BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 70(4), 2022, Article number: e141594
DOI: 10.24425/bpasts.2022.141594

CONTROL AND INFORMATICS

Optimization of the process of restoring the continuity of the WDS based on the matrix and genetic algorithm approach

Ariel ANTONOWICZ* and **Andrzej URBANIAK**

[1] Institute of Computing Science, Poznan University of Technology, ul. Piotrowo 2, 60-965 Poznań, Poland

Abstract. The article discusses an example of the use of graph search algorithms with the trace of water analysis and aggregation of failures in the occurrence of a large number of failures in the water supply system (WSS). In the event of a disaster, based on the water distribution system (WDS) network model, information about detected failures, the condition and location of valves, the number of repair teams, criticality analysis, the coefficient of prioritization of individual network elements, and selected objective function, the algorithm proposes the order of repairing the failures. The approach proposed by the authors of the article assumes the selection of the following objective functions: minimizing the time of lack of access to drinking water (with or without prioritization) and minimizing failure repair time (with or without failure aggregation). The algorithm was tested on three different water networks (small, medium, and large numbers of nodes) and three different scenarios (different numbers of failures and valves in the water network) for each selected water network. The results were compared to a valve designation approach for closure using an adjacency matrix and a strategic valve management model (SVMM).

Key words: WNTR; aggregation of failures; water distribution system; EPANET solver; graph searching algorithm; genetic algorithm; optimization; post-disaster events.

1. INTRODUCTION

Water is essential in every aspect of human life. Quality and accessibility determine the chance of a dignified and healthy life. Difficult access to water has a direct impact on education and human development. It also affects the economic development and well-being of entire societies and states. Knowing how important water is, it seems obvious to maintain the continuity of the process of its distribution in an urban area. The supervisors of this process are water treatment stations whose task is to remove mechanical impurities, disinfect, correct the pH level, and perform other activities to improve the water quality [1, 2]. Their task is also to cooperate with water treatment (and sewage treatment) companies to detect and remove several types of failure. Currently, work is underway on decision support systems (DSS) for the management, planning, and improvement of the efficiency of the water distribution process (WDP), including the planning of the expansion, maintenance, and renovation of pipelines [3–5]. Another important aspect that must be considered to ensure the correct operation of the WDS is leak detection. Despite many methods developed to detect leaks [6, 7], their reliability is insufficient. The reason for such a low ratio is the lack of real data (which are difficult to obtain). Currently, to obtain information about the capabilities of a given WDS, complex algorithms, machine learning, or genetic algorithms are used. The role of such an algorithm is to monitor, generate, infer, predict, and detect drastic changes in the system. Monitoring means collecting information about the condition of the water supply network – considering the measuring and executive elements. Generating is about the analysis of the failure occurrence; whereas inferring is about making a diagnosis based on the collected data. The goal of prediction is to forecast the state of the network based on current and historical data [8–10]. Using a brute-force algorithm and analyzing all possible cases of one or more failures that occur simultaneously with a different number of available valves using conventional software would require the creation of separate simulation models for each case. It should be remembered that each of these models has information such as current network topology, status, location of valves, and location and type of failure. Analyzing these cases would take an exceptionally long time despite the small size of the water supply network [11]. There are numerous solutions in the form of scripts, applications, or tools that facilitate the modeling of various conditions in the WDS. There are following examples of such solutions: water network tool for resilience (WNTR), EPANET solver, Teva-Spot, water protector [12, 13]. The probability of a catastrophic event that could damage a WDS varies depending on where it occurs. For example, based on publicly available information, the probability of an earthquake in the USA is 17%, in Poland 11% and in Japan as high as 66% [14, 15]. Based on this information and the knowledge of how valuable the WDS infrastructure is, it is worth considering developing the right tool. A tool which allows us to indicate the sequence of repairs of failure occurring post-disaster situation, so that the time of restoring the supply capacity is the shortest.

*e-mail: aantonowicz@cs.put.poznan.pl

Manuscript submitted 2022-01-11, revised 2022-03-24, initially accepted for publication 2022-05-10, published in August 2022.

© 2022 The Author(s). This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

In this paper, we propose a new algorithm, for sequencing repairs of failures in the water supply network. The article also proposes a method of selecting a valve to be closed based on the adjacency matrix method with water trace analysis. Using the water route facilitates the minimization of the set of valves necessary to isolate the damaged pipe. This article additionally aims to provide a free tool that can be an excellent development base for DSS. Using the WNTR (containing the EPANET solver) and the Python programming language, it is possible to develop and implement an application that, using a genetic algorithm, will indicate the optimal solution to the analyzed problems. This method assumes that the location of the failures is already known.

2. METHODOLOGY

EPANET is a public domain tool developed in the 1990s by the US Environmental Protection Agency Water Supply and Water Resources Division. The main advantage of the presented tool was the possibility to simulate the operation of the water supply network with different initial conditions (e.g. different water demand). EPANET quickly became an educational and research tool to better understand the movement and fate of drinking water constituents within network systems [16]. Moreover, this tool facilitates: hydraulic and water quality simulation, supplying a mixture of water from various sources to recipients, calculating the age of water throughout a system, tracking the spread of contamination, and specification of loss of chlorine residuals [16, 17]. The water network tool for resilience (WNTR) is a Python package designed to simulate and analyze the resilience of the water distribution network [11, 16, 17, 26]. Beyond the possibilities offered by EPANET, WNTR facilitates the modeling of untypical situations such as power outages, incidental contamination events, and pipe breaks, leaks modeling, earthquakes, etc. The WNTR library contains two simulator solvers. WNTR simulator, which allows only hydraulic simulation (but allows one to choose a calculation method: demand-driven or pressure-dependent demand), and EPANET simulator which allows hydraulic and quality simulation (based on the demand-driven method) [16, 18–20]. The Ray Python library is a simple and easy-to-use framework that provides a REST-API for the implementation of distributed applications. This tool enables us to run a parallel grid search to optimize an example objective function problem (e.g. one- or multi-dimensional Knapsack problem, which can be considered when the vehicles of repair teams are being loaded) [22].

The description of the methods used will be presented based on the small topology of the water supply network shown in Fig. 1. The sample network consists of eleven nodes (including one reservoir and one tank), twelve links (including one pump), and eleven valves. The nodes are marked with the letter N, the links with the letter P, and the valves with the letter V.

2.1. The segment-finding method – matrix approach

Information on the segments of the water supply network is extremely important in selecting the valves to be closed (to separate the conduit from the rest of the network). The method of determining the segments of the water supply network is based on the adjacency matrix.

The first step in the segmentation algorithm is to create the valve deficiency matrix. The matrix below is created based on the difference of two matrices: adjacency and valve location. The adjacency matrix is created based on the correlation of nodes and links. In the matrix, rows present the nodes, and the column represents the pipes [11, 23]. If a given pipe (e.g. P1) correlates with a node (e.g. N1), the value "1" is entered into the matrix with coordinates (N1, P1) [23]. Table 1. contains the matrix for the example topology presented in Fig. 1.

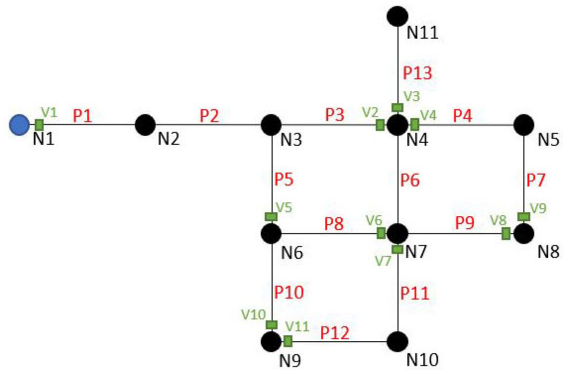

Fig. 1. Example of water network topology

Table 1
Adjacency matrix for topology presented in Fig. 1

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
N1	1												
N2	1	1											
N3		1	1		1								
N4			1	1		1							1
N5				1			1						
N6					1			1		1			
N7						1		1	1		1		
N8							1		1				
N9										1		1	
N10											1	1	
N11													1

The rows and the column in the valve location matrix represent the nodes and the pipes in the analyzed WDS analogously to the adjacency matrix. Table 2. contains the valve location matrix where "1" is in the matrix only if there is a valve on a pipe next to the analyzed node.

Using the adjacency matrix (Table 1) and valve location matrix (Table 2) the valve deficiency matrix presented in Table 3 is obtained as the difference of the first and second matrices, e.g.: Table 3 (N1, P1) = Table 2 (N1, P1) – Table 1 (N1, P1).

The second step of the algorithm is to create a list of all links included in the WDS, for which the segment determination pro-

Table 2
Valve location matrix for topology presented in Fig. 1

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
N1	1												
N2													
N3													
N4			1	1									1
N5													
N6					1								
N7								1			1		
N8							1		1				
N9										1		1	
N10													
N11													

Table 3
Valve deficiency matrix (based on Table 1 and Table 2)

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
N1													
N2	1	1											
N3		1	1		1								
N4						1							
N5				1			1						
N6								1		1			
N7						1			1				
N8													
N9													
N10											1	1	
N11													1

cedure will be performed. For the analyzed example, the above list is as follows: [P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12]. The next step of the algorithm is to analyze the valve deficiency matrix for each element in the list. The process of determining the WDS segments is as follows:

1. Create three empty lists: *pipe_list*, *node_list* and *list_of_valves*.
2. Add the analyzed pipe ID to *pipe_list*.
3. Search the column in Table 3 for that pipe to find a '1' at an end node without a valve. If a '1' is found in that column, add the row ID to the *node_list*. If there is no '1' in an analyzed column, the pipe has two valves, and the procedure is stopped.
4. Search row, where row ID is equal to node ID found in step 3. If a '1' is found in the row, add pipe ID to *pipe_list* and go back to Step 3. If there is no '1' in the row, all the incident pipes have a valve at that node and the procedure is stopped.
5. The set of identifiers for pipes and nodes determines a segment associated with the pipes in the *pipe_list*.
6. Based on the *pipe_list* and the data stored in Table 1, the *node_list* is extended to include all node IDs whose "1" is corresponding to the pipes.
7. Based on the data contained in Table 2 and the *node_list*, a *list_of_valves* is prepared at each node. When there is a valve on the *list_of_valves* which refers to a pipe from the *pipe_list*, other valves referring to the same node ID should be removed from this list.

For example, when analyzing the P9 pipe, the result lists of the algorithm are as follows:

- *pipe_list* = [P9, P6]
- *node_list* = [N4, N7, N8]
- *list_of_valves* = [V2, V3, V4, V6, V7, V8]

The result of the segment-finding algorithm is a set of network infrastructure elements (node, pipes, and valves) included in the identified segment.

2.2. The segment-finding method – graph searching approach

The algorithm for the valves to be closed was developed as a graph search application (GSA) using Python programming [24–27]. The network is represented in the form of a directed graph with weights, where the weight represents the presence (or absence) of valves on a given edge (pipe). The basic construction of the graph consists of analyzing the data contained in the hydraulic model of the WDS. Based on the information on the starting and ending nodes, as well as pipes and connections, the basic graph is determined. Then, based on the data on the valve location, the weights of the individual edges are determined. In the presented approach, the weight '0' means no valves on the edge, '1' means the valve on the beginning node of a given edge of the graph, '2' means the valve on the end node of a given edge, and '3' means two valves (on the beginning and end edge). The start and end nodes of the analyzed edge are determined by the network topology.

Figure 2 presented the modified topology from Fig. 1 in the form of a directed graph with weights. Based on the failure information (e.g. pipe directly connecting the nodes N7 and N8). The parallel algorithm (using the Python Ray library [21]) starts

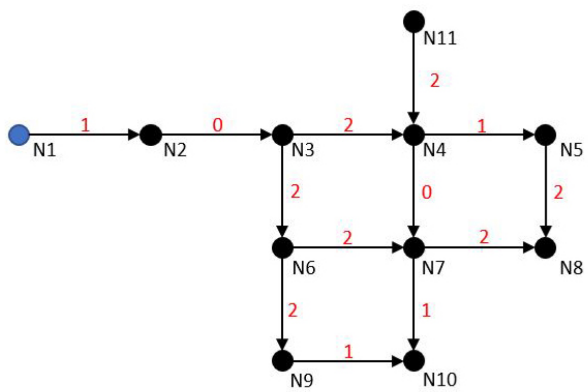

Fig. 2. Example of water network topology presented in Fig. 1 as graph with weights

to analyze all edges connected to nodes N7 and N8 (except the one that connects them directly). In the first iteration of the algorithm operation, the connecting edges are analyzed: [N5–N8, N4–N7, N6–N7, N7–N10]. After the edge analysis in the first iteration is completed, the algorithm returns the following information in the case of the edges:

- N5–N8 – there is a valve that can be closed (the valve closer to the node N8).
- N4–N7 – no valve to close – all edges connecting to node N4 should be analyzed: [N3–N4, N11–N4, N4–N5].
- N6–N7 – there is a valve that can be closed (the valve closer to node N7).
- N7–N10 – there is a valve that can be closed (the valve closer to node N7).

If no solution is found to cut off the broken pipe, the algorithm starts the second iteration by analyzing the newly added test cases. Similarly, to the first step, after completing the second iteration, the algorithm informs that:

- N3–N4 – there is a valve that can be closed (the valve closer to node N4).
- N11–N4 – there is a valve that can be closed (the valve closer to node N4).
- N4–N5 – there is a valve that can be closed (the valve closer to the node N4).

For the second step, the algorithm has not added any new case that needs to be analyzed. This means that it has found the valves that need to be closed to separate the broken pipe from the rest of the network. Both presented approaches achieved the same result. However, the first approach did it in a much shorter time (which will be discussed later).

2.3. Minimizing the closing of valves necessary to isolate the segment – water trace analysis

The segment finding method (matrix approach or graph search approach) is the key development base for the algorithm proposed by the authors of the article, who aim to determine the minimum set of valves necessary to isolate the indicated section. Based on the analysis of hydraulic simulations, it is apparent that it is not always necessary to close all the valves. Partial closing causes a sufficient reduction of water in the pipe to be repaired. This is most often caused by the location of a damaged conduit (e.g. end of a network), terrain, low pressure, or diameter of the conduit. The time required for the repair team to arrive, the location of the valve and the process of closing it itself is long, which is of immense importance in a crisis (many failures). To improve the efficiency of the process of determining the valves to be closed, a modified matrix graph method based on a hydraulic simulation of the water supply network model and flow analysis was proposed. Based on the determined set of valves (e.g. in the adjacency matrix approach), the algorithm generates a simulation scenario. The simulation scenario contains a set of hydraulic models which considers the combination in closing all the valves of the selected set. Each test case that does not result in flow in the analyzed pipe is saved. In the absence of a solution, the set of valves determined by the chosen approach is considered to be the minimum set of gate valves necessary to isolate the selected section. In the case of one or more solutions, the algorithm informs the user about the possibility of minimizing the number of valves and proposes the first solution from the list (the rest of the possibilities are also saved).

2.4. Modeling failures and determining the time of their repair

Failure is defined as damage to an object or system, which negatively affects its further functioning for a specified amount of time. In the example of the discussed tools, there are two ways of modeling failures. The first one is leak modeling, the second one is pipe break modeling.

Both cases are presented in Fig. 3. Leak modeling involves splitting the selected conduit and adding a node in it (which simulates outflow) while modeling a break involves removing the base pipe and creating two new separate conduits terminated with nodes simulating the flow of water from the broken pipe.

The repair time for particular types of failure (leak or break) is determined on the pipe diameter but also on the basis of the average time needed to prepare the substrate, the time of actual repair of the failure, the time needed to protect the substrate and the waiting time for repair (which may result in a need to pump out excess water). The individual times were compiled by experts and were published in the work [28].

Table 4 presents the average times (in minutes) of individual activities in relation to the diameter of the pipes. Based on the data stored in Table 4, it can be concluded that the proba-

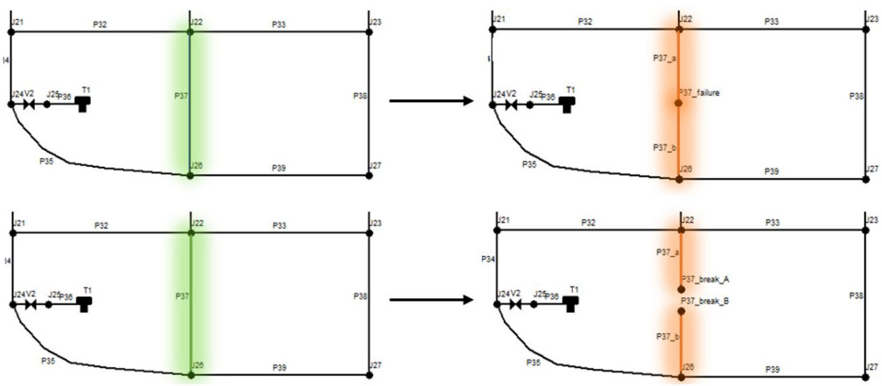

Fig. 3. Example of pipe failure modeling (leak and break)

Table 4
Average duration of exemplary repair activities [28]

ID	Activities	Pipe diameter [mm]		
		D1	D2	D3
		< 100	100–300	> 300
1	Open/Close valve	10–15	15–20	20–30
2	Failure location	20–25	55–60	50–60
3	Compacting asphalt pavement	70–120	120–160	160–240
4	Removal of paving slabs	30–35	45–50	60–65
5	Pipeline trench (mechanical)	80–100	100–140	150–240
6	Draining water from the excavation	30–60	80–120	120–180
7	Cleaning the cord	30–35	40–45	50 – 60
8	Installation of a repair band	25–30	40–50	60–90
9	Installation of sealing	40–45	50–55	60–90
10	Cutting out a section of the pipe	30–40	50–60	90–140
11	Installation of a new pipe	50–60	60–70	180–240
12	Water opening, venting, rinsing	20–25	30–35	50–60
13	Backfilling the trench	40–50	55–65	70–80
14	Finishing and cleaning activities	60–80	75–100	100–160

ble repair time of the pipes is as follows: (D1) 22 hours and 7 minutes, (D2) – 15 hours and 22 minutes, and (D3) – 10 hours and 13 minutes. It should be remembered that it is impossible to determine the actual total repair time due to the enormous number of external and internal disruptions that may occur.

Based on the calculated time, it can be concluded that with an increasing time needed for repair, the severity of the analyzed damage also increases. This means that failures of this type have a greater impact on the entire network infrastructure and the water distribution process than a failure whose expected repair time is shorter. Based on additional data from the literature [28], it is also possible to indicate a relationship between the time of repair of a failure and the probability of secondary contamination or new failures.

In the analyzed cases, the failure repair time is determined based on the above data (for each of the aforementioned activities, a value is drawn from the range determined by the average time of performing the activities). This means that, for example, in a random failure of a pipe with a diameter of D1, the valve opening time is a value randomly selected in the range of 10–15 minutes.

2.5. Prioritization and failures aggregation method

In the case of a crisis in which there are numerous failures, a simple classification (leak or break) defining their impact on the functioning of the collective water supply system is not sufficient. This is because only possible damage to the water distribution system has been identified, and not the functioning of the area in which it is located. It is important in such situations to consider the critical infrastructure (e.g. hospitals), which must have constant access to water or energy. This means that during the decision-making process and determining the order of failures, priority should be given to tasks that not only have a negative impact on the WDS but also prevent water supply to critical points.

The solution proposed by the authors enables the introduction of information about critical infrastructure. In the scenario configuration file, you should provide the identifiers of the nodes to which this type of object is attached. Then the algorithm, based on the set of failures and information about the identifiers of the segments in which they are located, starts an attempt to aggregate the failures.

The aim of this approach is to check whether among the reported failures there are those that have common gate valves that must be closed. If such a situation occurs, a simulation model of the water supply network is created, on which the simulation is run to check the effect of merging two segments into one. Such a simulation will facilitate an estimation of the time to repair two failures and the behavior of the network in the event of cutting off two segments at the same time. Thanks to this method, it is possible to minimize the total number of valves required to be closed to repair all failures. It is common that there are failures in the network that can be combined for repairs – this eliminates the situation in which the same valves must be opened/closed again.

2.6. Brute force and genetic algorithm approach

A brute-force algorithm was used to find the optimal solution to the scheduling task (to identify failures that must be fixed and identify repair teams). The result of the above method is a set of n failure lists (where n is the number of repair teams available). The order of the list reflects the order of the failures that the team should deal with. An example of the solution is shown: {rt1 = [P11–P12], rt2 = [P13], rt3 = [P2, P4]}.

The above example shows the availability of three repair teams (rt1, rt2, rt3) and five failures (P2, P4, P11, P12, P13). The algorithm, based on the location of the repair teams (the starting location is placed in the configuration file), determines the shortest paths to individual failures. The example solution indicates that the rt1 team will repair failures P11 and P12 simultaneously, which means that the failure aggregation algorithm created one failure P11–P12 from two separate failures P11 and P12. The rt2 team is responsible for repairing the P13 failure, while the rt3 team is responsible for repairing the P2 and P4. While checking all viable solutions in the case of a small water supply network and a small number of failures, it is not time-consuming, and the complexity grows with the increase in test cases.

To shorten the waiting time for the results, an approach using a genetic algorithm was proposed. Based on the information about failures and the determination of the probable time of their repair, an initial population is created, consisting of a set of individuals, containing n lists with a random solution (where n is the number of available repair teams). An example of an initial population is as follows (i.e. five individuals):

{
{rt1 = [P4, P2, P13], rt2 = [], rt3 = [P11–P12]},
{rt1 = [P4, P2], rt2 = [P13], rt3 = [P11–P12]},
{rt1 = [P4], rt2 = [P11–P12], rt3 = [P2, P13]},
{rt1 = [], rt2 = [], rt3 = [P2, P4, P11–P12, P13]},
{rt1 = [P13], rt2 = [P11–P12], rt3 = [P2, P4]}
}

Based on the initial population, the total time needed to restore supply to the water system is calculated for every individual. This result is then compared with the total probable recovery time score for all failures. The difference between the two values determines the usefulness of the individuals in the population. The smaller the difference, the better the solution. The new population is selected based on the tournament method. The algorithm is allowed by the crossing operator (considering the condition that a given failure may occur only once in one of the repair teams). The new population is treated as the current population, resulting in a redefinition of the membership function for new individuals in the new population. The algorithm ends its task when the total repair time is within the range declared in the configuration file or when the results of the new populations do not differ significantly from the previous ones.

3. CASE STUDY

The algorithm was tested on three different water networks (small, medium, and large number of nodes) and three different scenarios (different numbers of failures and valves in water network) for each selected water network. Table 5 contains basic information about the networks analyzed. The head loss factors for all networks are based on the Darcy-Weisbach formula and SI units. The selected water distribution systems include only the basic demands (there are no special collection points defined in the model i.e. large industrial plants). The structure of the network is a centralized network (all small villages are connected to the nearest biggest network) [17]. Based on the chosen water network models, Python interpreter, and necessary libraries (including the WNTR and Ray), a list of preliminary assumptions was prepared containing information necessary to conduct the simulation.

Table 5
Information about analyzed network models

	small	medium	large
Number of JUNCTIONS	114	1176	4868
Number of RESERVOIRS	1	2	2
Number of TANKS	1	1	4
Number of PIPES	171	1364	4038
Number of PUMPS	1	5	3
Total PIPE LENGTH [km]	12.1	97.6	185.5

The list contains information such as:

- the simulation time: 72 hours
- the simulation time step: 15 minutes
- the number of renovation teams: 3
- the number of critical infrastructure buildings: 2

To test the implemented solution, a scenario with a random number of failures and gate valves was created. In each scenario, the location of the failures and the valves may differ. Table 6 presents the number of failures analyzed (and the number of available valves) in relation to selected hydraulic models of water supply networks.

Table 6
Information about analyzed scenarios

	Number of failures	Number of valves
small – scenario 1	3	55
medium – scenario 1	10	200
large – scenario 1	10	425
small – scenario 2	6	55
medium – scenario 2	15	150
large – scenario 2	25	350
small – scenario 3	8	75
medium – scenario 3	15	300
large – scenario 3	20	325

4. RESULTS

Table 7 contains the results of the segment location algorithm that indicates the number of valves required to be closed. In the case of the brute force, matrix and graph searching algorithm, only the topology of the water network is considered (no information on flows and possible aggregation). All three methods indicated the correct valves, and the graph searching algorithm was the fastest. Only in the case of the large 1 scenario did the matrix algorithm turn out to be better. The shortest times are marked in green. The algorithm determining the water route and aggregation slightly affects the calculation time. However, it significantly influences the number of valves to be closed. As shown by the data in Table 7, the number of valves to be closed in individual scenarios has decreased by:

- small – scenario 1: 0 percent
- small – scenario 2: 16.6 percent
- small – scenario 3: 16.6 percent
- medium – scenario 1: 25.9 percent
- medium – scenario 2: 11.4 percent
- medium – scenario 3: 16.6 percent
- large – scenario 1: 18.2 percent
- large – scenario 2: 28.3 percent
- large – scenario 3: 14.5 percent

The orange color shows the minimum number of valves necessary to repair all failures.

In the case of the GSA-WT algorithm with aggregation, a task scheduling algorithm was applied using the genetic algorithm described in 2F of this article. The algorithm task was to determine the shortest route between failures (based on the number of repair teams and the location of the failures). The algorithms fulfilled their functionality, which means that they

Table 7
Segment-finding method results

Scenario			Brute force	Matrix approach (MA)	Graph searching approach (GSA)	MA with trace water	GSA with trace water	MA with trace water and aggregation	GSA with trace water and aggregation
SMALL	1	Execution time	00:04:42	00:01:51	00:00:58	00:02:15	00:01:22	00:03:14	00:02:44
		Valve to close	8	8	8	8	8	8	8
	2	Execution time	00:09:01	00:03:17	00:01:55	00:04:52	00:02:33	00:05:22	00:03:00
		Valve to close	18	18	18	16	16	15	15
	3	Execution time	00:13:22	00:03:59	00:01:59	00:05:12	00:02:36	00:05:44	00:02:51
		Valve to close	18	18	18	17	17	15	15
MEDIUM	1	Execution time	00:20:01	00:06:42	00:04:47	00:05:08	00:05:03	00:05:59	00:05:04
		Valve to close	27	27	27	26	26	20	20
	2	Execution time	00:24:23	00:06:00	00:04:41	00:05:02	00:05:12	00:05:42	00:05:23
		Valve to close	35	35	35	35	35	31	31
	3	Execution time	00:40:11	00:09:12	00:04:22	00:06:27	00:05:39	00:07:12	00:06:14
		Valve to close	42	42	42	41	41	35	35
LARGE	1	Execution time	01:44:11	00:17:33	00:17:39	00:18:49	00:19:02	00:20:05	00:19:54
		Valve to close	22	22	22	21	21	18	18
	2	Execution time	02:11:25	00:11:02	00:09:55	00:11:04	00:10:02	00:12:06	00:10:49
		Valve to close	67	67	67	60	60	48	48
	3	Execution time	01:33:55	00:08:23	00:07:34	00:08:01	00:09:01	00:09:18	00:09:18
		Valve to close	55	55	55	50	50	47	47

could be used in the case of scheduling tasks of repair teams in the event of a crisis. Figure 4 shows an example of a task scheduling algorithm for scenario 1 in a large network. In addition, Table 8 presents the results of the genetic algorithm for individual scenarios in comparison to brute force (complete review of all combinations of task distribution).

In most cases, the genetic algorithm found the optimal solution (in considerably shorter time). Only in three cases was

Table 8
Information about analyzed scenarios

Scenario	Brute force	Genetic algorithm	
	Time	Time	Is it optimal?
small – scenario 1	00:00:25	00:00:04	Yes
medium – scenario 1	00:16:14	00:03:44	Yes
large – scenario 1	00:18:27	00:03:11	Yes
small – scenario 2	00:00:51	00:01:02	Yes
medium – scenario 2	00:56:25	00:07:24	No
large – scenario 2	04:20:44	00:24:53	Yes
small – scenario 3	00:01:44	00:00:32	Yes
medium – scenario 3	00:55:21	00:06:54	No
large – scenario 3	02:13:11	00:16:39	No

Fig. 4. Example of graph searching algorithm with trace water and aggregation algorithm – task scheduling results

the solution not optimal but remarkably similar. In the case of medium scenario 2, the total difference in arrival at the failure was 14 minutes, for medium scenario 3: 8 minutes and large scenario 3: 22 minutes, which is a fully acceptable solution from the point of view of creating renovation tasks.

5. CONCLUSION

5.1. General thoughts – the pros and cons

The results of the presented scenarios show that the proposed algorithms are a good development base for decision support systems or early warning systems. Decision-makers and system developers should consider that the usefulness of this type of solution depends to a considerable extent on their coherence and integration with system handling failures [17]. The integration of the proposed approach with DSS or EWS directly determines the reliability of the results obtained [11,17]. Using the hydraulic model of the water distribution system and initial information on the algorithm return list of valves needed to be closed. Then, depending on the selected algorithm, we can determine the order of the closures and the sequence of failure repair.

All algorithms presented were developed using an open-source license. The great advantage of these types of approaches is the lack of licensing cost. Each version has been implemented in a modular manner, i.e. easily expandable. This means the software can evolve in real time – the developers can easily add new functionalities and modifications at any time.

The disadvantage of the proposed solution is the assumption that we have knowledge about the location failures. During a disaster (e.g. after an earthquake), we do not have holistic knowledge about failures, often only after some time do we learn about new ones. Therefore, it is worth considering a dynamic list of failures, which may change during the simulation. Another thing to be aware of is the assumption about valve closing time and conditions. Currently, the algorithm treats all valves as active. In fact, it often happens that, despite the presence of a valve in a given section, the status cannot be changed due to it is lack of efficiency.

5.2. Future research direction

The implemented algorithm is modular. It can be easily expanded with additional options. It is worth extending the current application with a task scheduling algorithm for specific renovation teams, assuming the capacity of repair vehicles, the skills of individual teams, warehouse inventory, etc. The application prepared in this way would be a great tool that can be used in a crisis.

The implemented algorithm is modular. It can be easily expanded with additional options. It is worth extending the current application with a task scheduling algorithm for specific renovation teams, assuming the capacity of repair vehicles, the skills of individual teams, warehouse inventory, etc. The application prepared in this way would be a great tool that can be used in a crisis.

In the EPANET environment and the INP file that contains the network topology data, there is no information about network valves, their location, and status. Therefore, it is worth developing standards for presenting information about the valves in a separate file or database, so that they can be loaded directly with the analyzed model. The information on the gate valves could also be updated following technical inspections.

In addition, it is worth considering a simple user interface, because currently everything is done via the console and commands that run individual scenarios.

REFERENCES

[1] L. Mays, *Urban Water Supply Handbook*, New York, Mc Graw Hill, 2002.

[2] World Health Organization, *Guidelines for Drinking-Water Quality*, Second edition, vol. 1, Recommendations, 2004.

[3] T. Hadzilacos *et al.*, "UtilNets: A Water Mains Rehabilitation Decision-Support System Computers", *Comput. Environ. Urban Syst.*, vol. 24 no. 3 pp. 215–232, 2000, doi: 10.1016/S0198-9715(99)00058-7.

[4] R.A. Francis, S.D. Guikema, and L. Henneman, "Baysesian Belief Networks for Predicting Drinking Water Distribution System Pipe Breaks", *Reliab. Eng. Syst. Saf.*, vol. 130, pp. 1–11, Oct. 2014, doi: 10.1016/j.ress.2014.04.024.

[5] S. Rimkevicius *et al.*, "Development of approach for reliability assessment of pipeline network system", *Appl. Energy*, vol. 94, pp. 22–33, Jun. 2012, doi: 10.1016/j.apenergy.2012.01.015.

[6] G. Moser, S. German, and P.F.C. Smith, "Performance comparison of reduced models for leak detection in water distribution networks", *Adv. Eng. Inf.*, vol. 29, no. 3, pp. 714–726, 2015, doi: 10.1016/j.aei.2015.07.003.

[7] R. Perez *et al.*, "Methodology for leakage isolation using pressure sensitivity analysis in water distribution networks", *Control Eng. Pract.*, vol. 19, no. 10, pp. 1157–1167, 2011, doi: 10.1016/j.conengprac.2011.06.004.

[8] R. Casey, P.F. Boulos, C.H. Orr, and C.M. Bros, "Valve Criticality Modeling", *Eighth Annual ASCE Water Distibution Systems Analysis Symposium*, USA, 2006, pp. 1–8, doi: 10.1061/40941(247)32.

[9] I. Piegdoń, B. Tchórzewska-Cieślak, and M. Eid, "Managing the risk of failure of the water supply network using the mass service system", *Eksploatacja i Niezawodność*, vol. 20 no. 2, pp. 284–291, 2018, doi: 10.17531/ein.2018.2.15.

[10] D. Kowalski, B. Kowalska, and M. Kwietniewski, "Monitoring of water distribution system effectiveness using fractal geometry", *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 63, no. 1, pp. 155–161, 2015, doi: 10.1515/bpasts-2015-0017.

[11] A. Anotnowicz, M. Nowak, and A. Urbaniak, "Task Scheduling Algorithm for Renovation Teams of Water Distribution Systems", in *Proc. 2020 21st International Carpathian Control Conference: Virtual Conference*, Slovak Republic, 2020, doi: 10.1109/ICCC49264.2020.9257273.

[12] G. Eason, B. Noble, and I. N. Sneddon, *Opflow*, AWWA, vol. 42, no. 5, 2016, pp. 11.

[13] A. Bałut and A. Urbaniak, "Management of Water Pipeline Netowrk Supported by Hydraulic Models and Information System", in *Proc. ICCC'2011*, Czech Republic, 2011, doi: 10.1109/CarpathianCC.2011.5945807.

[14] H. Fujiwara *et al.*, "Japan Seismic Hazard Information Stat", *4th International Conference on Earthquake Engineering Taipei*, Taiwan, 2006, p. 274.

[15] E.H. Field and 2014 WGCEP, "UCERF3: A new earthquake forecast for California's complex fault system", *U.S. Geological Survey 2015–3009*, p. 6, 2015, doi: 10.3133/fs20153009.

[16] L.A. Rossman, *Epanet 2 User Manual, Water Supply and Water Recourses Divisio National Risk*, Management Research Laboratory Cincinnati, OH 45268, EP/600/R-00/057, September 2000.

[17] A. Antonowicz, A. Bałut, A. Urbaniak, and P. Zakrzewski "Algorithm for Early Warning System for Contamination in Water Network", *2019 20th International Carpathian Control Conference (ICCC)*, 2019, pp. 1–5, doi: 10.1109/CarpathianCC.2019.8765966.

[18] K.A. Klise, R. Murray, and T. Haxton, "An overview of the Water Network Tool for Resilience (WNTR)", in *Proc. 1st Interational WDSA/CCWI Joint Conference*, 2018, p. 8.

[19] K.A. Klise, M. Bynum, D. Moriarty, and R. Murray, "A software framework for assessing the resilience of drinking water systems to disasters with an example earthquake case study", *Environ. Modell. Software*, vol. 95, p. 420–431, doi: 10.1016/j.envsoft.2017.06.022.

[20] K.A. Klise *et al.*, "Water Network Tool for Resilience (WNTR) User Manual" U.S. EPA Technical Report, EPA/600/R-17/265, p. 47.

[21] *Ray documentation*. [Online] Available: https://github.com/ray-project/ray.

[22] H. Jun, *Strategic valve location in a water distribution system*, Ph.D. Thesis, Faculty of the Virginia Polytechnic Institute and State University, 2005, Virginia.

[23] C. Sheppard, *Genetic Algorithms with Python*, Smashwords Edition S. l, 2016.

[24] A. Eiben and J. Smith, *Introduction to Evolutionary Computing (Natural Computing Series)*, Springer Berlin, Heidelberg, 1998.

[25] S. Raschka, *Python Machine Learning – Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytic*, Packt Publishing, 2015.

[26] R. Layton, *Learning Data Mining with Python – Harness the power of Python to analyze data and create insightful predictive models*, Packt Publishing, 2015

[27] L. Wrona and B. Jaworski, "Application of Genetic Algorithms in Graph Searching Problem", in *Proc. 2nd International Conference on Information Technology, ICIT 2010*, Poland, 2010.

[28] J. Bajer and A. Przebinda, "Czynniki wpływające na czas usuwania awarii przewodów wodociągowych i ich uzbrojenia", *Gaz Woda Tech. Sanit*, vol. 11, pp. 20–22, 2005 (in Polish).