water

MDPI

Article

Serious Sensor Placement—Optimal Sensor Placement as a Serious Game

Georg Arbesser-Rastburg * and Daniela Fuchs-Hanusch

Institute of Urban Water Management and Landscape Water Engineering, Graz University of Technology, Stremayrgasse 10/I, 8010 Graz, Austria; fuchs-hanusch@tugraz.at

* Correspondence: georg.arbesser-rastburg@tugraz.at

Received: 31 October 2019; Accepted: 19 December 2019; Published: 23 December 2019

Abstract: In this paper, we present a novel approach in water loss research combining two different topics: The optimal placement of pressure sensors to localize leaks in water distribution systems and Serious Gaming—games that are not only entertaining but that are also serving another purpose. The goal was to create a web interface, through which gamers could place sensors in a water distribution system model, in order to improve these sensor positions after they had been evaluated by a suitable algorithm. Two game objectives are to be pursued by the players: reaching a specified net coverage while not using more than a maximum number of sensors. For this purpose, an existing optimal sensor placement algorithm was extended and implemented, together with two hydraulic models taken from literature. The resulting Serious Game was then tested and rated in a case study. The results showed that human players are able to reach solutions that are similar regarding net coverage to those obtained by optimization, within in a short amount of time. Furthermore, it was shown that the implementation of the ideal sensor placement problem as a Serious Game motivates the players to get better and better results, while also providing them with an enjoyable gaming experience.

Keywords: serious gaming; water distribution systems; water loss reduction; leak localization

1. Introduction

Optimal Sensor Placement (OSP) algorithms have been developed for various purposes and different types of sensors, often making use of hydraulic models, to find ideal measurement positions in a water distribution system (WDS). Examples are the placement of pressure sensors for pipe roughness calibration [1] and quality sensors for water quality monitoring [2].

In the field of water loss detection and leak localisation, OSP algorithms are used to find the ideal number and the optimal positions of flow or pressure sensors. Many OSP algorithms use the sensitivity matrix as described by Pudar and Ligget in 1992 [3]; for instance, Farley et al. who were the first to develop a methodology to place pressure sensors for leak localization in 2008. They evaluated simulated leaks in a water distribution system and instantaneous Chi Squared Values of pressure changes. The Chi Squared Values are then separated into detected and undetected leaks using a threshold which leads to a binarized sensitivity matrix [4,5]. In 2013, this approach was extended by introducing a punishment of sensor positions with a response close to the threshold to find positions with a more clear response [6].

In 2012, Sarrate et al. used structural analysis of WDS for solving the OSP problem. They introduced a leak isolability index and used Depth-First-Search. However, this algorithm only works for medium sized network [7]. Therefore Sarrate et al. extended their algorithm by applying graph clustering to reduce the problem complexity [8,9]. In 2014, Sarrate et al. combined their methodology with the leak sensitivity matrix [10].

Pérez et al. also made use of a binarized sensitivity matrix to choose sensor positions that lead to as many unique leak signatures as possible in 2009 [11]. However, binarizing the sensitivity matrix leads to a loss in information [12].

In 2013, Casillas et al. subsequently used a non-binarized sensitivity matrix to calculate projections between the sensitivity matrix and a residual matrix, which includes a substitute for real-world measurement data, introducing an error index to minimize the number of incorrectly localized leaks. This error index then also served as objective function [13]. Casillas et al. proposed another algorithm in 2015, where overlapping leak signatures in the leak signature space are to be minimized [14].

Pérez et al. developed an OSP algorithm in 2014 to obtain the optimal sensor positions when adding new sensors to an already existing sensor network by evaluating the leak localization results. For this purpose, the maximum distance to pre-calculated leak scenarios and a gravity center of nodes with a correlation of more than 99% of the maximum correlation is minimized [15].

Cugueró-Escofet et al. developed another sensitivity-based algorithm in 2015 using a relaxed isolation index. Where Casillas et al. aimed to correctly localize as many leaks as possible, Cugueró-Escofet et al. tried to create clusters of geographically close nodes with similar leak signatures [16].

Steffelbauer et al. extended the algorithm by Casillas et al. in 2016 by introducing a punishment for nodes with a high pressure uncertainty due to uncertain demands. The resulting algorithm (Sensor Placement under Demand Uncertainties—SPuDU) uses an error index analogous to the one defined by Casillas et al. as objective function [17].

In 2018, Soldevilla et al. proposed an OSP algorithm using hybrid feature selection that tries to maximize the isolability of leaks when using classifier-based leak localization methods [18].

In 2019, Righetti et al. used a sensitivity analysis of pressure data generated using hydraulic simulations and a correlation analysis to select nodes as sensor positions that are, on the one hand, sensitive to leaks in the system and, on the other hand, uncorrelated with each other to ensure as little redundant information as possible is gathered by the sensors [19].

The term *Serious Game* was first introduced by Clark C. Abt in 1970 in his book *Serious Games*. He differentiated between serious and casual games. In his terminology, a serious game is a game that primarily serves educational and training purposes and not entertainment [20].

To this day, there is no universally accepted definition of the term serious game [21]. A more modern definition is the following:

> A serious game is a digital game created with the intention to entertain and to achieve at least one additional goal (e.g., learning or health). [21]

Following this definition, a serious game has to be a digital game; the additional goals, however, do not have to be in an educational context. Therefore, other areas are covered by this definition as well. Furthermore, the developer's intentions are sufficient to apply the term Serious Game, the goals do not have to be met.

Serious Games have already been developed for different fields, i.e., education [22], healthcare [23] and climate change [24] but also for flood and river basin management [25–27] and draughts [28].

Several serious games have been developed in the field of water distribution, as well. Three of them will be described in the following.

The first presented Serious Game is *Aqualibrium* [29], which is the digital version of the real-life competition of the same name. During the competition, 0.28 m long pipes of two different diameters (3 and 6 mm) have to be placed in a grid connecting three empty containers at predefined positions with an inflow point (a container filled with water placed on a stand). The goal is to fill the empty containers equally with one liter of water coming from the inflow point. At least 16 pipes have to be placed at the 24 possible pipe locations, while not creating any dead end pipes. At the end of the competition, the water in the three containers is measured using measuring cylinders or weight scales. The difference in water volume is transformed into penalty points, leading to a ranking of the competitors [30]. The digital version follows the same rules; however, the system evaluation is based on hydraulic simulation and is run by pressing the corresponding button in the web interface. Already evaluated solutions can be saved and then again loaded afterwards.

SeGWADE (Serious Game for Water Distribution System Analysis, Design, and Evaluation) [29] deals with the New York tunnel problem. Several nodes in an existing water distribution system do not reach a certain minimum service pressure; therefore, parallel pipes have to be added using diameters between 36 in (0.9 m) and 204 in (5.2 m). The network topology and the resulting different pipe lengths in combination with the chosen diameters lead to different costs that players should keep minimal, while still achieving the predefined minimum pressure at all nodes in the system [31–33]. SeGWADE's user interface presents itself similar to Aqualibrium's interface, also offering the possibility to save at any point and go back to a saved system variant.

Network Pipe Sizing is in some aspects similar to SeGWADE. In this game, players have the goal of redimensioning different water distribution systems using a set of different pipe diameters, while also trying to reduce the costs and ensuring that a certain minimum pressure is met throughout the system. Five different networks (i.e., levels in the game) can be successively played with an increase in network size and complexity from level to level [34].

In this paper, we present a novel approach to the OSP problem, mostly abstaining from any elaborate optimization algorithm but by making use of human intrinsic motivation for the optimization process. Therefore, we introduce a new Serious Game called Serious Sensor Placement (SSP) [35]. SSP is meant to be an example of the implementation of an optimization problem in WDS as a Serious Game that can be used for teaching students about optimization problems, algorithms, and especially the optimal sensor placement problem. Furthermore, it can be used for demonstrating optimization tasks in WDS to a wider audience. Since SSP is conceptualized as a Serious Game, it is also supposed to provide any player with an enjoyable gaming experience, independent from their level of knowledge concerning WDS.

This paper is structured as follows. In Section 2, we introduce the principles of SSP and describe an already conducted case study before the case study results are compared to optimization results obtained with the optimization algorithm Differential Evolution (DE) in Section 3. Finally, conclusions are drawn in Section 4.

2. Methods and Materials

In this section, a methodology to calculate a net coverage of a given set of sensors regarding leak localization is presented first, followed by a description of the new Serious Game Serious Sensor Placement and the conducted case study.

2.1. Net Coverage Calculation

To measure the performance of a set of sensor positions a net coverage in percent of pipes, on which leaks could be localized correctly, was introduced. This net coverage is calculated based on the OSP algorithm by Casillas et al. from 2013. This algorithm was chosen because it allows an explicit distinction between localizable and non-localizable leaks. SPuDU would have allowed for this, as well, but only at the cost of higher complexity following the introduction of punishment of nodes with high pressure uncertainties. Since the methodology of SSP uses the Casillas algorithm, this algorithm is described in the following.

The OSP algorithm by Casillas et al. is based on a leak localization method, where pressure measurements are compared to pre-calculated simulation results using projections. For the computation of these simulation results, leaks are simulated at all possible leak locations one at a time (i.e., leak scenarios). The leak scenario that leads to the best match with the measurement data is then assumed to be the location of the real leak.

To find optimal sensor positions, the OSP algorithm itself uses a sensitivity matrix S representing the sensitivity of nodes to leaks in the WDS and a residual matrix R that contains simulated pressure data as a substitute for real world pressure measurements.

The sensitivity matrix as presented by Pudar and Liggett [3] comes in the form:

$$S = \begin{pmatrix} s_{11} & \dots & s_{1n} \\ \vdots & \ddots & \vdots \\ s_{m1} & \dots & s_{mn} \end{pmatrix}, \tag{1}$$

where m is the number of possible leak scenarios, and n the number of possible sensor positions. The leak scenarios are calculated using hydraulic simulations where leaks are one at a time simulated at all possible leak locations. Therefore, at position i a leak of size f is simulated, and the resulting pressure at position j, $p_j^{f_i}$, is compared to the corresponding pressure in the original, undisturbed model, $\hat{p}_j$. This pressure difference is then normalized according to the leak size resulting in

$$s_{ij} = \frac{p_j^{f_i} - \hat{p}_j}{f_i}. \tag{2}$$

Furthermore, a mean node sensitivity for position j, $\bar{s}_j$, can be calculated using the following equation:

$$\bar{s}_j = \frac{1}{m} \sum_{i=1}^{m} s_{ij}. \tag{3}$$

The residual matrix is calculated in a similar way. The values in this matrix, however, are not normalized and, for higher robustness, a different leak size can be used. The components of this matrix are, therefore, calculated as

$$r_{ij} = p_j^{f_i} - \hat{p}_j. \tag{4}$$

Next, a binary vector q of length n is constructed, where $q_i = 0$ if no sensor is placed at position i, and $q_i = 1$ if otherwise:

$$q = [q_1, \dots, q_n]. \tag{5}$$

Using this vector, a diagonal matrix $Q(q)$ is built:

$$Q(q) = diag(q_1, \dots, q_n). \tag{6}$$

To calculate the correlation between the residual and sensitivity matrix for a given set of sensor positions the normalized projections of the residual matrix onto the sensitivity matrix are calculated, resulting in a projection matrix ψ with its individual components ψ_{ij} calculated as

$$\psi_{ij} = \frac{r_i^T Q(q) s_j}{|r_i Q(q)^T||Q(q) s_j|}, \tag{7}$$

where r_i^T is called the transposed residual vector r_i (i.e., leak scenario i of the residual matrix) and s_j the sensitivity vector j (i.e., leak scenario j of the sensitivity matrix). The values in this matrix represent the correlation between the "measured" data and the simulated data. If a leak is correctly localized by the algorithm the correlation of residual vector i and sensitivity vector j must be the highest if $i = j$. Therefore, the values situated at the principle diagonal should be the highest value of each row in the projection matrix.

To evaluate a set of sensor positions, an error index vector $\epsilon(q)$ is used, where $\epsilon_i = 0$ if the leak scenario i can be correctly localized at position i, and $\epsilon_i = 1$ if otherwise:

$$\epsilon_i(q) = \begin{cases} 0 & \text{if } \psi_{ii}(q) = max(\psi_{i1}(q), \dots, \psi_{im}(q)) \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

Finally, a mean error index $\overline{\epsilon}(q)$, which also serves as objective function, is calculated:

$$\overline{\epsilon}(q) = \sum_{i=1}^{m} \frac{\epsilon_i(q)}{m} \rightarrow \min_q \overline{\epsilon}(q) \qquad . \tag{9}$$

$\overline{\epsilon}(q)$, therefore, states the ratio of not correctly localized leaks and should be minimized using a suitable optimization algorithm.

A net coverage $\nu(q)$ is introduced that represents the percentage of correctly localized leaks in the system:

$$\nu(q) = (1 - \overline{\epsilon}(q)) * 100. \tag{10}$$

In EPANET [36], leaks are simulated at nodes by increasing the nodal demand or by using a pressure dependent emitter exponent. In SSP, the net coverage is expected to describe the number of pipes in the system, where leaks can be localized correctly. Therefore, to estimate whether a leak at a certain pipe can be localized, the following method is applied:

1. Placing new nodes (called *virtual nodes*) in the middle of each pipe.
2. Situating the sensors at the original nodes in the network.
3. Calculating the corresponding error index vector of the virtual node $\epsilon'(q)$.
4. Assigning the error index of the virtual node, $\epsilon'(q)$, to the original pipe.

Using this methodology, it is possible to calculate the net coverage achieved by a given set of sensors with regard to leaks at pipes by assigning the individual $\epsilon_i'(q)$ to each pipe. Figure 1 shows an example of this procedure.

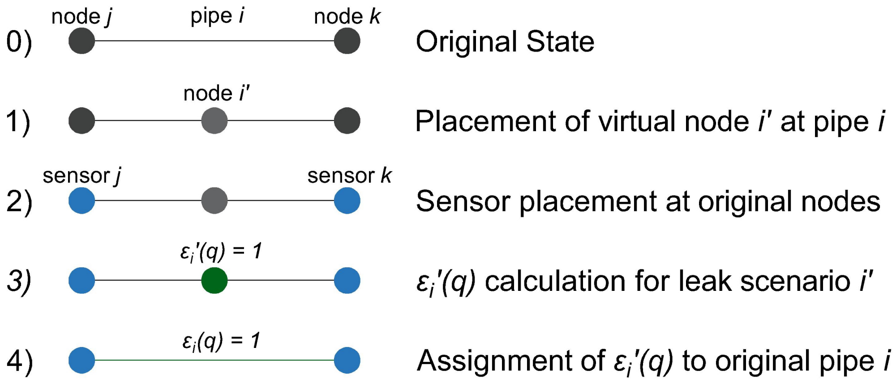

0)
node *j*
pipe *i*
node *k*
Original State
1)
node *i'*
Placement of virtual node *i'* at pipe *i*
2)
sensor *j*
sensor *k*
Sensor placement at original nodes
3)
ε_i'(q) = 1
ε_i'(q) calculation for leak scenario *i'*
4)
ε_i(q) = 1
Assignment of *ε_i'(q)* to original pipe *i*

Figure 1. Process scheme of simulating leaks on pipes for Optimal Sensor Placement calculations.

2.2. Serious Sensor Placement

In this section, the underlying software of SSP is described, as well as the implemented hydraulic models and the graphical user interface (GUI) itself.

2.2.1. Software Backbone

All pre-calculated simulation results were computed using OOPNET, an object-oriented programming interface between EPANET and Python [37]. OOPNET was used for sensitivity and residual matrix calculation, as well as calculating the achieved net coverage in SSP.

SSP's backbone itself is Flask [38], a WSGI (Web Server Gateway Interface) web framework written in Python that allows for the easy development of WSGI applications. The network is visualized using Leaflet [39], an open-source JavaScript library that was developed for displaying interactive 2D maps. Finally, PostgreSQL [40] was chosen as the results database. All solutions submitted by the players are stored in this database, with their usernames hashed for anonymity.

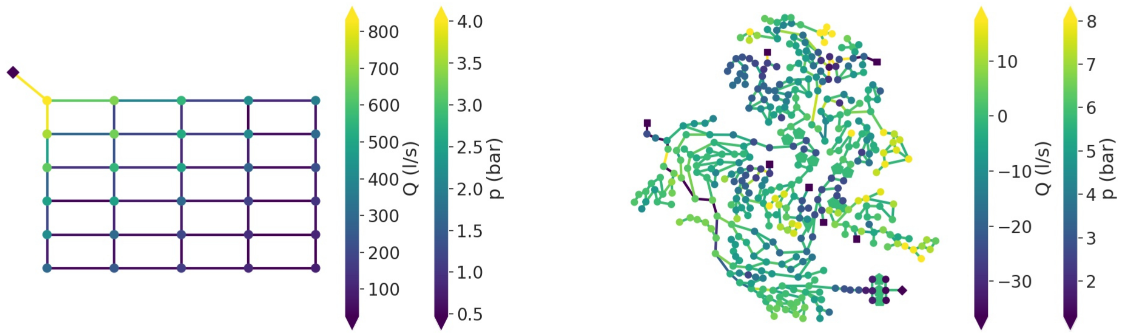
Since the computationally most demanding task, the calculation of the residual and sensitivity matrices, is done beforehand and the net coverages are calculated on a central server, SSP can be accessed and played even on devices with low processing power, like tablet computers.

2.2.2. Hydraulic Models in SSP

Two different hydraulic models have been implemented in SSP so far:

- the hydraulic model used by Poulakis et al. for demonstrating leak detection using a Bayesian probabilistic framework [41] and
- C-Town, which was developed for the Battle of Water Calibration Networks [42].

Figure 2 shows the two used hydraulic models and the occurring flows and pressures.



(**a**) Poulakis model (**b**) C-Town model

Figure 2. Plots showing the used hydraulic models with the occurring flows and pressures: (**a**) Poulakis and (**b**) C-Town model.

The Poulakis model consists of 50 pipes, 30 junctions, and one reservoir (the diamond shaped symbol in the top left corner of Figure 2a). The junctions are arranged in a 5 × 6 grid and connected by 1000 m long horizontal and 2000 m long vertical pipes. Diameters range from 300 to 600 mm, and the total length results in approximately 73 km.

C-Town is more complex than the Poulakis model, consisting of one reservoir, seven tanks (represented by squares in Figure 2b), five pumps, and one throttle control valve and one check valve, as well as 388 junctions and 432 pipes with a total length of approximately 57 km and diameters ranging from 51 to 610 mm.

The two chosen models serve different purposes: Since the Poulakis model consists of only a few links and nodes, the optimization of sensor positions is relatively simple. This allows putting the focus on the game mechanics, hence the use for demonstrating SSP or as a tutorial for first time players.

C-Town, however, is a much more complex network and was chosen to create a more demanding gaming environment.

Sensitivity and residual matrices were calculated, as shown in Section 2.1, using a fixed leak size of 2 L/s for the Poulakis model and 50 L/s for the C-Town model. No other leak sizes were tested since the optimal sensor positions in a network are not dependent on the leak size used for sensitivity and residual matrix computation [43]. The results were then exported as geoJSON files together with model component IDs, coordinates, types and status (open or closed).

For easier handling when visualizing the network using Leaflet, the sensitivity values were normalized to values from 0 (low sensitivity) to 1 (high sensitivity).

As using only normalized sensitivities for visualization purposes resulted in very few nodes showing high sensitivities, the sensitivities were finally not only normalized but also logarithmized. This was done to not overstate the absolute node sensitivities' importance, since the node sensitivity itself is not directly linked to the goodness of a sensor position.

Figure 3 shows a histogram depicting the difference between the two different normalization methods.

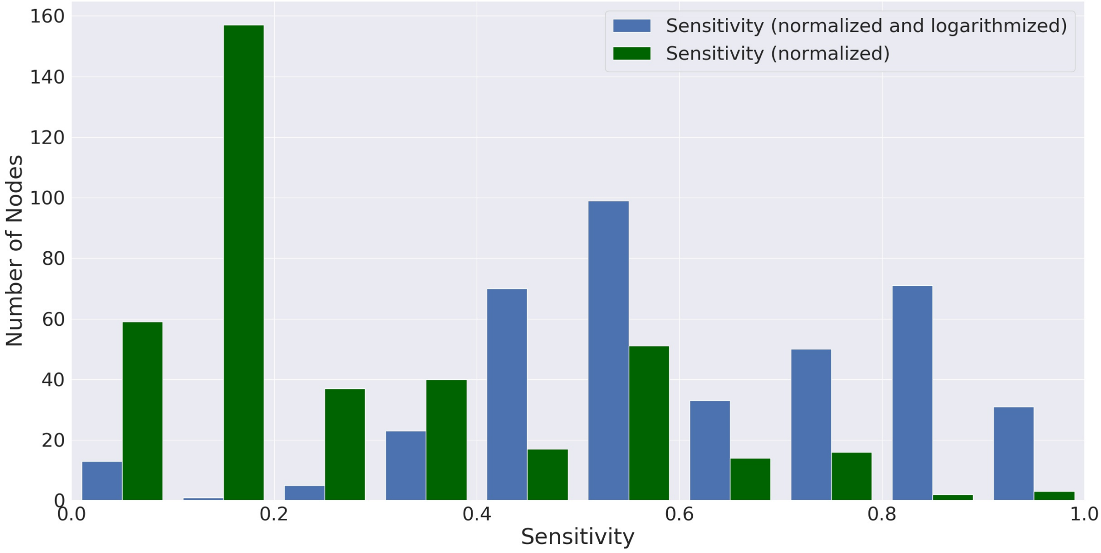


Figure 3. Histogram of normalized respectively logarithmized and normalized node sensitivities.

The normalized and logarithmized sensitivity of node j $\hat{s}_j$ is calculated using the following equation:

$$\hat{s}_j = \frac{\ln \bar{s}_j - \ln s_{min}}{\ln s_{max} - \ln s_{min}}, \tag{11}$$

where s_{min} is the minimum sensitivity, and s_{max} is the maximum sensitivity of all nodes.

2.2.3. Game Objectives

The players of SSP are supposed to pursue two objectives while playing SSP:

1. maximizing the sensor performance by finding optimal sensor positions and
2. keeping the number of placed sensors below a certain threshold.

To set these goals, OSP calculations were made, using the Casillas algorithm and Differential Evolution [44] for optimization. To get a first overview of the achieveable net coverages, the number of placed sensors was varied from two to 30. All OSP calculations were made optimizing 100 individuals over 100 generations.

Based on the optimization results, the game objectives were then defined for the C-Town model as follows:

1. Maximum number of sensors: 15 sensors
2. Minimum net coverage: 60%

Figure 4 shows the calculated net coverages, as well as the part of the solution space that fulfills both objectives. On the one hand, these goals are meant to be achievable since the minimum net coverage can be reached by placing only ten sensors but challenging, on the other hand, because the net coverages achieved by DE are only slightly above this goal. Furthermore, a higher net coverage can only be realized by placing much more sensors. A net coverage of 80%, for instance, can only be reached by using at least 29 sensors. This would pose a significantly more complex task for the players without much additional benefit.

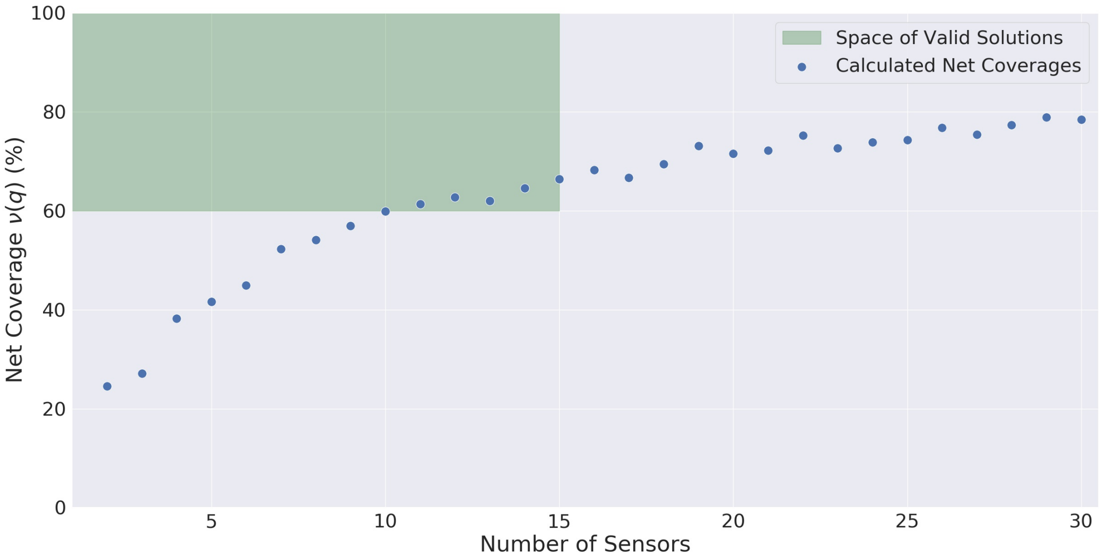
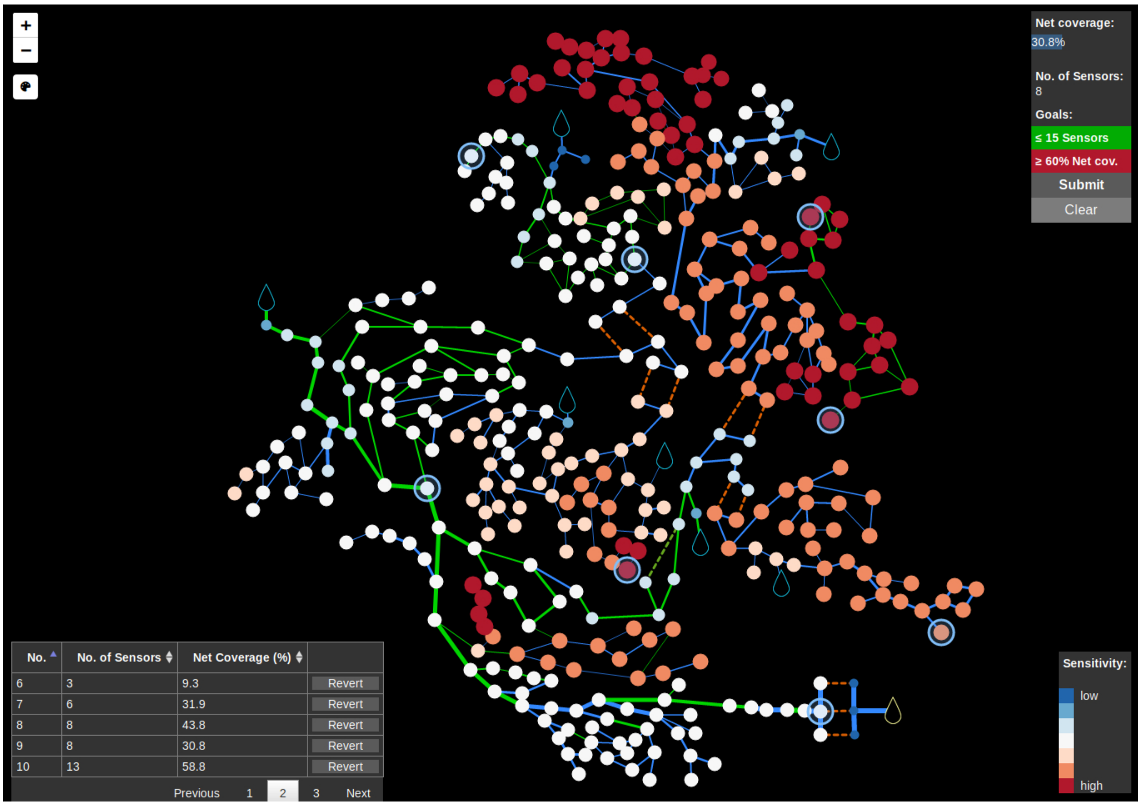


Figure 4. Calculated net coverages and space of solutions fulfilling both objectives (C-Town model).

2.2.4. Graphical User Interface

Figure 5 shows the SSP GUI when playing the C-Town model after having placed eight sensors.



Net coverage:
30.8%

No. of Sensors:
8

Goals:
≤ 15 Sensors
≥ 60% Net cov.

Submit
Clear

No.	No. of Sensors	Net Coverage (%)	
6	3	9.3	Revert
7	6	31.9	Revert
8	8	43.8	Revert
9	8	30.8	Revert
10	13	58.8	Revert

Previous 1 2 3 Next

Figure 5. Serious Sensor Placement (SSP) graphical user interface (GUI) showing the C-Town model.

The largest part of the GUI is reserved for the visualization of the hydraulic model. The model itself is visualized analogous to the EPANET models using nodes for junctions, tanks, and reservoirs and edges for pipes, valves, and pumps. Sensors can be placed at all nodes except reservoirs, which deliver water with a constant pressure and, therefore, cannot be used for leak localization based on pressure sensors. Tanks and reservoirs are depicted as water drops in either blue (tanks) or yellow (reservoirs). Line widths are either based on pipe diameters or set to a constant value for other link types (e.g., pumps).

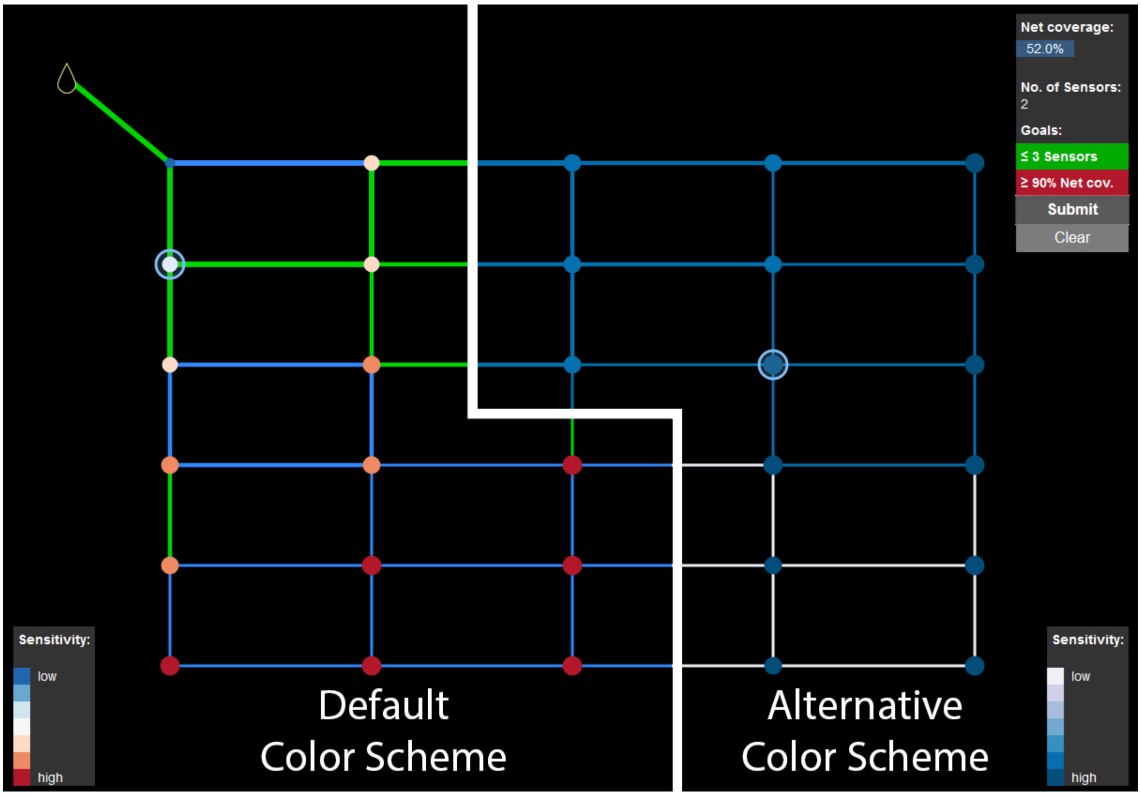
A box in the top right corner shows the number of placed sensors, the achieved net coverage resulting from the last submitted solution, and the objectives defined for the players. The button *Submit* starts the net coverage calculation and then updates the achieved net coverage and the number of placed sensors. Furthermore, the submitted sensor placement and its evaluation results are saved in a PostgreSQL database. A real time evaluation is not achievable due to computation times of about 2 s per set of sensor positions. The minimal net coverage and maximum number of sensors are displayed in front of a red background as long as they are not met. When a player meets one objective, the corresponding background changes to green. The *Clear* button removes all placed sensors and resets the number of placed sensors and the achieved net coverage.

The table in the bottom left corner allows one to revisit previously submitted solutions. The table shows a consecutive index, the achieved net coverage, and the number of sensors used. Players can sort the table by all three columns in either ascending or descending order. A fourth column contains a *Revert* button per row that players can use to reload the solution in the corresponding line.

In the top left corner of the screen, two buttons for zooming in and out and a button for switching between two color schemes (a Leaflet EasyButton [45] showing a Font Awesome icon [46]) are arranged. The default color scheme shows pipes in either green or blue, depending whether leaks on this pipe can be correctly localized or not. The alternative color scheme uses dark blue and white for this purpose. Node color and diameter depend on the pre-calculated nodal sensitivities as described in Equation (11). The default color scheme uses a color gradient ranging from dark blue (low sensitivity) to white (medium sensitivity) to dark red (high sensitivity). Alternatively, a color gradient from light grey (low sensitivity) to dark blue (high sensitivity) is used. This alternative color scheme is solely based on color contrast and is especially tailored to persons with impaired color vision. Furthermore, nodes with a low sensitivity present themselves with smaller diameter than nodes with a high sensitivity. The node diameter of node j, r_j, is calculated as follows:

$$r_j = \lfloor r_{min} + \hat{s}_j * (r_{max} - r_{min}) + 0.5 \rfloor, \tag{12}$$

where r_{min} and r_{max} are predefined maximum and minimum node diameters, and $\hat{s}_j$ is the normalized logarithmized sensitivity of node j. Figure 6 shows the two different color schemes using the Poulakis model after having placed and evaluated two sensors.



(**a**) GUI with default color scheme (**b**) GUI with alternative color scheme

Figure 6. Poulakis model in SSP using (**a**) the default color scheme and (**b**) the alternative color scheme.

2.3. Case Study

SSP was tested in a case study by 12 players. The testing group consisted of ten students, all working in the field of urban water management, with six focusing on water distribution systems. Furthermore, one programmer and one professor, who also specialized on water distribution system analysis, took part in this case study. The level of knowledge concerning WDS research, the optimal sensor placement problem, and mathematical optimization in general, therefore, varied in the testing group.

During the case study, the participants were first informed about the principal idea behind OSP. SSP itself was then introduced using the Poulakis model. It was demonstrated how to place and evaluate sensor positions. Further, the different control elements of SSP, like the buttons for changing the color scheme, were explained. Afterwards, the testers had 20 min to find ideal sensor positions in the C-Town model. Finally, the test persons were asked to anonymously evaluate SSP using a questionnaire.

Four different criteria were evaluated in the manner of school grades from 1 to 5 (lower is better):

1. Web Interface Clarity
2. Operating Elements Intuitivity
3. Gaming Fun
4. Motivation to reach better results

3. Results and Discussion

In this section, the optimization results obtained with DE and the results generated during the case study are compared, and the evaluation questionnaire results are discussed.

3.1. Optimization and Case Study Results

To mitigate the effects of DE's stochastic approach, ten further OSP optimization runs for the C-Town model were done per number of sensors, for two to 15 sensors, resulting in 140 optimization results. The results of these and the previous OSP optimization runs are jointly depicted in Figure 7 using boxplots. Several of the OSP calculation reruns resulted in higher net coverages than the ones shown in Figure 4. One of the optimization runs for $n = 15$ even reached a net coverage of 74.5%. Some optimizations runs for placing nine sensors also fulfilled the minimum net coverage goal of 60%.

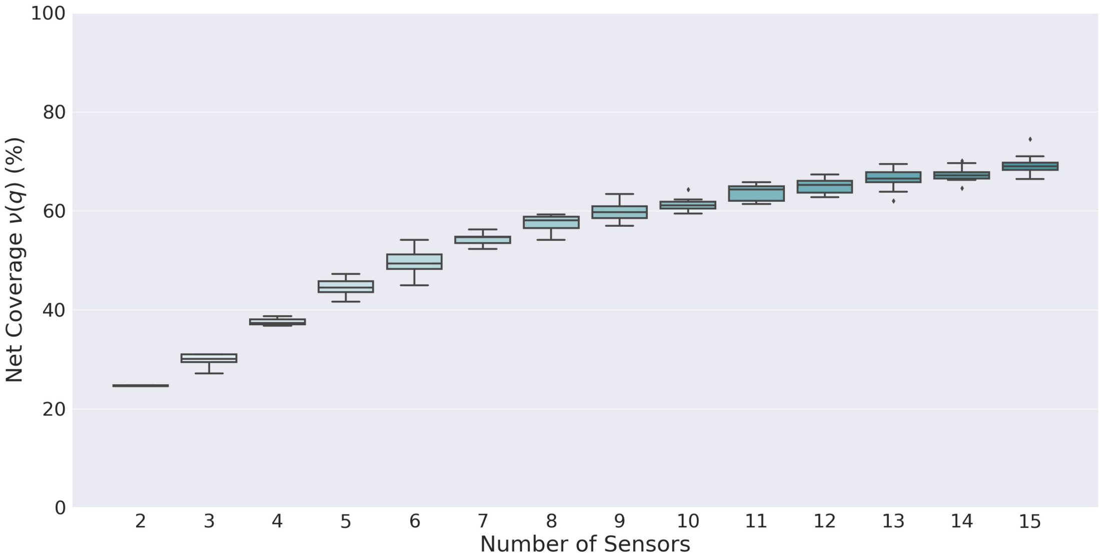


Figure 7. Result of the OSP calculation rerun.

Figure 8 shows the net coverages achieved by the testers during the case study, as well as the goal set for the players and the highest net coverage resulting from OSP calculations. Furthermore, the number of submitted solutions per player is shown. Only player results that met the maximum number of sensors objective were considered for this evaluation.

It can be seen that all but one of the probands (player no. 7) were able to achieve the minimum net coverage. One proband's best solution (player no. 8) reached the same net coverage as DE, and player no. 9 even slightly surpassed DE with a net coverage of 75%. On average, the players submitted 60 solutions during the 20-min-long case study. Tester no. 5 alone tried 122 out of the 721 solutions, while tester no. 3 only submitted 31 sets of sensor positions for evaluation. The player who achieved the highest net coverage (player no. 9) only tested 37 solutions while playing.

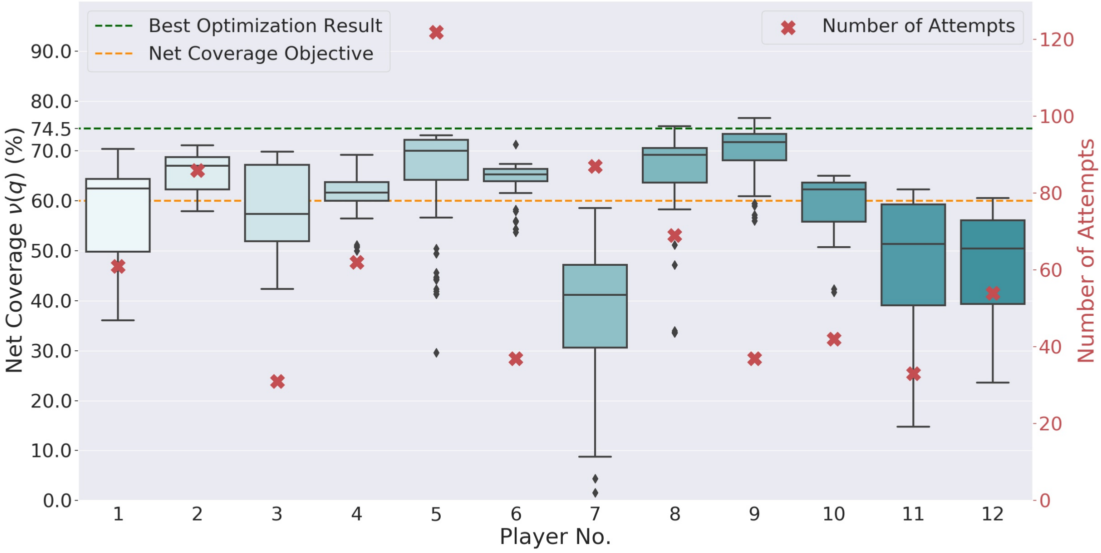


Figure 8. Net coverages achieved by the testers and number of attempts per tester.

Comparing the results gained by combining the OSP algorithm by Casillas et al. with DE and the results from the case study, it can be seen that human players are able to reach results similar to those computed using an optimizer. While 11 out of the 12 testers were able to achieve a net coverage above the minimum net coverage during the 20 min long case study, two were able to either reach or even surpass the net coverage obtained through DE.

Table 1 shows the percentage of solutions per number of placed sensors, taking into account all submitted solutions. Most of the solutions feature a number of placed sensors $n = 15$, with a dramatic decrease in terms of percentage towards $n = 14$ and $n = 16$. A further decrease towards the minimum number of used sensors $n_{min} = 2$ and the maximum $n_{max} = 19$ can be observed, as well.

Table 1. Percentage of solutions per number of placed sensors.

Number of placed sensors	<10	10	11	12	13	14	15	>15
Percentage of solutions	6.10%	1.8%	2.36%	6.93%	7.07%	11.93%	53.81%	9.99%

Taking a closer look on the individual player's solutions reveals more details. Five probands almost exclusively tried placing 15 sensors, while six players tried using fewer sensors, as well. Furthermore, all players except two also tried using more than 15 sensors at least once. Only one player did not place more than 14 sensors. The solutions for n_{max} and n_{min} were provided by a single player. So, it can be observed that, when a maximum number of placed sensors is defined, the players try to exhaust this goal.

Finally, Figure 9 shows the temporal course of achieved net coverages for every player. The solutions of the three players that achieved the highest net coverages (players no. 5, 8, and 9) are depicted in color, while for better overview, the other players solutions are shown in grey. It can be seen that some players took their time before submitting their first solution, while others immediately started their trial-and-error optimization. Furthermore, all players tried finding better solutions until the test phase was over. However, some players paused for several minutes before submitting a new set of sensor positions for evaluation. Some of the players obviously tried completely different sets of sensor positions during their optimization, resulting in greater changes of their achieved net coverages, while others seemingly tried to further optimize their current solutions, resulting in small changes of achieved net coverages.

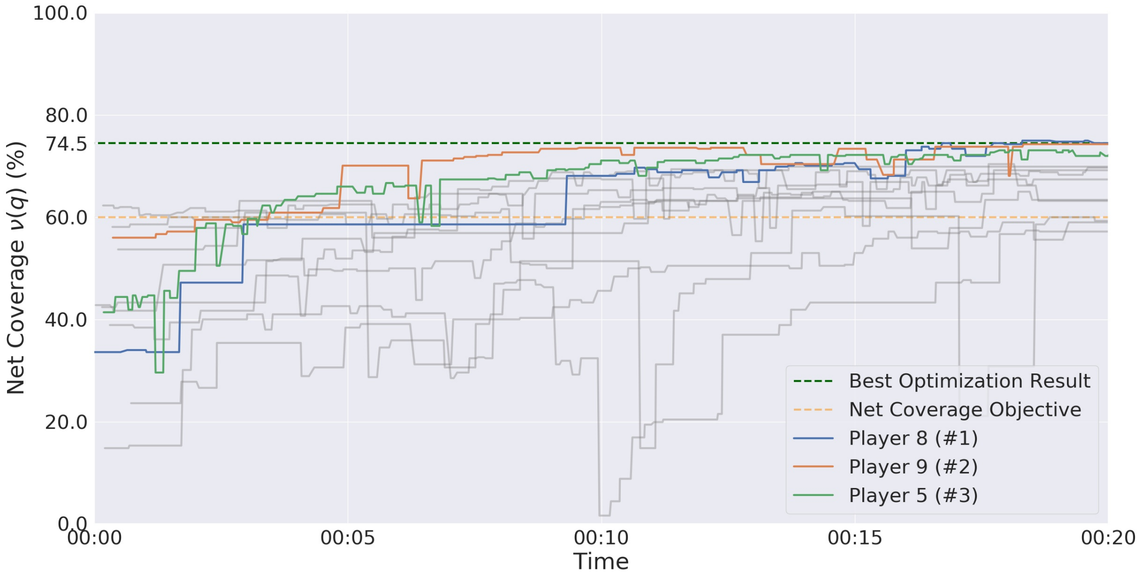


Figure 9. Temporal course of achieved net coverages per player.

3.2. Questionnaire Evaluation

Table 2 shows the results of the criteria evaluation using school grades from 1 to 5 (lower is better). The twelve testers rated SSP rather positively, mainly concerning the motivation to reach better results, the fun while playing, and the intuitivity of the operating elements.

Table 2. Questionnaire evaluation results (evaluation using school grades from 1 to 5 where lower is better).

Criterion	Minimum	Maximum	Mean	Median
Web Interface Clarity	1.00	3.00	1.92	2.00
Operating Elements Intuitivity	1.00	2.00	1.25	1.00
Gaming Fun	1.00	2.00	1.25	1.00
Motivation to reach better results	1.00	1.00	1.00	1.00

In addition, the testers communicated the following improvement suggestions:

1. The objective achievement could be marked using sound effects or pop-ups.
2. A Hall of Fame could show the highest net coverages achieved.
3. The time a user has spent playing the game could be shown in the GUI.
4. Hints could be implemented to help the players.
5. Costs resulting from the number of placed sensors could be displayed.
6. The scientific background of SSP could be described in the GUI.

4. Conclusions

In this paper, a novel Serious Game in the field of water distribution research was presented. For this game, a net coverage with regard to leak localization capabilities based on the optimal sensor placement algorithm by Casillas et al. was introduced. On the one hand, a minimum net coverage should be reached by the players, while on the other hand, a maximum number of placed sensors should not be exceeded.

The game itself was implemented combining Flask, Leaflet, and PostgreSQL based on calculations made using OOPNET. Afterwards, the Serious Game was tested by 12 players who tried to find ideal sensor positions in the C-Town model. During the 20-min-long case study, the testers were able to reach results similar to those obtained through DE, while also enjoying the game experience. Since the testers in the case study were mainly students, SSP was tested in a setting that represents its main purpose: education. The questionnaire evaluation showed that the overall rating was positive and that the game intention (i.e., to motivate the players to further improve their solutions) was met. However, some players recommended improvements of the GUI. Still it can be said that even the current simple GUI was sufficient to activate a player's intrinsic motivation to optimize his/her results.

Although SSP was only used for educational purposes so far, it could be helpful to demonstrate optimization tasks in WDS to a wider audience, as well.

In general, solving optimization problems with gaming approaches instead of classical optimization algorithms has the advantage of the users being able to apply their knowledge and expertise to solve the problem. The usage of "serious" planning tools, for instance, might lead to more practical results and could increase the commitment of the users compared to optimization-based approaches.

The presented Serious Game itself could also be extended to be used as a planning tool, since Leaflet is originally meant to be used for interactive 2D maps and could, therefore, be adapted to display real world WDS. However, further components would have to be implemented in SSP to create a suitable planning tool (e.g., cost estimations or the possibility to designate nodes, where the installation of measurement devices is infeasible).

Another aspect that should be investigated further is the impact of the level of a player's knowledge on the quality of his/her results. Due to the anonymized results obtained during our case study and the small number of testers, we were yet unable to quantify this influence.

Author Contributions: Conceptualization, G.A.-R. and D.F.-H.; Methodology, G.A.-R.; Software, G.A.-R.; Validation, G.A.-R. and D.F.-H.; Formal Analysis, G.A.-R.; Investigation, G.A.-R.; Resources, D.F.-H.; Data Curation, G.A.-R.; Writing—original draft preparation, G.A.-R. and D.F.-H.; Writing—review and editing, G.A.-R. and D.F.-H.; Visualization, G.A.-R.; Supervision, D.F.-H.; Project Administration, D.F.-H. All authors have read and agreed to the published version of the manuscript.

Funding: Open Access Funding by the Graz University of Technology.

Acknowledgments: We sincerely thank our peer reviewers for their very helpful comments which helped us improve and clarify this paper. Furthermore, we would like to express our gratitude to David Camhy whose expertise concerning IT infrastructure and programming was paramount for the successful implementation of this game.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DE	Differential Evolution
GUI	Graphical User Interface
OSP	Optimal Sensor Placement
SeGWADE	Serious Game for Water Distribution System Analysis, Design and Evaluation
SPuDU	Sensor Placement under Demand Uncertainties
SSP	Serious Sensor Placement
WDS	Water Distribution System

References

1. Savic, D.A.; Kapelan, Z.S.; Jonkergouw, P.M. Quo Vadis Water Distribution Model Calibration? *Urban Water J.* **2009**, *6*, 3–22, doi:10.1080/15730620802613380. [CrossRef]
2. Hart, W.E.; Murray, R. Review of Sensor Placement Strategies for Contamination Warning Systems in Drinking Water Distribution Systems. *J. Water Resour. Plan. Manag.* **2010**, *136*, 611–619, doi:10.1061/(ASCE)WR.1943-5452.0000081. [CrossRef]
3. Pudar, R.S.; Liggett, J.A. Leaks in Pipe Networks. *J. Hydraul. Eng.* **1992**, *118*, 1031–1046, doi:10.1061/(ASCE)0733-9429(1992)118:7(1031). [CrossRef]

4. Farley, B.; Mounce, S.R.; Boxall, J.B. Optimal Locations of Pressure Meters for Burst Detection. In *Water Distribution Systems Analysis 2008*; American Society of Civil Engineers: Reston, VA, USA, 2008; pp. 1–11. Available online: http://ascelibrary.org/doi/abs/10.1061/41024%28340%2963 (accessed on 22 December 2019).
5. Farley, B.; Mounce, S.R.; Boxall, J.B. Field Testing of an Optimal Sensor Placement Methodology for Event Detection in an Urban Water Distribution Network. *Urban Water J.* **2010**, *7*, 345–356. [CrossRef]
6. Farley, B.; Mounce, S.R.; Boxall, J.B. Development and Field Validation of a Burst Localization Methodology. *J. Water Resour. Plan. Manag.* **2013**, *139*, 604–613. [CrossRef]
7. Sarrate, R.; Nejjari, F.; Rosich, A. Sensor Placement for Fault Diagnosis Performance Maximization in Distribution Networks. In Proceedings of the 2012 20th Mediterranean Conference on Control Automation (MED), Barcelona, Spain, 3–6 July 2012; pp. 110–115. [CrossRef]
8. Sarrate Estruch, R.; Blesa Izquierdo, J.; Nejjari Akhi-Elarab, F.; Quevedo Casín, J. Sensor Placement for Leak Detection and Location in Water Distribution Networks. In Proceedings of the 7th IWA Specialist Conference on Efficient Use and Management of Water, Paris, France, 22–25 October 2013.
9. Sarrate, R.; Blesa, J.; Nejjari, F.; Quevedo, J. Sensor Placement for Leak Detection and Location in Water Distribution Networks. *Water Sci. Technol. Water Supply* **2014**, *14*, 795–803. [CrossRef]
10. Sarrate, R.; Blesa, J.; Nejjari, F. Clustering Techniques Applied to Sensor Placement for Leak Detection and Location in Water Distribution Networks. In Proceedings of the 22nd Mediterranean Conference on Control and Automation, Palermo, Italy, 16–19 June 2014; pp. 109–114. [CrossRef]
11. Pérez, R.; Puig, V.; Pascual, J.; Peralta, A.; Landeros, E.; Jordanas, L. Pressure Sensor Distribution for Leak Detection in Barcelona Water Distribution Network. *Water Sci. Technol. Water Supply* **2009**, *9*, 715. [CrossRef]
12. Quevedo, J.; Cuguero-Escofet, M.A.; Pérez, R.; Nejjari, F.; Puig, V.; Mirats, J. Leakage Location in Water Distribution Networks Based on Correlation Measurement of Pressure Sensors. In Proceedings of the IWA Symposium on System Analysis and Integrated Assessment, San Sebastian, Spain, 20–22 June 2011; pp. 290–297.
13. 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, doi:10.3390/s131114984. [CrossRef]
14. Casillas, M.V.; Garza-Castañón, L.E.; Puig, V. Optimal Sensor Placement for Leak Location in Water Distribution Networks Using Evolutionary Algorithms. *Water* **2015**, *7*, 6496–6515. [CrossRef]
15. Pérez, R.; Cugueró, M.A.; Cugueró, J.; Sanz, G. Accuracy Assessment of Leak Localisation Method Depending on Available Measurements. *Procedia Eng.* **2014**, *70*, 1304–1313. [CrossRef]
16. Cugueró-Escofet, M.; Puig, V.; Quevedo, J.; Blesa, J. Optimal Pressure Sensor Placement for Leak Localisation Using a Relaxed Isolation Index: Application to the Barcelona Water Network. *IFAC-PapersOnLine* **2015**, *48*, 1108–1113, doi:10.1016/j.ifacol.2015.09.675. [CrossRef]
17. Steffelbauer, D.B.; Fuchs-Hanusch, D. Efficient Sensor Placement for Leak Localization Considering Uncertainties. *Water Resour. Manag.* **2016**, *30*, 5517–5533, doi:10.1007/s11269-016-1504-6. [CrossRef]
18. Soldevila, A.; Fernandez-Canti, R.M.; Blesa, J.; Tornil-Sin, S.; Puig, V. Leak Localization in Water Distribution Networks Using Model-Based Bayesian Reasoning. In Proceedings of the 2016 European Control Conference (ECC), Aalborg, Denmark, 29 June–1 July 2016; pp. 1758–1763. [CrossRef]
19. Javadiha, M.; Blesa, J.; Soldevila, A.; Puig, V. Leak Localization in Water Distribution Networks Using Deep Learning. In Proceedings of the 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France, 23–26 April 2019; pp. 1426–1431. [CrossRef]
20. Abt, C.C. *Serious Games*; University Press of America: Lanham, MD, USA, 1987.
21. Dörner, R.; Göbel, S.; Effelsberg, W.; Wiemeyer, J. *Serious Games—Foundations, Concepts and Practice*, 1st ed.; Springer International Publishing: Basel, Switzerland, 2016.
22. Kickmeier-Rust, M.D.; Mattheiss, E.; Albert, D. An Educational Guide to Planet Earth: Adaptation and Personalization in Immersive Educational Games. In Proceedings of the 2nd International Workshop on Story-Telling and Educational Games, Aachen, Germany, 21 August 2009; pp. 36–45.
23. Brown, S.J.; Lieberman, D.A.; Gemeny, B.A.; Fan, Y.C.; Wilson, D.M.; Pasta, D.J. Educational Video Game for Juvenile Diabetes: Results of a Controlled Trial. *Med. Inform.* **1997**, *22*, 77–89, doi:10.3109/14639239709089835. [CrossRef] [PubMed]

24. Wu, J.S.; Lee, J.J. Climate Change Games as Tools for Education and Engagement. *Nat. Clim. Chang.* **2015**, *5*, 413–418, doi:10.1038/nclimate2566. [CrossRef]
25. Rebolledo-Mendez, G.; Avramides, K. Societal Impact of a Serious Game on Raising Public Awareness: The Case of FloodSim. In Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games, Sandbox '09, New Orleans, LA, USA, 4–6 August 2009; pp. 15–22. [CrossRef]
26. Douven, W.; Mul, M.L.; Son, L.; Bakker, N.; Radosevich, G.; Hendriks, A. Games to Create Awareness and Design Policies for Transboundary Cooperation in River Basins: Lessons from the Shariva Game of the Mekong River Commission. *Water Resour. Manag.* **2014**, *28*, 1431–1447, doi:10.1007/s11269-014-0562-x. [CrossRef]
27. Van der Wal, M.M.; de Kraker, J.; Kroeze, C.; Kirschner, P.A.; Valkering, P. Can Computer Models Be Used for Social Learning? A Serious Game in Water Management. *Environ. Model. Softw.* **2016**, *75*, 119–132, doi:10.1016/j.envsoft.2015.10.008. [CrossRef]
28. Hill, H.; Hadarits, M.; Rieger, R.; Strickert, G.; Davies, E.G.; Strobbe, K.M. The Invitational Drought Tournament: What Is It and Why Is It a Useful Tool for Drought Preparedness and Adaptation? *Weather Clim. Extrem.* **2014**, *3*, 107–116, doi:10.1016/j.wace.2014.03.002. [CrossRef]
29. University of Exeter, Centre for Water Systems. SeGWADE. Available online: http://waterseriousgames.org (accessed on 15 October 2019).
30. Meniconi, S.; Brunone, B.; van Zyl, K.; Mazzetti, E. Aqualibrium Competition: Laboratory Data and EPAnet Simulations. *Procedia Eng.* **2017**, *186*, 522–529, doi:10.1016/j.proeng.2017.03.265. [CrossRef]
31. Schaake, J.C.; Lai, F.H. *Linear Programming and Dynamic Programming Application to Water Distribution Network Design*; Report 116; MIT Press: Cambridge, MA, USA, 1969.
32. Savic, D.; Morley, M.; Khoury, M. Serious Gaming for Water Systems Planning and Management. *Water* **2016**, *8*, 456, doi:10.3390/w8100456. [CrossRef]
33. Morley, M.S.; Khoury, M.; Savić, D.A. Serious Game Approach to Water Distribution System Design and Rehabilitation Problems. *Procedia Eng.* **2017**, *186*, 76–83. [CrossRef]
34. Laucelli, D.B.; Berardi, L.; Simone, A. A Teaching Experiment Using a Serious Game for WDNs Sizing. *EPiC Ser. Eng.* **2018**, *3*, 1104–1112.
35. Arbesser-Rastburg, G. Serious Sensor Placement. Available online: https://sww-ssp.tugraz.at/ (accessed on 4 December 2019).
36. Rossman, L.A. *EPANET 2: Users Manual*; Water Supply and Water Resources Division National Risk Management Research Laboratory—Environmental Protection Agency: Cincinnati, OH, USA, 2000. Available online: https://nepis.epa.gov/Adobe/PDF/P1007WWU.pdf (accessed on 22 December 2019).
37. Steffelbauer, D.B.; Fuchs-Hanusch, D. OOPNET: An Object-Oriented EPANET in Python. *Procedia Eng.* **2015**, *119*, 710–718, doi:10.1016/j.proeng.2015.08.924. [CrossRef]
38. Ronnacher, A. Flask. Available online: https://palletsprojects.com/p/flask/ (accessed on 15 October 2019).
39. Agafonkin, V. Leaflet. Available online: https://leafletjs.com/ (accessed on 15 October 2019).
40. The PostgreSQL Global Development Group. PostgreSQL. Available online: https://www.postgresql.org/ (accessed on 15 October 2019).
41. Poulakis, Z.; Valougeorgis, D.; Papadimitriou, C. Leakage Detection in Water Pipe Networks Using a Bayesian Probabilistic Framework. *Probabilistic Eng. Mech.* **2003**, *18*, 315–327, doi:10.1016/S0266-8920(03)00045-6. [CrossRef]
42. Ostfeld, A.; Salomons, E.; Ormsbee, L.; Uber, J.; Bros, C.; Kalungi, P.; Burd, R.; Zazula-Coetzee, B.; Belrain, T.; Kang, D.; et al. Battle of the Water Calibration Networks. *J. Water Resour. Plan. Manag.* **2011**, *138*, 523–532. doi:10.1061/(ASCE)WR.1943-5452.0000191. [CrossRef]
43. Blesa, J.; Nejjari, F.; Sarrate, R. Robust Sensor Placement for Leak Location: Analysis and Design. *J. Hydroinform.* **2015**, 136–148. [CrossRef]
44. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359, doi:10.1023/A:1008202821328. [CrossRef]

45. Montague, D. Leaflet EasyButton. Available online: https://github.com/CliffCloud/Leaflet.EasyButton (accessed on 15 October 2019).
46. Fonticons, Inc. Font Awesome. Available online: https://fontawesome.com/ (accessed on 15 October 2019).