The faster the better: On the shortest paths role for near real-time decision making of water utilities

Carlo Giudicianni[a,*], Manuel Herrera[b], Armando Di Nardo[a,c], Gabriele Oliva[d], Antonio Scala[c]

[a] *Dipartimento di Ingegneria, Università degli Studi della Campania 'L. Vanvitelli', via Roma 29, Aversa 81031, Italy*
[b] *Institute for Manufacturing – Dept. of Engineering, University of Cambridge, 17 Charles Babbage Rd., CB3 0FS Cambridge, United Kingdom*
[c] *Institute for Complex Systems – Italian National Research Council, via dei Taurini 19, Roma 00185, Italy*
[d] *Dipartimento di Ingegneria, Università Campus Bio-Medico di Roma, via Álvaro del Portillo 21, Roma 00128, Italy*

Abstract

Near real-time monitoring and control of critical infrastructure is essential for the operation and management of cities in a world that is, today, more complex and interconnected than ever. Such an infrastructure can be represented as complex networks an some of their related indices and statistics, many of them based on the shortest paths, play a pivotal role in the decision making for public services such as internet, energy or water. Particularly, the literature has shown that shortest paths are key for resilience and criticality assessment in a water distribution systems (WDS). This paper proposes a procedure to speed-up the computation of shortest paths in a WDS, as it can straightforwardly benefit any critical infrastructure. The proposal is based on a reduced dimension of a complex network representing any critical infrastructure. Despite the consequent decrease in the number of all possible paths in the network, the main advantage and novelty of this proposal is to continue finding the exact solution for the shortest paths. Experimental results show that the procedure brings a computational-time reduction consistently over 50% and up to 90% in some cases. In addition, the paper reveals how the use of shortest paths benefits WDS operation and management, as well as playing a key role in near real-time contamination detection and leakage control.

Keywords: Utility networks, Critical infrastructures, Water distribution systems, Network visualisation, Complex networks, Management science

*Corresponding author: Tel.: +39-081-000-0000;
Email address: carlo.giudicianni@unicampania.it (Carlo Giudicianni)

1. Introduction

Modern society is strongly dependent on infrastructure systems (i.e. transportation, power grids, telecommunications, water systems), which support cities growth and economic prosperity. These infrastructures continually face natural and man-made threats that cause economic and social disruption, leading their operators to continuously work on improving safety and security and, eventually, on speeding up mitigation actions. Today, reliability and performance assessment, continuous operation, monitoring and protection of critical infrastructures are national priorities for countries worldwide [1]. They represent an interdisciplinary challenge encompassing environmental, water, electricity and urban planning issues. Furthermore, as cities increase in their size, these infrastructures are getting larger and tangled, showing a complex behaviour due to the high degree of inter-dependency among them. As a consequence, the management of such infrastructures is becoming an arduous task to address, as this involves the development of new, agile tools and methodologies to support their decision making process. Water distribution systems (WDS) are among the most important critical infrastructures in a city. They guarantee the supply of drinking and industrial water to metropolitan areas and, therefore, their operation and management are of crucial importance to ensure social welfare, and resilience to any disruption that may place at risk the health of a city inhabitants. WDSs face two major vulnerabilities:

- *Contamination*: WDSs are vulnerable to malicious and intentional attacks since they are made up of thousands of exposed elements. In general, water can be easily polluted by chemical or biological contaminants, which spread all over the system by flowing and potentially have a dramatic impact on the population health [2].

- *Leakages*: WDSs are constituted by aged buried pipelines which are easily eroded by the environment. In addition, the daily pressure variability strongly stresses water pipes. These factors lead to failure and burst of pipes, causing leakages and wasting water [3, 4].

The downside in the management of such infrastructures is that the underlying details of the physics involved in their functioning complicates the analysis to a relevant extent, making it difficult to achieve useful insights in reasonable time [5]. Complexity science has proven to be a particularly adequate tool for a timely and agile analysis and management for WDSs [6, 7](and, in a general context, for critical infrastructures [8]). A complex system approach is suitable especially in the case of limited information about the infrastructure [9, 10]. In particular, a complex network representation allows to abstract the model away from the high degree of physical details and to focus only on a number of key aspects, in a manageable way [11, 12].

An essential tool in complex network analysis is the computation of the shortest paths. Centrality based algorithms, measuring the relative importance of

the network nodes, and community detection procedures, finding topologically-related nodes, are instances of methods relying in the shortest paths. This paper proposes a strategy to efficiently compute shortest paths, named *multiscale shortest path* (MS-SP). This is based on a dimension-reduction process, starting from the common scenario of a network already divided into communities. Such a network is modified to obtain a novel, dual representation of it with reduced complexity (in terms of the number of nodes and edges). This representation is named *multiscale network* and is equivalent to the original network in terms of computing the shortest paths.

An antecedent of this paper can be found on the work of [13] which provides a valid approximation to the shortest path problem for social networks. In such a work, the authors propose a combined process for community detection and network reduction by collapsing communities into nodes of a new network. Although such an algorithm had a scope similar to the work presented in this paper, the main innovation herein is that the network reduction process is based on the so-called landmark nodes. That is, the algorithm identifies a subset of key nodes that lie at the boundary of the communities and transforms the community into hyper-links connecting such boundary nodes, rather than collapsing the communities in single nodes. As a result, the network collapses into a reduced-size graph were boundary nodes are interconnected by edges that are weighted in a suitable manner to guarantee that the minimum path between two nodes in the original network can be computed in terms of the minimum path between the boundary nodes that are closest to the source and destination, respectively.

A major advantage of adopting the proposed MS-SP, with respect to other network reduction procedures, is that it takes into account all the connectivity information. Hence, it is possible to compute the exact value of any shortest path when the collapsed network layout is in use. This is not possible for traditional network reduction methods which normally collapse clustered areas into hyper-nodes. Actually, by collapsing the clusters in single nodes, the internal distance between boundary nodes cannot be taken into account, and the value of shortest path between two points is always an approximation. Furthermore, the proposed process of size reduction allows to get a more faithful representation of the original network by keeping all the landmark elements. As it will be subsequently discussed, this feature is the starting point for the creation of a novel management tool for WDSs.

Appendix A and Appendix B provide formal proofs to validate the speed of the proposed calculation of the shortest routes. This validation is also tested on two utility networks confirming its high operational performance. In addition to the aforementioned computational advantages, the dimension-reduction process also leads to a novel, dual representation of a WDS (or another networked infrastructure) where it is even possible to obtain a visualisation of the shortest paths. On top of these outcomes, the paper also shows the benefits on the use of the shortest paths for a water utility near real-time decision making. In particular, the paper presents a strategy to simplify the water quality sensor placement problem (contamination) and a graphical tool to optimise an adaptive

dynamic reconfiguration of district metered areas to efficiently address leakage control procedures.

2. Theoretical framework for a faster shortest paths algorithm

The networked asset connectivity of a critical infrastructure can be represented as a graph. In graphs representing real world systems, or complex networks, the connections between the nodes are often not homogenous and it is necessary to associate weights to the graph edges to better represent such a graph. A weighted graph is defined by $G = \{V, E, W\}$, having a finite number n of nodes $v_i \in V$ with $i \in \{1, \ldots, n\}$ and edges $(v_i, v_j) \in E \subset V \times V$ from node v_i to node v_j. For each edge $(v_i, v_j) \in E$ we denote by $w_{ij} \in W$ the associated weight. A graph is said to be *undirected* if $(v_i, v_j) \in E$ whenever $(v_j, v_i) \in E$, and it is said to be *directed* otherwise. In the following we will consider undirected graphs. For undirected graphs, we assume the weights satisfy $w_{ij} = w_{ji}$ for all $(v_i, v_j) \in E$. Let the *weighted adjacency matrix* of a graph $G = \{V, E, W\}$ be the $n \times n$ matrix A with the same structure as G, i.e., such that $A_{ij} = w_{ij}$ if $(v_i, v_j) \in E$ and $A_{ij} = 0$, otherwise. In the case of undirected graphs, matrix A is symmetric. A *path* over a graph $G = \{V, E, W\}$, starting from a node $v_i \in V$ and ending in a node $v_j \in V$, is a subset of links in E that connect v_i and v_j; the *length* of the path is the sum of the weights associated to the links in the path. A *minimum path* that connects v_i and v_j is the path from v_i to v_j of minimum length. An undirected graph is *connected* if for each pair of nodes $v_i, v_j \in V$ there is a path over G that connects them.

2.1. Antecedents

A main part of the paper focuses on the novel development of an efficient algorithm to compute the shortest paths in a complex network. There are previous work in the literature sharing a similar objective. In this regard, it highlights the work of [14], encompassing an extensive survey of various heuristic shortest path (SP) algorithms developed in the last years. It is worth to mention the interesting strategy adopted for practitioners and applied researchers to exploit network's domain-specific information. This is the case of traffic systems researchers adopting the natural hierarchies of the roads to significantly speed up the SP computational time [15, 16]. Overall, there are two widely investigated strategies for approximate the SP computation in large-scale complex networks. One of them is the $landmark - based$ method. This requires to pre-compute the shortest paths between special nodes (landmark nodes) and all the other nodes in the network, saving these distances in a database. The shortest-path between two nodes is, then, approximated by combining those distances stored in the database [17, 18]. The other one is a $topology - based$ approach. This strategy lies in the structure of networks and their partition into discrete areas [19, 20]. In this regard, [21] propose an approximated landmark-based method for point-to-point distance estimation in large-scale networks, also adding the partitioning variant. The landmark set is selected for each network area and the

shortest paths consequently saved in a database. The authors also demonstrated that selecting the optimal set of landmark nodes is an *NP-hard* problem. The proposal herein can be seen as a combination of both landmark and topology based approaches.

2.2. Shortest path algorithm

A widely known and applied algorithm to compute the shortest path between two nodes is Dijkstra's shortest paths algorithm (D-SP) [22]. D-SP can be summarised as follows. Given a weighted graph $G = \{V, E, W\}$ with $|V| = n$ nodes, a start node v_s and a goal node v_g, the algorithm keeps track of three variables for each node:

- $\texttt{visited}(v_i)$ which is equal to one if the node has already been visited during the algorithm and is zero otherwise;
- $\texttt{distance}(v_i)$ which is the current estimate for the distance of node v_i from the start node v_s;
- $\texttt{parent}(v_i)$ which is the identifier of the node immediately before node v_i in the path connecting v_s and v_i.

The algorithm also keeps track of the node currently being examined, which is referred to as v_*.

During the initialisation phase, the algorithm sets $\texttt{visited}(v_s) = 1$ and $\texttt{visited}(v_i) = 0$, for all $v_i \in V \setminus \{v_s\}$. Moreover, it sets $\texttt{distance}(v_s) = 0$ and $\texttt{distance}(v_i) = \infty$, for all $v_i \in V \setminus \{v_s\}$. Finally, the algorithm selects $\texttt{parent}(v_i) = \emptyset$ for all $v_i \in V$ and sets $v_* = v_s$. Then, the main cycle of the algorithm is executed; such a main cycle is composed of the following conceptual steps:

Step 1 For all neighbours v_i of v_* such that $\texttt{visited}(v_i) = 0$ set the distance of node v_i from v_s as the minimum between the previous estimate and the sum of the distance of v_* from v_s and the weight of the link $w_{*\,i}$ connecting v_* and v_i, i.e.,

$$\texttt{distance}(v_i) = min\left\{\texttt{distance}(v_i), \texttt{distance}(v_*) + w_{*\,i}\right\};$$

moreover, if the distance is updated for node v_i the algorithm keeps track of the fact that the minimum path from v_s to v_i features the edge (v_*, v_i) by setting

$$\texttt{parent}(v_i) = v_*.$$

Step 2 Set $\texttt{visited}(v_*) = 1$

Step 3 If $\texttt{visited}(v_t) = 1$ then stop, the algorithm is terminated.

Step 4 Otherwise, select the node with minimum current distance among the not visited ones as the new current node, i.e.,

$$v_* = v_j, \quad \text{where } j = \underset{i\,|\texttt{visited}(v_i)=0}{\arg\min} \{\texttt{distance}(v_i)\}$$

and go back to Step 3.

Note that a straightforward application of the above algorithm yields a computational complexity $\mathcal{O}(|V|^2)$ where $|V|$ is the number of nodes in the graph; moreover, when the graph is particularly sparse, i.e., when $|E| \ll |V|(|V|-1)/2$, where $|E|$ is the number of edges, it is possible to reduce complexity by using an implementation that relies on data structures such as the so-called Fibonacci heaps [23].

2.3. Multiscale Shortest Path (MS-SP) algorithm

Given a graph $G = \{V, E, W\}$, the proposed approach to calculate the shortest path from a node v_s to a node v_t is based on a dimension reduction procedure. To this end, the network is decomposed into clusters and the nodes/edges in each cluster are collapsed in a way that guarantees that the shortest path computed over the resulting graph corresponds to the one representing the original graph. These clusters are formed by grouping elements with similar characteristics or with a higher connection density than that external to the community. Network community detection algorithms [24] can be used in case the initial clustering of the network is not available. This is the case of the Louvain algorithm [25] which has been adopted in this paper to deal with the preliminary part of the process. The choice of Louvain algorithm is due to its properties of computational efficiency and scalability that make it suitable even for large-size networks. Actually, Louvain uses an iterative process to improve the scalability of the overall community detection based on modularity optimisation [26]. It is known that it runs in time $\mathcal{O}(|E|)$, where $|E|$ is the number of the graph edges [27]. Let's consider that we apply the Louvain clustering algorithm to a graph, G, decomposing the set of nodes V into q disjoint sets $V_1, \ldots, V_q$, each of them corresponding to a cluster.

In the following, we denote by E_i the set of edges in the original edge set E that connect nodes in the same cluster, i.e.,

$$E_i = \{(v_a, v_b) \in E \,|\, v_a, v_b \in V_i\};$$

moreover, we define

$$\hat{E}_{ij} = \{(v_a, v_b) \in E \,|\, v_a \in V_i \text{ and } v_b \in V_j\}$$

and

$$E_{ij} = \hat{E}_{ij} \bigcup \hat{E}_{ji}.$$

Finally, we define the set of *boundary nodes* $V_i^b \subseteq V_i$ as the set of nodes in V_i that belong to at least one edge in E_{ij} for some $j \in \{1, \ldots, q\} \setminus \{i\}$, i.e.

$$V_i^b = \{v_a \in V_i \,|\, \exists (v_a, v_b) \in E,\ v_b \notin V_i\}.$$

In other words, E_{ij} is the set of edges that connect nodes in V_i and nodes in V_j, and it holds $E_{ij} = E_{ji}$. Specifically, by running the clustering procedure described above, the network is decomposed into q clusters. The dimension reduction strategy consists in the construction of a graph

$$\widetilde{G} = \{\widetilde{V}, \widetilde{E}, \widetilde{W}\},$$

where $\widetilde{V}$ includes the set of boundary nodes and the start and goal nodes, i.e.,

$$\widetilde{V} = \{v_s, v_t\} \bigcup_{i=1}^{q} V_i^b.$$

As for the edge set $\widetilde{E}$, we have that

$$\widetilde{E} = \widetilde{E}_{\texttt{in}} \bigcup \widetilde{E}_{\texttt{out}},$$

where $\widetilde{E}_{\texttt{out}}$ is the union of the edges connecting boundary nodes, i.e.,

$$\widetilde{E}_{\texttt{out}} = \bigcup_{i,j \in \{1,\dots,q\}} E_{ij}$$

and $\widetilde{E}_{\texttt{in}}$ is the union of sets $\widetilde{E}^i_{\texttt{in}}$ of edges that directly connect the boundary nodes in the i-th cluster. Note that, if the start or goal nodes are in the i-th cluster, then the start or goal nodes are considered as a boundary node.

With respect to the graph weights, we select $\widetilde{w}_{ab} = w_{ab}$ whenever $(v_a, v_b) \in \widetilde{E}_{\texttt{out}}$, while for each pair of boundary nodes v_a, v_b that belong to the same cluster i (including the start or goal node if they belong to cluster i), we compute the minimum path p^i_{ab} between v_a and v_b over the subgraph of G induced by considering just the nodes V_i in the i-th cluster and we set the weight as the length of the path p^i_{ab}, i.e.,

$$w_{ab} = \sum_{(v_h, v_k) \in p^i_{ab}} w_{hk}.$$

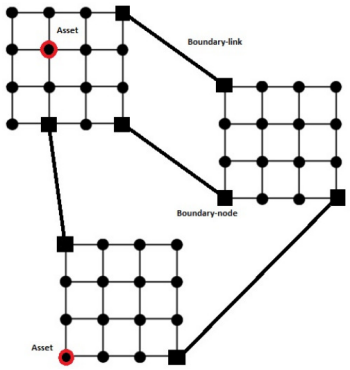
At this point, the algorithm finds the minimum path between nodes v_s and v_t by computing the minimum path between v_s and v_t over $\widetilde{G}$. Note that, by keeping track of the minimum paths involving boundary nodes in each cluster (treating v_s and v_t as boundary nodes), we are able to reconstruct the minimum path over G in terms of the minimum path over $\widetilde{G}$.

The algorithm is graphically explained by Figure 1 in which there are 3 groups: the upper-left cluster contains three boundary nodes (and the start node), the right cluster has three boundary nodes and the lower cluster has two boundary nodes (plus the target node). As a result of the decomposition, we obtain a network with $|\widetilde{V}| = 10$ nodes (i.e., the boundary nodes plus the start and goal) and $|\widetilde{E}| = 16$ edges; in particular, the four edges connecting nodes in different clusters are kept, while for each pair of boundary (or start/goal) nodes

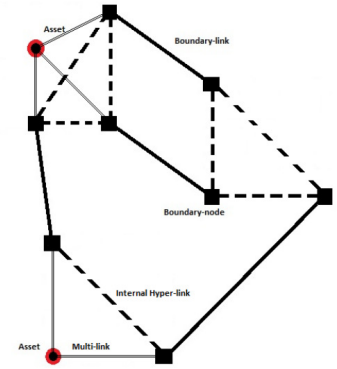
in each cluster a new link is added, whose weights correspond to the length of the minimum path, computed over the subgraph induced by the nodes in the cluster. The minimum path is computed over $\widetilde{G}$. Proofs of the correctness and time complexity of the proposed algorithm are reported in Appendix Appendix A and Appendix B, respectively.

Summarising, the main novelties of the proposed dimension reduction process are:

- the choice of an optimal number of clusters from a topological point of view (according to the relationship found in [28]);
- the idea of collapsing clusters in a subset of landmark nodes;
- the choice of the landmark nodes as the boundary nodes of each clusters;
- the idea of linking landmark nodes internally with link weighted with shortest path values and externally with boundary links defined by the clustering process.



(a) Original network showing key elements



(b) MS network showing key elements

Figure 1: Graphical explanation of the dimension reduction process for computing the SP algorithm

Algorithm 1: Multiscale shortest paths, MS-SP, procedure

Input: Original WDS network, G; with $|V| = n$ nodes. DMA division into q groups of nodes, $V = \{V_1, \dots V_q\}$. Boundary nodes set $V^b = \{V_1^b, \dots V_q^b\}$.

Output: Shortest paths between all the nodes of the original WDS network.

Data: DMA membership per each node of the original WDS network.

```
/* Shortest paths, SP, computed by Dijkstra's algorithm                    */
```

Let $\widetilde{G}$ an MS network

for $h \in \{1, \dots, n\}$ **do**

  $v_h^b \leftarrow \min SP(v_h, V^b(v_h) \mid v_h \in V(v_h))$

  `/*` v_h^b `boundary node in the DMA of` v_h`,` $V(v_h)$`, closer to` v_h `*/`

for $i, j \in \{1, \dots, n\}$, $i \neq j$ **do**

  Let v_i the initial node and v_j the sink node

  Check DMA membership: $v_i \in V(v_i)$ and $v_j \in V(v_j)$

  if $V(v_i) = V(v_j)$ **then**

    return $SP(v_i, v_j)$

  else

    if $v_i \wedge v_j \in V^b$ **then**

      return $SP(v_i, v_j) \equiv SP((v_i, v_j) \mid \widetilde{G})$

  else

    if $v_i \in V^b \wedge v_j \notin V^b$ **then**

      `/* Connect boundary node` v_i `to boundary node` v_j^b `*/`

      `/* Connect boundary node` v_j^b `to node` v_j `*/`

      return $SP((v_i, v_j^b) \mid \widetilde{G}) + SP(v_j^b, v_j)$

  else

    if $v_i \notin V^b \wedge v_j \in V^b$ **then**

      `/* Connect node` v_i `to boundary node` v_i^b `*/`

      `/* Connect boundary node` v_i^b `to boundary node` v_j `*/`

      return $SP(v_i, v_i^b) + SP((v_i^b, v_j) \mid \widetilde{G})$

  else

    if $v_i \notin V^b \wedge v_j \notin V^b$ **then**

      `/* Connect node` v_i `to boundary node` v_i^b `*/`

      `/* Connect boundary node` v_i^b `to boundary node` v_j^b `*/`

      `/* Connect boundary node` v_j^b `to node` v_j `*/`

      return $SP(v_i, v_i^b) + SP((v_i^b, v_j^b) \mid \widetilde{G}) + SP(v_j^b, v_j)$

3. Experimental validation of the MS-SP procedure

Urban utilities such as water, gas, or electric power networks can be modelled as quasi-planar graphs (e.g., edges forming vertices wherever two edges

cross) with spatially organised weighted edges $G = \{V, E, W\}$. In the case of water distribution systems the set V of n vertices/nodes encompasses junctions, water sources and demand points. The set E of m edges/links includes pipes, pump stations, and valves. Eventually, W is a function that assigns a weight to each edge quantifying the physical characteristics (diameter, length, roughness, material and age). A complex network can capture and -dynamically and distributedly- store all this information, making it possible to capture the inherent heterogeneity of a WDS. This is achieved by labelling the complex network elements in relation to their function in the system and weighting them by their importance, accessibility, and physical characteristics. The findings of this paper can be straightforwardly applied to a weighted graph adding the natural WDS heterogeneity to the shortest paths calculation.

In particular, WDSs are strongly constrained by their geographical embedding [28] in that connections between distant nodes are unlikely to be found, due to physical and economic constraints.

3.1. Study cases

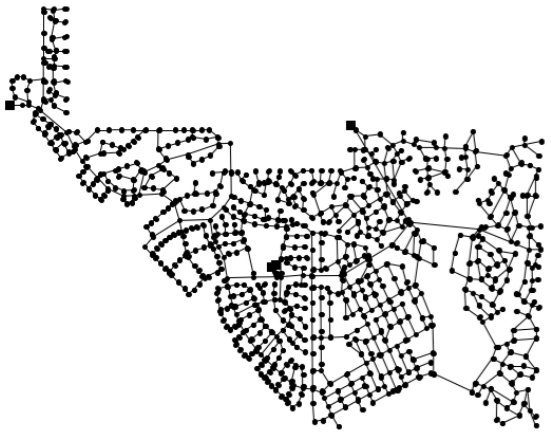
MS-SP is firstly tested on the real medium-size Colorado Springs (US) [29] water utility - which currently serves a population of about 370,000 inhabitants. Figure 2(a) shows its network layout. This encompasses 1,782 junctions and 4 reservoirs (n = 1,786 nodes), 1,985 pipes, 6 pumps and 4 valves (m = 1995 links). Figure 2(b) is a dual representation of Figure 2(a). Figure 2(b) clearly demonstrates the size reduction of the Colorado water network after its transformation into a MS network. This naturally highlights both highly interconnected network areas and bottleneck links, which are likely related to vulnerable parts of the WDS. Colorado Springs is one of the benchmark water networks, widely used by the urban hydraulics community. This has an added value for the sake of the reproductibility of this paper proposal.

The second case-study corresponds to the large-scale water utility which serves the Spanish city of Alcalá de Henares (Spain). It counts on a population of 201,000 inhabitants. The water distribution network model (see Figure 3(a)) encompasses 11,473 junctions, 3 reservoirs (n = 11,476 nodes), and 12,454 pipes, (m = 12,454 links). Figure 3(b) shows the corresponding MS network layout.

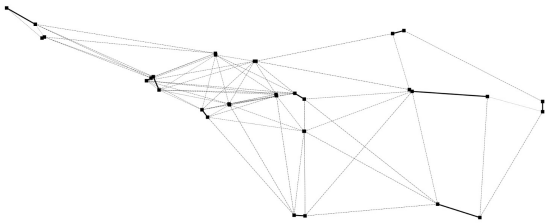
3.2. Results

This subsection introduces results corresponding to the topological analysis of the original water network and the MS network for both study cases. A description of relevant topological metrics, used to get these results, is reported in Appendix C.

Table 1 enumerates the main topological metrics computed for both case studies on the original and the MS network. The total number of links m_b for the MS network is equal to the sum of the boundary links m_{ex} and the internal hyper-links m_{in}. The size problem reduction is evident on nodes (from $n = 1,786$ to $n_b = 33$ for Colorado and from $n = 11,476$ to $n_b = 114$ for Alcalá) and also on links (from $m = 1992$ to $m_b = 83$ for Colorado and from $m = 12.454$



(a) Water network layout of Colorado Springs



(b) Multiscale water network of Colorado Springs

Figure 2: Multiscale dimension reduction for Colorado Springs water network

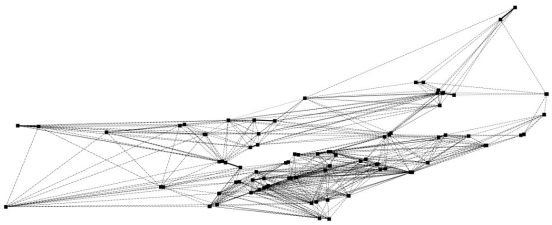
to $m_b = 596$ for Alcalá). The average node degree $\overline{K}$ strongly increases for the both the MS network (from $\overline{K} = 2.23$ to $\overline{K} = 5.03$ for Colorado and from $\overline{K} = 2.17$ to $\overline{K} = 10.46$ for Alcalá).

Table 1: Topological characteristics of the original water network and the MS network layout for Colorado Springs and Alcalá de Henares

Metric	**Colorado**	**Colorado-MS**	**Alcalá**	**Alcalá-MS**
n or n_b	1786	33	11,476	114
m or m_b	1995	83	12,454	596
$\overline{K}$	2.23	5.03	2.17	10.46
q	0.0012	0.1571	0.0002	0.0933
D	69	8	163	9
l	25.94	3.15	64.88	3.87
λ_2	0.00053	0.23512	0.00009	0.15884
$\Delta\lambda$	0.1293	0.1735	0.0957	0.0587

The dimension reduction working with the MS network makes the network density increases up to 2 orders of magnitude (from $q = 0.001$ to $q = 0.157$ for Colorado and from $q = 0.0002$ to $q = 0.0933$ for Alcalá). This augmented inter-connectivity is also reflected by the two spectral metrics measuring the robustness of network. The algebraic connectivity and the spectral gap also increase when moving from the original to the MS network, as it is shown in

(a) Water network layout of Alcalá



(b) Multiscale water network of Alcalá

Figure 3: Multiscale dimension reduction for Alcalá water network

Table 1. Still, the increasing link density does not represent a serious issue given the sparsity of the original network topology. The new the topological metrics for the MS network reflect a shift in its structure. The dual network representation can be seen now as a low interconnected small-world cluster (whose links are the internal hyper-links). In fact, after the size reduction due to the application of the MS-SP algorithm, each cluster of the MS network becomes into a fully connected layout, weakly linked to other clusters through out the boundary links. The typical small-world behaviour is also confirmed by the low value of communication metrics such as the diameter and the average path length, which scale approximately with $\log(n)$. This is a common feature of small-world network topologies as it is possible to see in Table 1.

Simulation results with respect to the computation of the shortest paths both for Colorado and Alcalá water utilities, are reported in Table 2 and Table 3. A suitable number of clusters C is taken in both cases to optimise the overall connectivity of the partitioned network, according to the relationship $C_{opt} \propto n^{0.28}$ reported in [28], where C_{opt} is the optimal number of clusters from a topological point of view. As a result, the number of clusters for Colorado is set to $C = 8$, while $C = 13$ for Alcalá's network. Up to 10 paths are generated by connecting random pairs of source and target to validate the proposed MS-SP

Table 2: Simulation results for the Colorado Springs water network

Pairs	**D-SP value** [-]	**MS-SP value** [-]	**D-SP time** [s]	**MS-SP time** [s]	**Red. time** [%]
1	13	13	0.0010	0.0001	90.0
2	21	21	0.0015	0.0005	66.6
3	29	29	0.0022	0.0006	72.6
4	33	33	0.0026	0.0007	72.9
5	38	38	0.0032	0.0004	87.4
6	41	41	0.0033	0.0006	81.7
7	52	52	0.0043	0.0007	83.6
8	56	56	0.0036	0.0005	85.8
9	60	60	0.0041	0.0006	85.2
10	66	66	0.0039	0.0004	89.6

algorithm. For each pair, the shortest path is computed by running the code 10 times and averaging the computational time.

Table 3: Simulation results for the Alcalá water network

Pairs	**D-SP value** [-]	**MS-SP value** [-]	**D-SP time** [s]	**MS-SP time** [s]	**Red. time** [%]
1	32	32	0.0007	0.0003	50.2
2	40	40	0.0019	0.0004	80.5
3	53	53	0.0032	0.0005	84.7
4	60	60	0.0041	0.0006	85.1
5	72	72	0.0027	0.0004	84.4
6	88	88	0.0098	0.0010	90.1
7	94	94	0.0102	0.0015	85.3
8	102	102	0.0129	0.0011	91.6
9	115	115	0.0094	0.0015	82.5
10	116	116	0.0097	0.0017	91.9

MS-SP algorithm provides the exact value of the shortest path between each pairs of randomly generated source and target nodes. This represents a clear advantage with respect to previous methodologies whom provide approximated results. Table 2 and Table 3 clearly state the D-SP and the MS-SP provide the same results (difference is equal to zero).

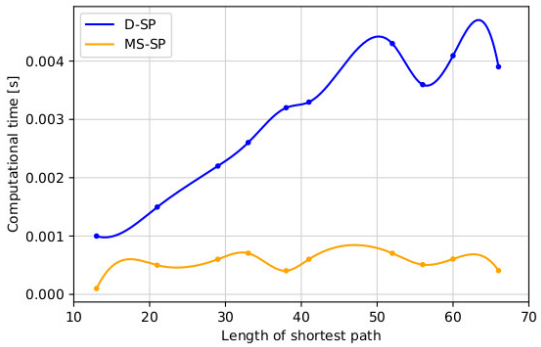


Figure 4: Computational time for D-SP and MS-SP algorithms, for Colorado water network

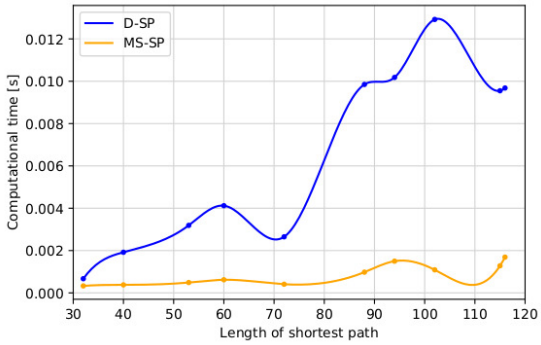


Figure 5: Computational time for D-SP and MS-SP algorithms, for Alcalá water network

The computational time for D-SP algorithm grows with the distance between source and target nodes, as it is expected. However, computational times for MS-SP show to be a plateau value of an order of magnitude smaller than that D-SP method. This is clearly shown in Figures 4 and 5 (with the results on Colorado and Alcalá utility networks). Table 2 shows the difference in percentage between the D-SP and the proposed MS-SP computational time for Colorado. The difference on time varies from 66% to 90%. Table 3 shows the difference in percentage between the D-SP and the proposed MS-SP computational time for Alcalá. This difference on time varies from 50% to 92%. Both differences on computational time stand as a conspicuous time reduction for computing the shortest path.

The MS-SP algorithm is implemented in Python 3.6. All the simulations run on a Linux Xubuntu 18.04 PC with 2.13 GHz Intel® Core^{TM} i3 CPU m330 64 GB of memory and 4.00 GB of RAM.

4. Shortest paths role for water systems management

Two novel applications of the shortest path algorithm for the monitoring and management of WDSs have been tested on Parete (Italy) water utility. This WDS currently supplies to a population around 11,000 inhabitants. Figure 6(a) shows its network layout. This encompasses 182 junctions and 2 reservoirs with fixed head of 110 m a.s.l. ($n = 184$ nodes), and 282 pipes ($m = 282$ links). The main trunks highlighted in red. The hydraulic analyses have been carried out by using the U.S. Environmental Protection Agency free software, EPANET [30], and considering a day of maximum consumption in the year, when the total demand at nodes ranges from 7.6 L/s at nighttime to 77.2 L/s both in the morning and midday peaks. The average water demand is 36.3 L/s.

4.1. Contamination detection

The most efficient action to enhance the security of a WDS against the effects of a contamination intrusion lies in installing a water quality sensor network

[31]. This represents a proactive, cost-effective and reliable strategy, allowing an assessment of the system water quality and an early detection of its potentially dangerous conditions. From practical and economic points of view, securing the entire network by placing sensors all over the system is not feasible, conditioned to the budget availability. Therefore, sensors should be placed in locations that maximise the capability of detecting contamination [32]. Water utilities have to face the issue of identifying the most suitable locations for sensor placement. In this regard, the optimal sensor placement problem is still an open challenge for researchers and practitioners, given its associated computational burden because of considering all possible contamination events along with the WDS complexity. It has been proven by [33] that the optimal sensor placement in a network represents a NP-hard combinatorial optimisation problem.

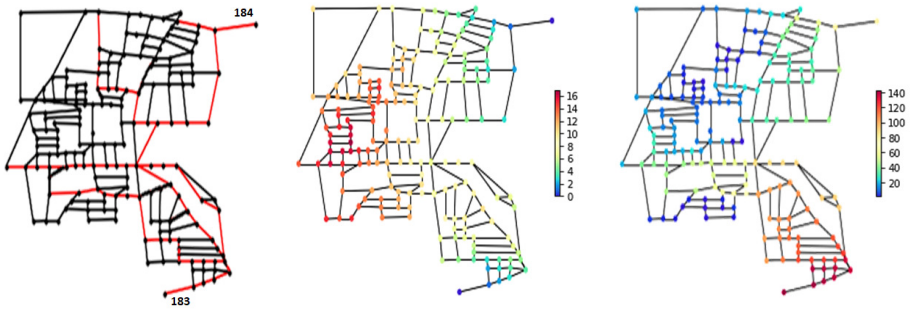
During the Battle of the Water Sensor Networks [34], several future research directions were identified. Among them, there highlight the following two:

- For a big-sized WDS the adopted event matrix represents only a small portion of the entire space of possible contaminant injection events. As a consequence, the generation of different event matrices will likely produce different solutions. The research challenge is to define procedures for which a rare subset from the entire set of contamination events can be computed (events with a small probability to occur, but with an extreme impact).

- Equal likelihood of threat and need for protection have been employed for all the elements of a WDS. There are needed novel tools for identifying areas of higher risk of threat and areas of greater need of protection.

The purpose of the current analysis is to provide a tool, based on the topological properties of the graph associated to WDSs, which allow to *a priori* define the most critical nodes (which can trigger the most extreme impacts) to consider for the design of an efficient water quality sensor system. Accordingly, it will be possible to reduce the computational burden, as well as, simplify the management by prioritising the protection to such areas, and consequently reducing the costs. In fact, the challenge of sensor placement is usually addressed by considering a set of contamination scenarios; each of them defined by the time when and the location where the contamination starts. However, in the literature, the creation of such scenarios consider all the WDS assets having a similar importance in terms of contamination spreading. The current shortest-paths based framework can be used as a decision-support system to ascertain the most critical scenarios to consider further. This will benefit the deployment of more effective preventive maintenance plans facing possible contamination events and more efficient strategies of system monitor and control.

The relationship between the shortest path of each node from the sources (reservoirs) and the node criticality level (as the downstream contaminated area extension if a contamination event would start from the node itself) has been investigated. The shortest path from the two reservoirs has been calculated for each node and the smallest result has been assigned as a feature to the referring node. Figure 6(b) shows the corresponding heatmap for the Parete WDS. In this



(a) Water network layout of Parete. Main trunks highlighted in red.

(b) Minimum shortest path of each node from the reservoirs.

(c) For each node, number of downstream nodes that would be contaminated if a contamination started from the node itself.

Figure 6: Water distribution network of Parete.

way, nodes are automatically split into two subsets (as many as the number of reservoirs), depending on whether the minimum shortest path is associated with one reservoir or with the other. The idea lies in the low number of reservoirs, or system inlets, which typically are in a WDS. Hence, it is possible to define a limited number of reference points, with a proper and well defined hydraulic function, for the calculation of shortest paths. Other system inlets are water tanks. Once water tanks are identified in a WDS, the framework proposed herein can be directly applied.

The water quality module of the EPANET software has been used to trace the flow originated from a node to the rest of the system and, consequently, the spreading of a potential contamination that moves along with the water flow. This has been done for the all nodes of the network, and the total number of reached (affected) nodes has been calculated. This number has been assigned as a feature to the referring node. Figure 6(c) shows the corresponding heatmap for the Parete WDS. By looking at the Figure 6(b) and Figure 6(c) an asymmetric correspondence can be detected; closer is the node to the reservoir, higher the number of affected downstream nodes, and vice-versa.

The minimum shortest path has been normalised with respect to the graph diameter $D = 20$ (see Appendix Appendix C for the definition), and the total number of affected nodes has been normalised with respect to the total number of nodes composing the network ($n = 184$ nodes). Figure 7 shows this relationship. Dots with red shades and dots with purple shades refer to nodes for which the minimum shortest path comes from reservoir ID-184 (red rhombus) and ID-183 (purple rhombus), respectively. Star shape stands for nodes belonging to the main trunks.

A closer look to Figure 7 provides a number of insights:

- The relationship is well fitted by a linear trend, confirming the strong correlation between the topology and the hydraulic behaviour of the WDS.

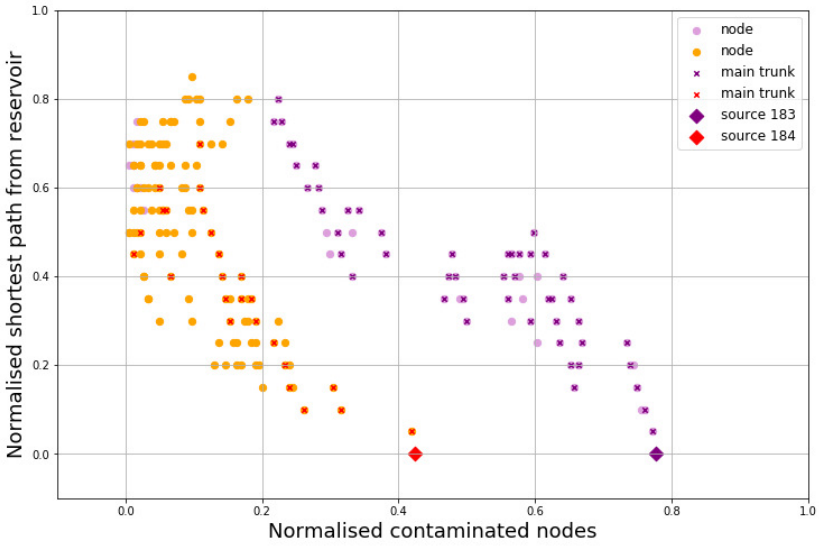


Figure 7: Relationship between the normalised shortest path from the reservoir and the normalised number of contaminated nodes

- Two clearly different trends can be spotted for the two subsets of nodes (referring reservoirs), allowing to define two areas of different protection level.
- The global most critical area is closer to reservoir ID-183 (contamination events starting from nodes far from it less than 20% of D will affect roughly the 80% of the nodes composing the WDS).
- For each subset, the most critical nodes are closer to the referring reservoir (smaller the corresponding shortest path is, greater the number of nodes it will affect). This allows to define nodes requiring higher level of protection. Furthermore, critical nodes can belong to the main trunks, but, they can also be regular demand nodes.

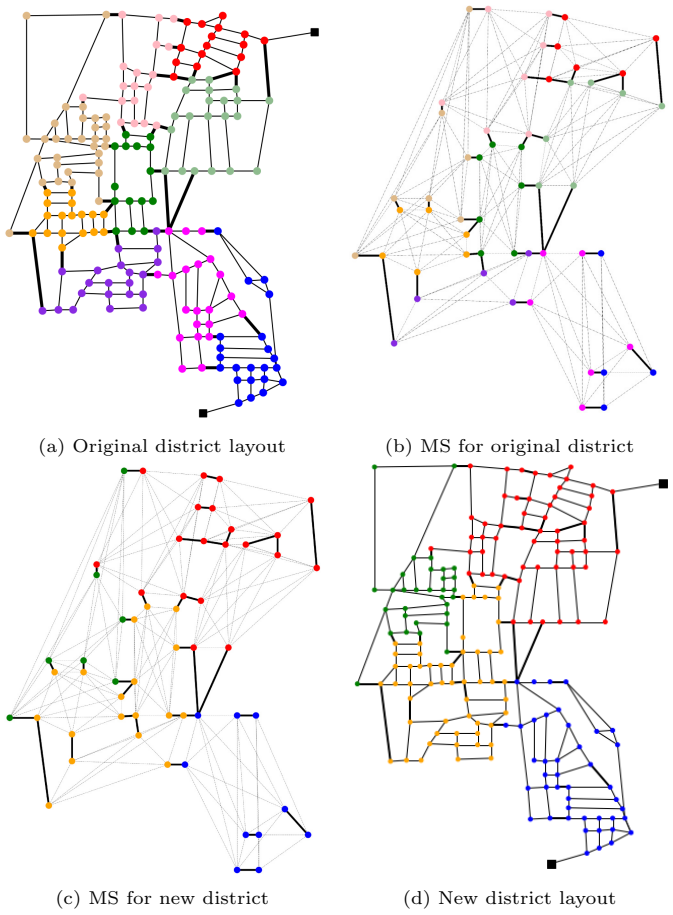
Another application of shortest path algorithms within a contamination context, come from the fact that it is often not possible to run hydraulic simulations on a WDS. This is due to lack of specific information about the water system. A topological approach for placement of water quality sensors represents an efficient strategy. This makes possible to disregard the hydraulic calibration of the models and to reduce the computational complexity of further procedures. For instance, the analysis can be restricted to just nodes closer to the reservoirs according to their minimum shortest path (whether they belong to the main trunks or not). The proposed approach defines the most critical spreader in a water network, by linking a topological-based information to hydraulic behaviour of the system. Despite the current approach only takes into account

the network layout for the shortest paths computation, it is clear that the procedure can be straightforwardly extended to consider geometric and hydraulic characteristics of the system assets (e.g. weighting the network by the corresponding pipe diameters, lengths, and roughness or demand and pressure for junctions), when additional information is available. In this way, it is possible to take into account the hydraulics of the system. This would enrich the findings on the relationship between shortest paths and the most critical nodes of a WDS, leading to a more explainable and plausible identification of them besides to gain model explainability and, consequently, trust from water utilities.

4.2. Leakage control

An appealing by-product of the proposed algorithm comes associated with the small-world property of the MS layout that preserves essential information about the original system while it leads to a significant network size reduction. This is key for addressing WDS partitioning into district metered areas (DMA) and it is one of the most useful management strategies for water utilities [35]. A DMA partition (also known as a WDS sectorisation) splits the system into smaller, monitored districts connected one to another by pipes equipped with gate-valves and/or flow-meters. This helps water companies to perform management and maintenance operations related to pressure and leakage control. Despite the multiple benefits of this management strategy, a permanent WDS sectorisation typically needs to close a large number of boundary pipes, leading to a general pressure drop at the WDS consumption points and, consequently, to an inefficient supply. Overall, the reason is that the more partitioned the network the more dissipated the supply energy. In addition, lower water pressures at the end user may deteriorate the hydraulic performance and reliability of the system. The definition of an optimal DMA configuration, that balances the aforementioned positive aspects towards a more resilient system, is still a challenge for water utilities and engineering practitioners.

The MS network implicitly takes into account the WDS structural knowledge, thanks to the shift to a low-interconnected small-world-clusters structure but inheriting key information of the original system. For instance, the aggregation phase this structural knowledge comes from a pairwise must-link (boundary links) and cannot link (internal links) constraints to be respected at each step of clustering by means of a semi-supervised approach [36]. This ensures the possibility to exploit the devices already installed for the original sectorisation, a rapid and cost-effective reconfiguration and management of the system, and finally a computational complexity reduction of the design procedure. As a consequence, the topological properties of the MS network ensures that the semi-supervised clustering algorithm, applied for the definition of new clusters, provides a solution in which the novel set of boundary links is a subset of the boundary links of the original cluster layout. On top of this, an initial step of network community detection (for WDS which are not already partitioned into DMAs) splits a network in such a way that each cluster comprises assets densely connected to each other; along with a low connectivity to items belonging to



(a) Original district layout (b) MS for original district

(c) MS for new district (d) New district layout

Figure 8: Dynamic district aggregation/disaggregation through the MS layout

other clusters or DMAs. Therefore, the new cluster layout will certainly cross the former boundary links and will not split the original DMAs.

The MS network, then, provides a dual solution for DMA partitioning that can be used dynamically, depending on the spatio-temporal variability of a WDS functioning conditions (ageing pipes, demographic increase, and water resource scarcity) and extraordinary conditions (unplanned demand peaks, insufficient pumping, short storage capacity, pipe breaks) that compromise the system performance. The MS dual solution can operate as an adaptive/dynamic DMA approach providing aggregation/disaggregation of districts according to specific ongoing conditions. The resulting MS-based DMAs are a top-down/bottom-up dynamic WDS partitioning that is able to work towards a smart and efficient water infrastructure management in response to unplanned and/or abnormal functioning conditions. For example, the initial smaller DMAs could be dynamically: a) aggregated (grouped) into bigger areas to improve network resilience

and pressure management, and to ensure water quality; and b) periodically disaggregated according to any specific objective such as leakage monitoring at night. The hydraulic performance and resilience to failure of a WDS are preserved for each network configuration; and the system energy and water quality related disadvantages of a closed, DMA topology are eliminated without losing the possibility to gain the benefits on control and operation associated with a WDS partitioned into DMAs.

Figure 8 represents a visual explanation of the fundamentals behind MS network for DMA reconfiguration and dynamic management. First, the current clustering/DMA layout of the WDS is detected (Figure 8(a)). Then, the corresponding MS network is built (Figure 8(b)). The boundary nodes of each cluster are highlighted by their corresponding district colour, the boundary links are in bold black line, and the internal links are in thin dashed grey line. Boundary links represent the connectivity between different clusters, while internal links stand for the internal connectivity of each cluster. The internal connectivity is approached according to the shortest path connecting each pair of boundary nodes belonging to the same cluster. Figure 8(c) shows how the original DMAs are subsequently aggregated in an MS network by applying a clustering algorithm. The new DMA configuration is finally defined at 8(d). Overall, Figure 8 shows the importance of the dual, multiscale representation introduced in this paper. In addition, the computation of shortest paths play a key role to approach how densely connected is each DMA and the general WDS layout. This eases the process of aggregation and disaggregation of DMAs for their dynamic management. The importance of the dynamic DMA management for water utilities resides in to balance system control (in issues such as management of leakage, contaminant, pressure) while having an optimal, efficient energy use [37].

5. Conclusions

This paper proposes a novel method to efficiently solve the shortest path problem in critical, networked infrastructures. The paper also shows how the process is specially useful for large-scale systems and near real-time decision making support. The algorithm is based on a community structure principle, which aids to collapse the original network into a set of interconnected, landmark nodes, through the also novel concept of a multiscale (MS) network. The MS network is a novel, dual representation, and visualisation method, of a networked infrastructure that eases to compute an efficient version of the shortest paths algorithm by a significant reduction of the network dimension.

The paper also provides a mathematical proof of the proposal and so a formal confirmation of the efficiency of the proposed MS shortest path (MS-SP) algorithm, in providing the the exact solution for the problem in a significantly lower computational time than using Dijkstra's algorithm. In addition, an experimental validation based on the study cases of two urban water utilities. The paradigm of decision making in water distribution systems has been used

throughout the manuscript showing how shortest paths, and therefore a their faster version, are key for the water supply operation and management.

The paper closes presenting two applications of the shortest paths for near real-time operation and management of water utilities. First, it is shown that the relationship between the minimum shortest path from reservoirs to water consumption nodes. This has been used as a basis to analyse the spreading of a contaminant throughout the system. The paper shows how it is possible to obtain a surrogate model of the hydraulic simulation analysis on contaminant spreading by such shortest paths connecting consumption nodes and reservoirs. As a consequence, this allows to define, beforehand, critical WDS areas, to speed-up the water quality monitoring and control, and to reduce the computational burden of the sensor placement problem. The second application shows how the MS network allows to simplify and make cost-effective the adaptive, dynamic reconfiguration of monitored district metered areas according to the variability of the system functioning conditions.

Future work will investigate the possibility to extend the proposed MS-SP algorithm to weighted and dynamically informed networks to include asset condition and network flow characteristics to the shortest paths solution. A weighted network provides a more accurate approach of the infrastructure it is representing. Furthermore, adding information on asset characteristics and conditions varying over time lead to a dynamic shortest paths computation. This makes possible, for instance, working with an adaptive definition of district metered areas for a WDS. A topic that is directly related to dynamic procedures for sensor placement and for data dimension reduction within a context of efficient management and smart monitoring and control. This framework has the potential to be used also in other public and critical infrastructure besides a WDS, where network flow/traffic dynamics already is a research avenue. For instance, finding faster procedures to compute shortest paths, depending on the time of the day and network status (link congestion awareness, e.g.) is of the higher interest in telecommunications systems and transport networks. In addition, it is foreseen a promising research activity based on shortest paths communicating multiple, interconnected infrastructures. Such an increased dimensionality of a complex network, within a system of systems framework, will become of main importance for the risk and resilience assessment of a critical infrastructure in a more than ever interconnected society and services.

6. Conflict of Interest Statement

Declarations of interest: none.

7. Acknowledgement

The research has been funded by Università degli Studi della Campania 'L. Vanvitelli' through the programme "VALERE: VAnviteLli pEr la RicErca.

References

[1] C. Alcaraz, S. Zeadally, Critical infrastructure protection: Requirements and challenges for the 21st century, International journal of critical infrastructure protection 8 (2015) 53–66.

[2] D. J. Kroll, Securing our water supply: protecting a vulnerable resource, PennWell Books, 2006.

[3] U. EPA, Control and mitigation of drinking water losses in distribution systems (2010).

[4] A. Di Nardo, M. Di Natale, C. Gisonni, M. Iervolino, A genetic algorithm for demand pattern and leakage estimation in a water distribution network, Journal of Water Supply: Research and Technology—AQUA 64 (1) (2015) 35–46.

[5] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, C. Faloutsos, Efficient sensor placement optimization for securing large water distribution networks, Journal of Water Resources Planning and Management 134 (6) (2008) 516–526.

[6] Q. Shuang, M. Zhang, Y. Yuan, Node vulnerability of water distribution networks under cascading failures, Reliability Engineering & System Safety 124 (2014) 132–141.

[7] A. Candelieri, D. Conti, F. Archetti, A graph based analysis of leak localization in urban water networks, Procedia Engineering 70 (2014) 228–237.

[8] I. Eusgeld, W. Kröger, G. Sansavini, M. Schläpfer, E. Zio, The role of network theory and object-oriented modeling within a framework for the vulnerability analysis of critical infrastructures, Reliability Engineering & System Safety 94 (5) (2009) 954–963.

[9] J. M. Torres, L. Duenas-Osorio, Q. Li, A. Yazdani, Exploring topological effects on water distribution system performance using graph theory and statistical models, Journal of Water Resources Planning and Management 143 (1) (2016) 04016068.

[10] C. Giudicianni, M. Herrera, A. Di Nardo, R. Greco, E. Creaco, A. Scala, Topological placement of quality sensors in water-distribution networks without the recourse to hydraulic modeling, Journal of Water Resources Planning and Management 146 (6) (2020) 04020030.

[11] Z. Wang, A. Scaglione, R. J. Thomas, Generating statistically correct random topologies for testing smart grid communication and control networks, IEEE transactions on Smart Grid 1 (1) (2010) 28–39.

[12] A. Di Nardo, M. Di Natale, C. Giudicianni, G. Santonastaso, D. Savic, Simplified approach to water distribution system management via identification of a primary network, Journal of Water Resources Planning and Management 144 (2) (2017) 04017089.

[13] M. Gong, G. Li, Z. Wang, L. Ma, D. Tian, An efficient shortest path approach for social networks based on community structure, CAAI Transactions on Intelligence Technology 1 (1) (2016) 114–123.

[14] L. Fu, D. Sun, L. Rilett, Heuristic shortest path algorithms for transportation applications: State of the art, Computers & Operations Research 33 (2006) 3324–3343.

[15] G. Jagadeesh, T. Srikanthan, K. Quek, Heuristic techniques for accelerating hierarchical routing on road networks, IEEE Transactions on Intelligent Transportation Systems 3 (4) (2002) 301–309.

[16] S. Jung, S. Pramanik, An efficient path computation model for hierarchically structured topographical road maps, IEEE Transactions on Knowledge and Data Engineering 14 (5) (2002) 1029–1046.

[17] L. Tang, M. Crovella, Virtual landmarks for the internet, In IMC 2003.

[18] J. Kleinberg, A. Slivkins, T. Wexler, Triangulation and embedding using small sets of beacons, in: Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on, IEEE, 2004, pp. 444–453.

[19] M. Henzinger, P. Klein, S. Rao, M. Rauch, S. Subramanian, Faster shortest-path algorithms for planar graph., Special Issue of Journal of Computer and System Science on selected papers of STOC 1994 55 (1) (1997) 3–23.

[20] N. Jing, Y. Huang, E. Rundensteiner, Hierarchical encoded path views for path query processing: an optimal model and its performance evaluation., IEEE Transactions on Knowledge and Data Engineering 10 (3) (1998) 409–431.

[21] M. Potamias, F. Bonchi, C. Castillo, A. Gionis, Fast shortest path distance estimation in large networks, CIKM 09 Proceedings of the 18th ACM conference on Information and knowledge management, Hong Kong, China, November 02-06, 2009 (2009) 867–876.

[22] E. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik 1 (1959) 269—271.

[23] M. L. Fredman, R. E. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms, Journal of the ACM (JACM) 34 (3) (1987) 596–615.

[24] S. Fortunato, D. Hric, Community detection in networks: A user guide, Physics Reports 659 (2016) 1–44.

[25] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment 2008 (10) (2008) P10008.

[26] M. E. Newman, Fast algorithm for detecting community structure in networks, Physical review E 69 (6) (2004) 066133.

[27] V. A. Traag, Faster unfolding of communities: Speeding up the louvain algorithm, Physical Review E 92 (3) (2015) 032801.

[28] C. Giudicianni, A. Di Nardo, M. Di Natale, R. Greco, G. F. Santonastaso, A. Scala, Topological taxonomy of water distribution networks, Water 10 (4) (2018) 444.

[29] I. Lippai, Colorado springs utilities case study: Water system calibration/optimization, Pipelines 2005: Optimizing Pipeline Design, Operations, and Maintenance in Today's Economy, American Society of Civil Engineers: Reston, VA, USA.

[30] L. Rossman, Epanet2 users manual national risk management research laboratory, us environmental protection agency, Cincinnati, Ohio2000.

[31] ASCE, Guidelines for designing an online contaminant monitoring system, Tech. rep., American Society of Civil Engineers (2004).

[32] S. A. McKenna, D. B. Hart, L. Yarrington, Impact of sensor detection limits on protecting water distribution systems from contamination events, Journal of water resources planning and management 132 (4) (2006) 305–309.

[33] X. Xu, Y. Lu, S. Huang, Y. Xiao, W. Wang, Incremental sensor placement optimization on water network, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2013, pp. 467–482.

[34] A. Ostfeld, J. G. Uber, E. Salomons, J. W. Berry, W. E. Hart, C. A. Phillips, J.-P. Watson, G. Dorini, P. Jonkergouw, Z. Kapelan, et al., The battle of the water sensor networks (bwsn): A design challenge for engineers and algorithms, Journal of Water Resources Planning and Management 134 (6) (2008) 556–568.

[35] C. Giudicianni, M. Herrera, A. di Nardo, K. Adeyeye, Automatic multiscale approach for water networks partitioning into dynamic district metered areas, Water Resources Management (2020) 1–14.

[36] M. Herrera, S. Canu, A. Karatzoglou, R. Pérez-García, J. Izquierdo, An approach to water supply clusters by semi-supervised learning, in: 5th International Congress on Environmental Modelling and Software, 2010, pp. 315–322.

[37] C. Giudicianni, M. Herrera, A. di Nardo, A. Carravetta, H. M. Ramos, K. Adeyeye, Zero-net energy management for the monitoring and control of dynamically-partitioned smart water systems, Journal of Cleaner Production 252 (2020) 119745.

Appendix A. Correctness of the MS-SP algorithm

The following theorem establishes that the path found via the MS-SP algorithm is, indeed, a minimum path.

Theorem 1. *The minimum path between nodes v_s and v_t over $\widetilde{G}$ is equivalent to the one connecting v_s and v_t over the original graph G.*

Proof 1. *Let's v_s and v_t belonging to the same cluster in G. Then, by construction, the minimum path found over $\widetilde{G}$ corresponds to the one over G. Let's assume now that v_s and v_t belong to different clusters with node sets V_s and V_t. By construction, since the clusters are connected only via edges joining boundary nodes belonging to different clusters, the minimum path joining v_s and v_t in G features a path from v_s to a node $v_{s'} \in V_s$, a path from $v_{s'}$ to a node $v_{t'} \in V_t$ and path from a node $v_{t'}$ to v_t (note that if $v_s = v_{s'}$ or $v_t = v_{t'}$ the path joining them is the empty set).*

At this point, we observe that the path connecting v_s to any $v_{s'} \in V_s$ and the path connecting v_t to any $v_{t'} \in V_t$ are, by construction, minimum paths; similarly, the path connecting any $v_{s'} \in V_s$ and $v_{t'} \in V_t$ with (recall that we assumed $s \neq t$) is a minimum path. Hence, by construction, the minimum path found over $\widetilde{G}$ corresponds to a minimum path $p_{st} = p_{ss'} \bigcup p_{s't'} \bigcup p_{t't}$ over the original graph, for some $v_{s'} \in V_s$ and $v_{t'} \in V_t$. The proof is complete.

Appendix B. Time complexity of the MS-SP algorithm

In the following, it is shown the computational cost of the proposed algorithm. It is important to point out that, the core idea of working on a size-reduced graph does not depend on the chosen clustering algorithm. As a consequence, a faster method can be adopted making the proposed MS-SP algorithm even more convenient from a computational point of view.

Proposition 1. *The computational complexity of the proposed approach, including the clustering procedure and the construction of the reduced graph $\widetilde{G}$, is equal to*

$$max\left\{\mathcal{O}(|E|), \mathcal{O}(n_b^2), \mathcal{O}\left(\sum_{i=1}^{q} |V_i|^2 |V_i^b|^2\right)\right\},$$

where V_i is the set of nodes in the i-th cluster and V_i^b is the set of boundary nodes in the i-th cluster and $n_b = \sum_{i=1}^{q} |V_i^b|$ is the cardinality of the set of all boundary nodes identified by applying Louvain algorithm.

Proof 2. *The computational complexity of the Louvain method is $\mathcal{O}(|E|)$. Moreover, once the clusters are formed, we need to scan all the edges to identify the set of boundary nodes, a procedure that requires $\mathcal{O}(|E|)$.*

At this point, MS-SP algorithm computes the shortest path over the subgraph induced by each cluster among each pair of boundary nodes in that cluster; each cluster has $|V_i|$ nodes, hence the computation of the shortest path from one node in the cluster to all other nodes in the cluster requires $O(|V_i|^2)$, since each cluster has $\mathcal{O}(|V_i^b|)$ distinct pairs of boundary nodes, we have that the computational complexity for each cluster is $\mathcal{O}(|V_i|^2|V_i^b|^2)$. Since the clusters are q we get $\mathcal{O}(\sum_{i=1}^{q}|V_i|^2|V_i^b|^2)$.

To conclude, the application of Dijsktra's algorithm on the reduced-size network has a complexity $\mathcal{O}(n_b^2)$; the proof follows since the two operations are done in series, hence the computational complexity is equal to the largest among the computational complexities of the above procedures.

Note that the computational complexity of the computation of the minimum path, after the graph $\widetilde{G}$ has been created is remarkably smaller, is $n_b \ll |V|$ for real world networks. Similarly, the complexity of the clustering procedure, although being theoretically upper bounded by $\mathcal{O}(|V|^2)$, is likely to be remarkably smaller. This specially occurs when the graph is sparse and $|E| \ll |V|(|V|-1)/2$, $|V|(|V|-1)/2$ is the number of edges in a complete graph.

As for the calculation of the minimum paths among the boundary nodes in the same cluster, we observe that there may be instances where complexity is above Dijsktra's algorithm[1]; however the likelihood of facing such instances is nearly zero in the case of WDSs and, in general, for graphs that have high sparsity and modularity. In fact, as discussed in the next remark, for those graphs the complexity of the construction of $\widetilde{G}$ is likely to be well below the one of Dijsktra's Algorithm. This fact is experimentally demonstrated in the next section.

Remark 1. *Note that the complexity of computing the minimum paths locally at every cluster has a complexity $\mathcal{O}\left(\sum_{i=1}^{q}|V_i|^2|V_i^b|^2\right)$. However, when the network has a clear modular structure, the number q of clusters is likely to be sublinear[2] in $|V|$ (e.g., $q = |V|^\gamma$ with $\gamma \in (0,1)$). Hence, on average, also the cardinality $|V_i|$ of the node set of the clusters is likely to be sublinear, i.e.,*

$$|V_i| \approx n/q = |V|^{1-\gamma}.$$

. Moreover, the cardinality of V_i^b is likely to satisfy $|V_i^b| \ll |V_i|$ and, in several practical cases, can be assumed to be constant for planar graphs and WDSs (see

[1] Consider for instance the case where the graph is full and is arbitrarily divided into 4 clusters with the same number of nodes; in this extreme case $V_i^b = V_i$ and thus the complexity of the proposed algorithm would be $\mathcal{O}(|V|^4)$.

[2] For instance, in [28] it is shown that for real WDSs the optimal number of clusters is $q \approx n^{0.3}$.

[28]), i.e., $|V_i^b| \approx \mathcal{O}(1)$*. Hence, in practical cases of interest for this paper, we have*

$$\mathcal{O}\left(\sum_{i=1}^{q} |V_i|^2 |V_i^b|^2\right) \approx \mathcal{O}\left(|V|^{1+\gamma}\right) < \mathcal{O}\left(|V|^2\right).$$

Remark 2. *Note that the construction of* $\widetilde{G}$ *can be slightly modified in order to be basis for the calculation of* all *shortest paths. In fact, it is sufficient to compute all shortest paths among every node in each cluster (i.e., requiring a computational complexity* $\mathcal{O}(\sum_{i=1}^{q} |V_i|^2 |V_i|^2) = \mathcal{O}(\sum_{i=1}^{q} |V_i|^4)$*) and storing information on the paths within each cluster. In this way, the graph* $\widetilde{G}$ *for calculating a path from any node* v_s *to any node* v_t *can be constructed by considering the links connecting boundary nodes and those connecting the start and goal nodes to the boundary nodes, an operation that requires at most* $\mathcal{O}(|V|)$ *in the worst case).*

Appendix C. Topological metrics

The topological comparison between the original layout and the dual network for the two study cases presented in Section 3 has been carried out in terms of:

- *Links Density* q which is the ratio between the total number m of network edges and the maximum number of edges $m^* = n(n-1)/2$ of a network with n nodes:

$$q = \frac{2m}{n(n-1)} \tag{C.1}$$

- *Average Node Degree* $\overline{K}$ is the average value of the node degree k_i (number of edges concurring in the node) over all nodes n:

$$\overline{K} = \frac{2m}{n} \tag{C.2}$$

- *Diameter* D is defined as the maximum shortest distance (the maximum geodesic length) d_{ij} between any pair of vertices i to node j (computed as the number of edges along the shortest path connecting them):

$$D = \max d_{ij} \tag{C.3}$$

- *Average Path Length* l is the average number of steps along the shortest paths for all possible pairs of nodes in the network:

$$l = \frac{2\sum d_{ij}}{n(n-1)} \tag{C.4}$$

- *Algebraic Connectivity* λ_2 corresponds to the second smallest eigenvalue of graph Laplacian matrix L

- *Spectral Gap* $\Delta\lambda$ is the difference between the first and second eigenvalue of the adjacency matrix A.