each leak domain is defined as:

$$\mathbf{rad}^j = \max_{k \in [1, \cdots, s]} \rho(\tilde{\mathbf{r}}^j, {}^k\tilde{\mathbf{r}}^j) \tag{8}$$

where $\rho(\tilde{\mathbf{r}}^j, {}^k\tilde{\mathbf{r}}^j)$ are the Euclidean distances between the barycenter of the node j and each of the partial signatures obtained by simulating the s different magnitudes.

2.3. Leak Location Method

To estimate the leak location based on the LSS representation, first the m leak signatures of the network nodes are computed with the LSS method. Then, when a leak occurs, its signature in the LSS is determined, *i.e.*, the $(n-1)$-dimensional point $\tilde{\mathbf{r}}^*$ representative of this leak from the real residual measurements is computed. Finally, the Euclidean distance ρ between $\tilde{\mathbf{r}}^*$ and the various leak signatures $\tilde{\mathbf{r}}^j$ in the LSS is computed. The leak node is then estimated as the one whose signature is the closest to the current leak signature:

$$id = arg \min_{j\in[1,\cdots,m]} \rho(\tilde{\mathbf{r}}^*, \tilde{\mathbf{r}}^j) \tag{9}$$

where id is the index of the leaky node and $arg \min$ is the argument of the minimal distance evaluated.

2.4. Incorporating a Time Horizon Analysis

In practice, for a given network, the demand usually varies along time, and it is important to carry out the leak location analysis taking into account a given time horizon [11]. To address this point, it is proposed to record, for each potential leak node j, the various signatures it has along a time horizon that corresponds to one cycle T of the demand pattern. Since the demand varies along this pattern, the position of the leak signatures changes accordingly. Then, in the presence of a leak, a comparison is made between the positions of its signatures in the LSS along T and the positions of the reference leak signatures. Such a comparison is performed by summing the Euclidean distances for each time instant considered. Thus, if T instants of time are considered, there are T residuals ${}_t\tilde{\mathbf{r}}^*$, $t \in [1,\cdots,T]$ obtained from the measured pressures, which are compared to T leak signatures ${}_t\tilde{\mathbf{r}}^j$. The index of the leaky node is then estimated as the one that minimizes the sum of the distances between the node signature and the current leak signature along the time horizon:

$$id = \arg \min_{j\in[1,\cdots,m]} \left(\sum_{t\in[1,\cdots,T]} \rho({}_t\tilde{\mathbf{r}}^*, {}_t\tilde{\mathbf{r}}^j) \right) \tag{10}$$

3. Sensor Placement: Problem Formulation

The objective is to find for a WDN and a given number of sensors n the sensor locations that maximize the performance of the leak location when using the LSS-based leak location method. This method mainly addresses the problem of locating leaks once they have been detected. Leak detection is typically performed by means of the analysis of changes in the night demand flows [4]. By proceeding in this way, it is easier to discriminate in most of the cases between noise and real leaks. Here, we propose an explicit formulation of the sensor placement optimization problem, first considering a single time instant and then extending it to a horizon.

3.1. Considering a Single Time Instant

A given sensor placement corresponds to a unique configuration defined as a m-dimensional binary vector $\mathbf{x}$, such that:

$$\mathbf{x} = \begin{bmatrix} x_1, & \cdots, & x_m \end{bmatrix} \tag{11}$$

where $x_i = 1$ if a sensor is installed (*i.e.*, the pressure is measured) in the node i and $x_i = 0$, otherwise. Then, assuming a projection onto the hyperplane of index $\lambda \in [1\cdots n]$, as discussed in Section 2, we can compute for each node j its leak signatures $\tilde{\mathbf{r}}^j(\mathbf{x},\lambda)$ using (7) and its leak domains $\mathbf{rad}^j(\mathbf{x},\lambda)$ by means of (8). For a given node, the domain of its leak signature represents the region where the signature of the measured leak may appear if the leak actually occurred at this node. Thus, when two leak domains intersect, the risk that two leaks are non-isolable increases. Based on such an analysis, we propose to decide placing the sensors such that the overlapping between leak domains considering the signatures of all network nodes is minimized. For each pair of leak nodes (j, j') whose domains may overlap, a given sensor placement represented by the vector $\mathbf{x}$ and a given projection λ, we define the associated error $\varepsilon(\mathbf{x}, \lambda)^{jj'}$, such as:

$$\varepsilon^{jj'}(\mathbf{x}, \lambda) = \begin{cases} 1 & \text{if } \rho\left(\tilde{\mathbf{r}}^{j}(\mathbf{x}, \lambda), \tilde{\mathbf{r}}^{j'}(\mathbf{x}, \lambda)\right) \leq \mathbf{rad}^{j}(\mathbf{x}, \lambda) + \mathbf{rad}^{j'}(\mathbf{x}, \lambda) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

with $j = 1, \cdots, m-1$ and $j' = j+1, \cdots, m$, where ρ is the Euclidean distance in the LSS space. Then, the error index that takes into account all of the nodes' leaks is computed as:

$$\epsilon(\mathbf{x}, \lambda) = \sum_{j=1}^{m-1} \sum_{j'=j+1}^{m} \varepsilon^{jj'}(\mathbf{x}, \lambda) \tag{13}$$

and the sensor placement problem is formulated as an optimization problem subject to equality constraints of the form:

$$\begin{gathered} \min_{\mathbf{x}, \lambda} \epsilon(\mathbf{x}, \lambda) \\ \text{s.t.} \\ \sum_{i=1}^{m} x_i = n \end{gathered} \tag{14}$$

3.2. Considering a Time Horizon Analysis

As discussed before, the leak localization approach can incorporate a time horizon analysis (see Subsection 2.4). Following such a scheme, it is possible to use a time horizon in the evaluation function, with the aim of improving leak isolation, and consequently, the sensor placement result. To achieve this goal, the error function used in the optimization problem defined in Equation (14) is modified in order to work with the mean number of overlaps along the considered time horizon that is computed as:

$$\bar{\epsilon}(\mathbf{x}, \lambda) = \frac{1}{T} \sum_{t=1}^{T} \epsilon_t(\mathbf{x}, \lambda) \tag{15}$$

where $\epsilon_t(\mathbf{x}, \lambda)$ is the error at the time instant t computed using Equation (13) and T is the length of the time horizon selected.

4. Sensor Placement: Resolution Methods

In this section, we describe three different methods that can be used to obtain solutions to the optimization problem Equation (14) associated with the sensor placement problem. First, a semi-exhaustive approach that leads to the optimal placement is presented. Then, two stochastic methods leading to near-optimal solutions are proposed.

4.1. Semi-Exhaustive Search Approach

As stated in Section 3, the problem of sensor placement involves searching an optimal couple $(\mathbf{x}, \lambda)$ that defines a sensor configuration and projection hyperplane. One trivial approach to solve this problem would be to check the entire set of $n \cdot \binom{m}{n}$ configuration/projection combinations. However, this would result in a very high computational cost. An alternative method that still ensures the optimal sensor location is to search every possible combination, but using a lazy evaluation mechanism to reduce the computation cost. The idea is to discard the couples $(\mathbf{x}, \lambda)$ as soon as it is proven that they cannot be candidates for the optimum. During the search, the couple configuration/projection that achieves so far the minimum number of overlaps between all possible pairs of signature domains is stored. For each new $(\mathbf{x}, \lambda)$ considered, we count the overlaps between possible pairs of signature domains, as long as it does not exceed the best solution stored. If this threshold is reached, the search procedure discards this combination and continues the search with the next one to be considered. Otherwise, the combination becomes the new best solution found so far.

However, even with such a lazy mechanism, this search is only limited to networks of a small size or equipped with only very few sensors. In the following, we present two stochastic optimization approaches that make problems of higher complexity tractable, but that only ensure near-optimal solutions.

4.2. Stochastic Optimization Approaches

The two stochastic optimization techniques that are proposed to solve the sensor placement problem presented in Section 3 are the genetic algorithm (GA) and the particle swarm optimization (PSO) method. Both approaches require the establishment of a cost function that has to be minimized. We propose only to work with the configuration $\mathbf{x}$ defining the sensor placement as the variable to optimize and to deal with the variable λ by considering for a given $\mathbf{x}$ only the best LSS hyperplane projection that leads to the lowest number of overlaps. Therefore, considering the modified error index Equation (13), the cost function is written as follows:

$$\epsilon(\mathbf{x}) = \min_{\lambda \in [1 \cdots n]} \sum_{j=1}^{m-1} \sum_{j'=j+1}^{m} \varepsilon(\mathbf{x}, \lambda)^{jj'} \tag{16}$$

and the optimization problem is now in the form:

$$\begin{gathered} \min_{\mathbf{x}} \epsilon(\mathbf{x}) \\ \text{s.t.} \\ \sum_{i=1}^{m} x_i = n \end{gathered} \tag{17}$$

Before presenting the general optimization scheme, let us provide a short introduction to the GA and PSO methods.

Genetic algorithms are well-known search and optimization tools based on the principles of natural genetics and natural selection [25,26]. Because of their broad applicability, ease of use and global perspective, they have been increasingly applied to various search and optimization problems in the recent past [27]. Some fundamental ideas of genetics are borrowed and used artificially to construct search algorithms that are robust and require minimal problem information. A GA transforms a population of individual objects, each with an associated fitness value, into a new generation of the population using the Darwinian principle of reproduction and survival of the fittest and analogs of naturally-occurring genetic operations, such as crossover (sexual recombination) and mutation. Each individual in the population represents a possible solution to a given problem. The search of a very good (or the best) solution to the problem is performed by genetically breeding the population of individuals over a series of generations. Genetic algorithms are also used to solve multi-objective optimization problems, where the idea is to minimize or maximize a number of objective functions also satisfying the constraints involved in the problem. The application of GAs to the problem of sensor placement is implemented using the GA Toolbox of MATLAB [28], where each chromosome corresponds to the possible presence or absence of a sensor at a given node. The sensor placement is based on the construction of binary genomes, which is necessary in order to allow the population vectors to switch between the presence (“one”) or the absence (“zero”) of a sensor located in the corresponding node. The parameters of the GA are established by experience after several trial and error tests and taking into account the most common configurations. GA is allowed to use the crossover scattered function in order to create the crossover children. To produce the mutation, a Gaussian function is selected. The selection function is set as a stochastic uniform function.

PSO is a population-based stochastic optimization technique developed by [29]. It is inspired by the social behavior of bird flocking or fish schooling and has been successfully applied to many research problems [30].

PSO shares many similarities with evolutionary computation techniques, such as GAs. The system is initialized with a population of random solutions and searches for the optimum value by updating generations. However, unlike GA, PSO has no evolution operators, such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. In recent years, PSO-based methods have been applied to a variety of problems, achieving high levels of efficiency [31,32]. The application of PSO to our problem relies on the PSO toolbox of MATLAB developed by [33] and offered in the MathWorks repository, where a particle can be represented by the possible presence or absence of a sensor at a given node.

The pseudocode of the stochastic optimization method using either GA or PSO is shown in Algorithm 1. First, the variables of the algorithm are initialized (Line 1), including the number of generations, the bit string type population and the convergence tolerance (set as 10^{-10}). Restrictions to the search are also added, which ensures that the number of sensors placed is correct (Line 2). Note that in the PSO case, such a constraint is in practice converted into a high penalty constraint, since equality constraints were not accepted for binary variables. A seed size is chosen (Line 3), which allows one to create an initial matrix with random sensor positions (Lines 4 and 5). Then, the stochastic optimization method evaluates from the initial population matrix and the matrix of residuals a set of sensor placements (Lines 7 to 11) and returns the one that corresponds to the lowest error (Line 12). After it iterations, the method returns the sensor configuration that minimizes the error the most (Line 14), with the corresponding LSS hyperplane of projection λ. Such a configuration minimizing the number of leak signature domain overlaps will allow one to obtain the maximum benefit of the LSS-based leak location method.

Algorithm 1 Sensor placement based on genetic algorithms or particle swarm optimization.

```
Require: the number of sensors $n$, the number of nodes $m$, the $s \times m \times m$ three-dimensional matrix $R$ of residuals
    for the $s$ leak magnitudes modeled and the maximum number of iterations $it$.
Ensure: a near-optimal sensor configuration $\mathbf{x}_{min}$ with error index $\epsilon_{min}$.
Initialize the search with the previous solution.
Set the appropriate restrictions.
Choose the seed size.
for The number of iterations selected do
    Build the initial population matrix with rows randomly initialized.
    Inputs: The initial solution, restrictions, residuals and sensitivities. //Start GA- or PSO-based search.
    while An optimization criterion is not reached do
        Get a possible configuration.
        Analyze the configuration according to Equation (13).
        Evaluate the solution using Equation (16).
    end while
    Evaluate the found solutions and choose the one with the minimal value. //End GA- or PSO-based search.
end for
Find the best solution $\mathbf{x}_{min}$ with the minimal error index $\epsilon_{min}$ between all of the iterations and save it as the
    near-optimal sensor placement.
```

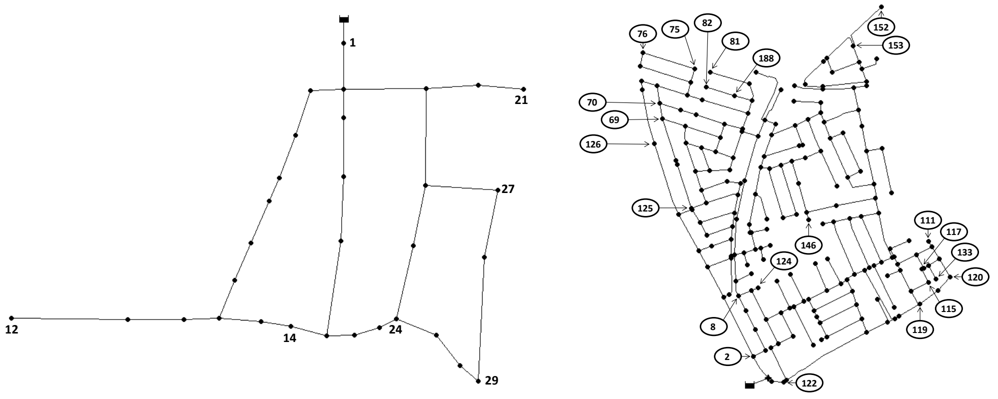
5. Experimental Results

The proposed sensor placement approach presented in the previous sections is applied to two networks. The WDN of Hanoi in Vietnam [34] is the first case study. This academic benchmark has been used in several works [35,36], where the objective was either to design the network or to optimize the operations performed within it. It consists of one reservoir, 31 demand nodes and 34 pipes, as shown in Figure 1a. The second case study is based on the WDN of Limassol in Cyprus proposed as a test case in the EFFINET European project [37]. This real network has already been used in leak location and sensor placement works [21] and consists of one reservoir, 197 demand nodes and 239 pipes, as shown in Figure 1b. All of the experiments were performed in MATLAB, using a Windows 7 computer with a Pentium Dual Core processor of 2 GHz, a memory (RAM) of 4 GB and a 64-bit operating system. The network models are simulated using the EPANET software [24]. Leak magnitudes are simulated by changing a node emitter coefficient (EC) that is related to the flow rate through the following equation:

$$EC = \frac{q}{p^{\gamma}} \tag{18}$$

where q is the flow rate, p is the fluid pressure and γ is the pressure exponent. In the Hanoi network, in order to build the leak signatures, seven EC values are considered going from two to eight with a step size of one, which leads to leak magnitudes varying from 20 to 80 liters per second (lps), which represent 1.7% of the total demand. For the Limassol network, seven EC are also used going from 0.3 to 0.9 with a step size of 0.1, which leads to leaks varying from 2 lps to 6 lps that represent 1% of the total demand.

In the following tables, the optimal sensor placements presented are the ones for which the number of overlaps between leak signature domains is minimum. Additionally, the efficiency is provided, which corresponds to the percentage of leaks correctly located. This performance index is computed by testing leaks in all possible nodes with all possible magnitudes within the range used to build the signature models and domains. The measurements are simulated with a random Gaussian white noise with the mean amplitude corresponding to 0.5% of the expected measurement value.



(a) (b)

Figure 1. The Hanoi (**a**) and Limassol (**b**) water distribution networks (WDNs) used as benchmarks to evaluate the sensor placement method.

5.1. Semi-Exhaustive Search Application

First, the semi-exhaustive approach is tested in the case of 2, 3 and 4 sensors for the Hanoi WDN. Results are summarized in Table 1. It can be noticed that the method allows one to obtain an efficiency of 93.1%, *i.e.*, the percentage of correct leak locations, with two sensors and increases up to 100% in the case of four sensors. Notice also how the computational time tends to increase exponentially with the number of sensors implied, which comes from the fact that all possible sensor combinations have to be considered despite the lazy evaluation mechanism introduced.

Table 1. Sensor placement in the Hanoi network using the semi-exhaustive approach.

Sensors	Node Indexes	Overlaps	Efficiency (%)	Time (s)
2	12, 21	5	93.1	11.2
3	12, 21, 29	1	98.6	181.0
4	1, 12, 21, 29	0	100.0	1637.5

In the Limassol network (see Table 2), the semi-exhaustive search in the case of two sensors leads to evaluating $19,306$ possible placements, which takes more than 5 h. The best configuration obtained gives a total of 253 overlaps and an efficiency of 60.6%. With three sensors, there are more than 1.2×10^6 possible placements, and it is not possible to test all of them in a reasonable time. Thus, we propose to start with the best result when placing two sensors and search for the best third sensor to add. This test will serve as a reference to evaluate the performance of the stochastic optimization approaches later on. The best third sensor is found in 541 s at Node 75, which leads to 166 overlaps and a final efficiency of 72.0%.

Table 2. Sensor placement in the Limassol network using the semi-exhaustive approach.

Sensors	Node Indexes	Overlaps	Efficiency (%)	Time ($s \times 10^3$)
2	8, 152	253	60.6	19.8
2 + 1	8, 75, 152	166	72.0	19.8 + 0.54

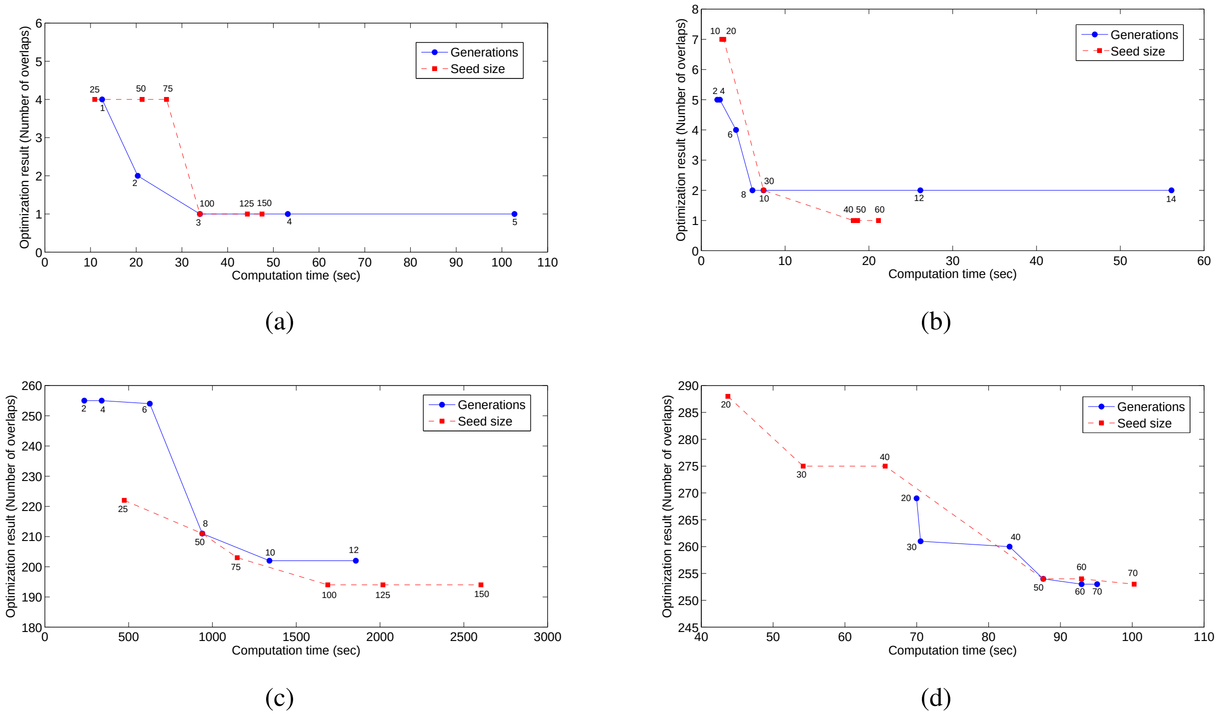
These results show how the semi-exhaustive search allows a good leak detection efficiency thanks to the optimal sensor placement, but is clearly limited to small networks.

5.2. Application of GA and PSO Approaches

Before the application of the GA and PSO approaches, an analysis to determine the best parameter selection has been performed. In this analysis, the size of the seed matrix and the number of generations for each of the networks are chosen after several tests. The seed size is the number of rows in the initialization matrix used at the beginning of the search in the algorithm. This matrix is full of possible random solutions for the sensor placement. First, a seed size is selected to run the test in order to select the appropriate number of generations. Then, the tests are repeated varying the seed size and keeping the number of generations according to the previous result. The results of the tests are shown in Figure 2.

Once the appropriate seed size and number of generations have been selected, several executions of the optimization process with different seed initializations are performed to improve the search efficiency. That is, whatever the method is, it is run several times (each run is called an iteration), and at each iteration, the best solution found is saved and included in the seed matrix of the next iteration. This is clearly a non-standard way to use GA/PSO, but such an approach was chosen during the implementation process after some trials, because it proved to be an efficient way to search for the global optimum. In comparison, a single GA/PSO iteration, but involving more generations, tends to be more easily trapped in local minima. On the opposite side, including more iterations (each with less generations) allows the approach to extend the search when profiting from the solutions from precedent iterations. As an example, when placing two sensors in the Limassol network, running 10 iterations with five generations in each of them gives as a result a placement with 253 overlaps, while running a single iteration with 50 generations finds as the best placement a result with 263 overlaps. As we can see, the iterations approach is more efficient than a single iteration with more generations.

In the case of the Hanoi network, the parameters after the tests are selected as follows: For the GA, the seed matrix was set to 100 rows filled with random solutions, and three generations are allowed. For the GA, three iterations were allowed in order to increase the efficiency of the method. For the PSO, a seed matrix of 50 rows is selected, and 10 generations are allowed. The PSO algorithm is run 50 times (*i.e.*, 50 iterations were allowed) in order to increase the efficiency.



(a)

(b)

(c)

(d)

Figure 2. Evaluation of the parameters to select for application of GA and PSO. (**a**) Tuning for the GA in the Hanoi network; (**b**) tuning for the PSO in the Hanoi network; (**c**) tuning for the GA in the Limassol network; (**d**) tuning for the PSO in the Limassol network.

For the Limassol network, the parameters were selected as follows: A seed matrix of 100 rows and five generations are allowed. The GA algorithm is run 10 times (*i.e.*, a total of 10 iterations). For the PSO, a seed matrix of 50 rows was selected allowing 50 generations. The PSO algorithm will be run 150 times; this means that we are allowing 150 iterations.

5.2.1. Tests on the Hanoi WDN

Both GA and PSO approaches are applied to the Hanoi example. In the following, we show the results obtained after several tests performed with both methods. It is important to mention that all of the conclusions correspond to the cases observed with the settings chosen for both the GA and PSO methods. Results obtained when only a single time instant was considered are shown in Table 3. As one can see, both methods find optimal solutions in all cases, since, in this case, the number of overlaps matches the results of the semi-exhaustive approach. However, note that sometimes, the solution corresponds to a different sensor combination, but resulting in an equivalent efficiency. It is remarkable that GA and PSO find the optimum in a much faster way than with the semi-exhaustive approach when the number of sensors starts to increase. Finally, PSO tends to be substantially faster on these scenarios.

Table 3. Sensor placement in the Hanoi WDN using stochastic approaches and a single time instant.

Sensors	**Node Indexes**		**Overlaps**		**Efficiency (%)**		**Time (s)**	
	GA	**PSO**	**GA**	**PSO**	**GA**	**PSO**	**GA**	**PSO**
2	12, 21	12, 21	5	5	93.1	93.1	69.9	17.3
3	12, 21, 27	12, 14, 21	1	1	98.6	98.6	113.9	38.0
4	1, 12, 21, 29	1, 12, 21, 24	0	0	100.0	100.0	119.9	57.5

GA and PSO approaches were also applied to the Hanoi example when considering a 24-h time horizon to build the LSS model. Results are shown in Table 4, where it can be seen that the efficiency tends to increase compared to the tests based on a single time instant, but only slightly and at the expense of an important increase in computational cost.

Table 4. Sensor placement in the Hanoi WDN using stochastic approaches and a time horizon analysis.

Sensors	**Node Indexes**		**Overlaps**		**Efficiency (%)**		**Time (s$\times 10^3$)**	
	GA	**PSO**	**GA**	**PSO**	**GA**	**PSO**	**GA**	**PSO**
2	12, 21	12, 21	7	7	93.5	93.5	2.2	0.5
3	12, 14, 21	12, 14, 21	1.08	1.08	98.8	98.8	4.1	1.2
4	1, 12, 21, 27	1, 12, 21, 27	0.08	0.08	100.0	100.0	3.7	1.3

5.2.2. Tests on the Limassol WDN

GA and PSO approaches were also applied to the more complex case of the Limassol network. The results obtained considering only a single time instant are shown in Table 5.

First, one can see that with two sensors, both GA and PSO find the optimum solution already found with the semi-exhaustive method. Moreover, in the case of three sensors, solutions found by means of these two methods are better than in the case of the 2 + 1 semi-exhaustive approach presented in the previous subsection and with a lower computational cost. Besides, it should be remarked that with the considered settings, PSO works more quickly than GA, but when the problem increases, *i.e.*, when more sensors are installed and more combinations are possible, GA finds better combinations. From additional experiments not reported in the table, we have seen that even by increasing the number of iterations, the PSO tends to be trapped in a local suboptimum. Such behavior has already been reported in the literature [38]. It may be explained by the fact that PSO has the memory of past successes and tends to explore around the recorded configurations, whereas when it is necessary to leap from one region to another distant region, crossover operations like those in GA are probably preferred [39]. Besides, it should be remarked that with the considered settings, PSO works more quickly than GA, but when the problem increases, *i.e.*, when more sensors are installed and more combinations are possible, GA finds better combinations. From additional experiments not reported in the table, we have seen that even by increasing the number of iterations, the PSO tends to be trapped in a local suboptimum. Such behavior

has already been reported in the literature [38]. This may be explained by the fact that PSO has the memory of past successes and tends to explore around the recorded configurations, whereas when it is necessary to leap from one region to another distant region, crossover operations, like those in GA, are probably preferred [39].

Table 5. Sensor placement in the Limassol WDN using stochastic approaches and a single time instant.

Sensors	Node Indexes		Overlaps		Efficiency (%)		Time (s×10^3)	
	GA	**PSO**	**GA**	**PSO**	**GA**	**PSO**	**GA**	**PSO**
2	8, 152	8, 152	253	253	60.6	60.6	11.0	3.3
3	75, 120, 152	76, 115, 152	145	151	74.1	74.0	19.6	6.0
4	76, 82, 120, 152	69, 82, 119, 152	114	131	78.0	76.2	31.2	15.1
5	75, 82, 120, 126, 152	70, 115, 125, 152, 188	106	122	79.0	77.4	36.8	17.4

GA and PSO were also applied to the Limassol network while incorporating a time horizon analysis. However, only the case of two and three sensors was considered, since the computational time becomes prohibitive for a higher number of sensors. Moreover, for the GA method, only one iteration it is authorized and 15 iterations for the PSO instead of the 10 and 150 iterations used respectively for the single time instant case. Results are summarized in Table 6. It shows that contrary to the Hanoi case, for this more complex scenario, the incorporation of a time horizon significantly improves the efficiency. It is remarkable to see that the LSS-based leak location method relying on only three sensors and the GA method is able to locate almost 85% of the leaks correctly. However, the time horizon analysis clearly comes at the expense of an important additional computational cost.

Table 6. Sensor placement in the Limassol WDN using stochastic approaches and a time horizon analysis.

Sensors	Node Indexes		Overlaps		Efficiency (%)		Time (s×10^3)	
	GA	**PSO**	**GA**	**PSO**	**GA**	**PSO**	**GA**	**PSO**
2	124, 153	124, 153	424.8	424.8	77.7	77.7	32.3	27.5
3	76, 133, 169	75, 122, 117	289.8	306.2	84.9	84.8	82.4	66.4

5.3. Optimization Criterion and Leak Detection Efficiency

Whatever the optimization resolution method used, the optimization problem presented above relies on the minimization of intersections between leak signature domains. However, it is important to verify that decreasing the number of intersections effectively improves the leak location efficiency. To be relevant, comparisons must be made on scenarios involving the same number of sensors to place, since otherwise, the number of overlapping domains to be checked would be different. Figure 3 presents for different number of sensors placed in the Limassol network, the variation of leak location efficiency as a function of the number of overlaps found. These values are taken directly from the results of the optimization methods or by manually choosing combinations of nodes already appearing within the set of solutions of these methods. This figure clearly illustrates the negative correlation between the efficiency and the number of overlaps found. It is important to highlight how the performance of the correct sensor placement clearly improves the efficiency of the leak location method. In this graph, we can see that a higher efficiency can be reached with four sensors when they are optimally placed, whereas such efficiency will not be achieved using five sensors if they are not optimally installed. Thanks to this analysis, the importance of optimizing the sensor placement according to the leak location method to be used is proven.

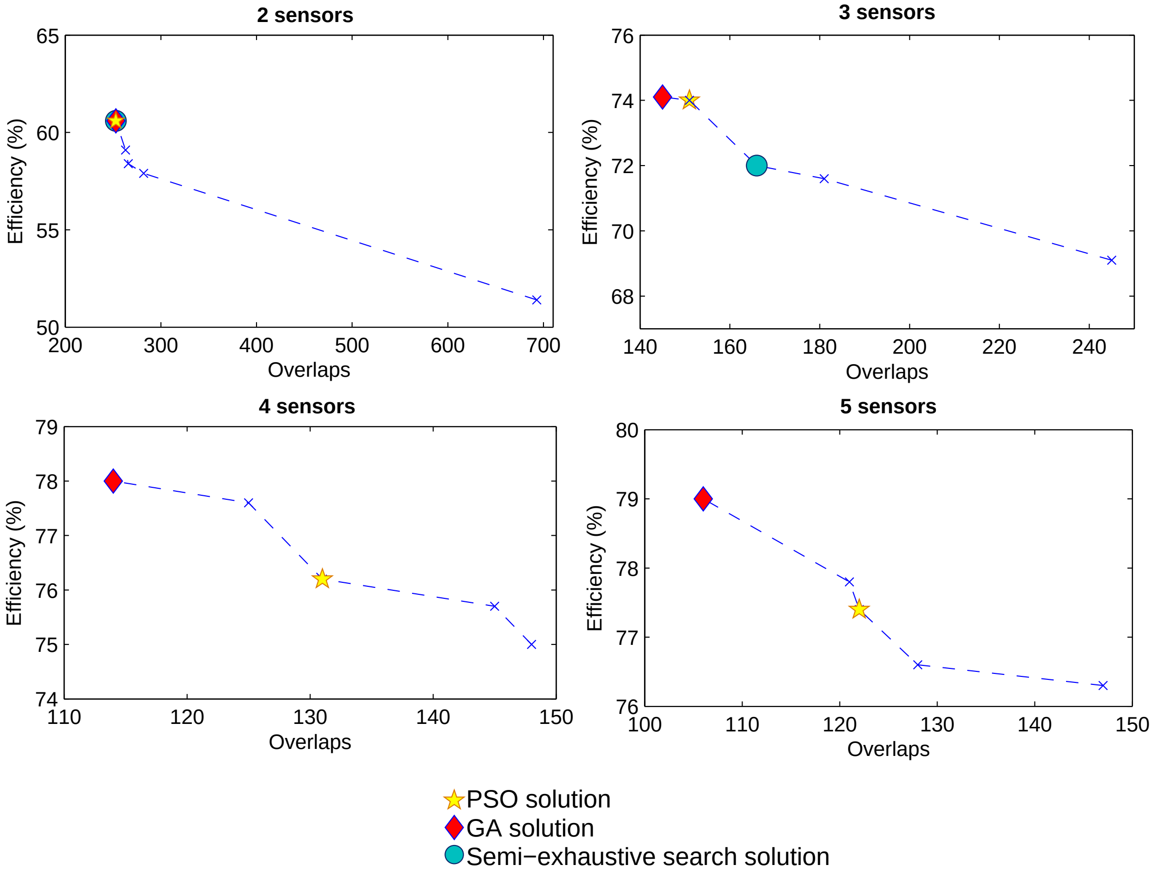


Figure 3. Efficiency in leak location according to the number of overlaps in the configuration for the Limassol network.

6. Conclusions

This paper describes a new approach to place a given set of sensors in a WDN in order to maximize the efficiency of the leak detection. The solution is motivated by the application of an original leak representation method based on what is known as the leak signature space. A semi-exhaustive method based on a lazy evaluation mechanism is proposed to solve low complexity problems. For more complex scenarios, either a GA or a PSO method is applied to the underlying optimization problem. Results demonstrate the validity of the approach. They also show that with the parameters selected for the stochastic optimization methods, PSO tends to give results faster than the GA approach and is very effective for small networks or few sensors. However, the solutions found by the GA tend to correspond to better placements and with higher efficiency for larger networks or more sensors. It also appears that a time horizon analysis can significantly improve the performance in the case of complex scenarios. Finally, the negative correlation between the optimization function and the efficiency of the leak detection method is verified.

As future work, we would like to extend this work to more complex WDN, such as the Barcelona network used in [11], consisting of 3320 nodes. Furthermore, the location of smaller leaks would be interesting to address, taking into account that they can be easily masked by the noise in the sensors. Finally, we have seen that even with the stochastic optimization tools proposed in this paper, the application of the time horizon analysis was not tractable for the Limassol network with more than three sensors. Then, we would like to investigate the possible application of a parallel computation in a multi-core processor, combined with the possibility of an accurate tuning method for the selection of the parameters used in the GA and PSO approaches. Furthermore, we would like to investigate new approaches to tackle this limitation and obtain an improved leak location efficiency.

Acknowledgments

The authors wish to thank the support from the Research Chair in Supervision and Advanced Control of Tecnológico de Monterrey, Campus Monterrey. This work has also been partially funded by the Spanish Ministry of Science and Technology through the Project ECOCIS (Ref. DPI2013-48243-C2-1-R) and the Project HARCRICS (Ref. DPI2014-58104-R) and by EFFINET Grant FP7-ICT-2012-318556 of the European Commission.

Author Contributions

All authors contributed to the approach development. Myrna V. Casillas performed the experiments. All authors wrote the paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Brothers, K.J. A practical approach to water loss reduction. *Water 21 IWA Task Force* **2003**, *1*, 54–55.
2. World-Health-Organization, UNICEF. Accounting for Water Loss. In *Global Water Supply and Sanitation Assessment 2000 Report*; World-Health-Organization, UNICEF: New York, NY, USA, 2000; p. 80.
3. Hamilton, S.; Charalambous, B. Leak detection technologies. *Leak Detect.: Technol. Implement.* **2013**, *1*, 104.

4. Puust, R.; Kapelan, Z.; Savic, D.A.; Koppel, T. A review of methods for leakage management in pipe networks. *Urban Water J.* **2010**, *7*, 25–45.
5. Wu, Z.Y.; Farley, M.; Turtle, D.; Dahasahasra, S.; Mulay, M.; Boxall, J.; Mounce, S.; Kleiner, Y.; Kapelan, Z. Water Loss Reduction. In *Water Modeling and Water Loss Management*; Bentley Systems: Pennsylvania, PA, USA, 2011.
6. Colombo, A.F.; Lee, P.; Karney, B.W. A selective literature review of transient-based leak detection methods. *J. Hydro-Environ. Res.* **2009**, 2, 212–227.
7. Pudar, R.S.; Liggett, J.A. Leaks in Pipe Networks. *J. Hydraul. Eng.* **1992**, *118*, 1031–1046.
8. Pérez, R.; Puig, V.; Pascual, J.; Quevedo, J.; Landeros, E.; Peralta, A. Methodology for leakage isolation using pressure sensitivity analysis in water distribution networks. *Control Eng. Pract.* **2011**, *19*, 1157–1167.
9. Perez, R.; Sanz, G.; Puig, V.; Quevedo, J.; Cuguero Escofet, M.; Nejjari, F.; Meseguer, J.; Cembrano, G.; Mirats Tur, J.; Sarrate, R. Leak Localization in Water Networks: A Model-Based Methodology Using Pressure Sensors Applied to a Real Network in Barcelona [Applications of Control]. *Control Syst. IEEE* **2014**, *34*, 24–36.
10. Casillas, M.V.; Garza-Castañón, L.; Puig, V. Extended-Horizon Analysis of Pressure Sensitivities for Leak Detection in Water Distribution Networks. In Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Mexico City, Mexico, 29–31 August 2012; pp. 570–575.
11. Casillas, M.V.; Garza-Castañón, L.; Puig, V. Model-based Leak Detection and Location in Water Distribution Networks considering an Extended-horizon Analysis of Pressure Sensitivities. *J. Hydroinform.* **2014**, *16*, 649–670.
12. Worden, K.; Burrows, A. Optimal sensor placement for fault detection. *Eng. Struct.* **2001**, *23*, 885–901.
13. Yassine, A.; Ploix, S.; Flaus, J.M. A Method for Sensor Placement Taking into Account Diagnosability Criteria. *Int. J. Appl. Math. Comput. Sci.* **2008**, *18*, 497–512.
14. Krysander, M.; Frisk, E. Sensor Placement for Fault Diagnosis. *Syst. Man Cybernetics Part A: Syst. Hum. IEEE Trans.* **2008**, *38*, 1398–1410.
15. Krause, A.; Leskovec, J.; Guestrin, C.; Vanbriesen, J.; Faloutsos, C. Efficient sensor placement optimization for securing large water distribution networks. *J. Water Resour. Plan. Manag.* **2008**, *134*, 516–526.
16. Aral, M.; Guan, J.; Maslia, M. Optimal Design of Sensor Placement in Water Distribution Networks. *J. Water Resour. Plan. Manag.* **2010**, *136*, 5–18.
17. Multiobjective Sampling Design for Water Distribution Model Calibration. *J. Water Resour. Plan. Manag.* **2003**, *129*, 466–479.
18. Sanz, G.; PÃl'rez, R. Sensitivity Analysis for Sampling Design and Demand Calibration in Water Distribution Networks Using the Singular Value Decomposition. *J. Water Resour. Plan. Manag.* **2015**, *141*, 04015020.
19. Sarrate, R.; Nejjari, F.; Rosich, A. Sensor Placement for Fault Diagnosis Performance Maximization in Distribution Networks. In Proceedings of the 18th Mediterranean Conference on Control and Automation (MED), Barcelona, Spain, 3–6 July 2012; pp. 1–6.

20. Christodoulou, S.; Gagatsis, A.; Xanthos, A.; Kranioti, S.; Agathokleous, A.; Fragiadakis, M. Entropy-Based Sensor Placement Optimization for Waterloss Detection in Water Distribution Networks. *Water Resour. Manag.* **2013**, *27*, 4443–4468.
21. Casillas, M.V.; Puig, V.; Garza-Castañón, L.E.; Rosich, A. Optimal Sensor Placement for Leak Location in Water Distribution Networks Using Genetic Algorithms. *Sensors* **2013**, *13*, 14984–15005.
22. Casillas, M.V.; Garza-Castañón, L.E.; Puig, V.; Vargas-Martinez, A. Leak Signature Space: An Original Representation for Robust Leak Location in Water Distribution Networks. *Water* **2015**, *7*, 1129–1148.
23. Todini, E.; Pilati, S. Computer applications in water supply: volume 1—systems analysis and simulation. In *Chapter A Gradient Algorithm for the Analysis of Pipe Networks*; Research Studies Press Ltd.: Taunton, UK, 1988; pp. 1–20.
24. Rossman, L. *EPANET 2 User's Manual*; United States Envionmental Protection Agency: Cincinnati, OH, USA, 2000.
25. Goldberg, D. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesle: Reading, MA, USA, 1989.
26. Koza, J.R. Survey of Genetic Algorithms and Genetic Programming. In Proceedings of the Wescon 95—Conference Record: Microelectronics, Communications Technology, Producing Quality Products, Mobile and Portable Power, Emerging Technologies, San Francisco, CA, USA, 7–9 November 1995; pp. 589–594.
27. Elsayed, S.M.; Sarker, R.A.; Essam, D.L. A new genetic algorithm for solving optimization problems. *Eng. Appl. Artif. Intell.* **2014**, *27*, 57–69.
28. Chipperfield, A.; Fleming, P.; Pohlheim, H.; Fonseca, C. *Genetic Algorithm TOOLBOX For Use with MATLAB*; *Technical Report*; Department of Automatic Control and Systems Engineering, University of Sheffield: South Yorkshire, UK, 1994.
29. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
30. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization. *Swarm Intell.* **2007**, *1*, 33–57.
31. Kuila, P.; Jana, P.K. Energy efficient clustering and routing algorithms for wireless sensor networks: Particle swarm optimization approach. *Eng. Appl. Artif. Intell.* **2014**, *33*, 127–140.
32. Chang, B.M.; Tsai, H.H.; Shih, J.S. Using fuzzy logic and particle swarm optimization to design a decision-based filter for cDNA microarray image restoration. *Eng. Appl. Artif. Intell.* **2014**, *36*, 12–26.
33. Chen, S. Another Particle Swarm Toolbox; MATLAB Central File Exchange. Available online: http://www.mathworks.com/matlabcentral/fileexchange/25986-another-particle-swarm-toolbox (accessed on 29 October 2015).
34. Fujiwara, O.; Khang, D.B. A two-phase decomposition method for optimal design of looped water distribution networks. *Water Resour. Res.* **1990**, *26*, doi:10.1029/WR026i004p00539.

35. Bolognesi, A.; Bragalli, C.; Marchi, A.; Artina, S. Genetic Heritage Evolution by Stochastic Transmission in the optimal design of water distribution networks. *Adv. Eng. Softw.* **2010**, *41*, 792–801.
36. Zheng, F.; Zecchin, A.; Simpson, A. Self-Adaptive Differential Evolution Algorithm Applied to Water Distribution System Optimization. *J. Comput. Civ. Eng.* **2013**, *27*, 148–158.
37. Effinet-project. EFFINET. Available online: http://effinet.eu/ (accessed on 29 October 2015).
38. Perez, R.; Behdinan, K. Particle swarm approach for structural design optimization. *Comput. Struct.* **2007**, *85*, 1579–1588.
39. Kennedy, J.; Eberhart, R. A discrete binary version of the particle swarm algorithm. Systems, Man, and Cybernetics, 1997. In Proceedings of the 1997 IEEE International Conference on Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; Volume 5, pp. 4104–4108.