



HAL
open science

A Graph-Based Approach for Representing Wastewater Networks from GIS Data: Ensuring Connectivity and Consistency

Omar Et-Targuy, Ahlame Begdouri, Salem Benferhat, Carole Delenne,
Thanh-Nghi Do, Truong-Thanh Ma

► To cite this version:

Omar Et-Targuy, Ahlame Begdouri, Salem Benferhat, Carole Delenne, Thanh-Nghi Do, et al.. A Graph-Based Approach for Representing Wastewater Networks from GIS Data: Ensuring Connectivity and Consistency. ISDS 2023 - 1rst International Conference on Intelligent Systems and Data Science, Nov 2023, Can Tho (Vietnam), Vietnam. pp.243-257, 10.1007/978-981-99-7649-2_19. hal-04377851

HAL Id: hal-04377851

<https://hal.science/hal-04377851>

Submitted on 8 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Graph-Based Approach for Representing Wastewater Networks from GIS Data: Ensuring Connectivity and Consistency

Omar Et-Targuy^{1,2,3,5}, Ahlame Begdouri², Salem Benferhat¹, Carole Delenne^{3,5}, Thanh-Nghi Do⁴, and Truong-Thanh MA⁴

¹ CRIL, CNRS UMR 8188, University of Artois, France.

² LSIA, Université Sidi Mohamed Ben Abdellah (USMBA), Fez, Morocco.

³ HSM, University of Montpellier, CNRS, IRD, Montpellier, France

⁴ CICT, Can Tho University, Vietnam

⁵ Inria, Team Lemon, Montpellier, France

Abstract. Wastewater network management is a critical aspect of ensuring public health and environmental sustainability. Geographic Information Systems (GIS) are widely employed for managing and analyzing wastewater network data. However, storing data in separate databases in GIS poses challenges in accurately representing the connectivity of the wastewater network components, which are connected in reality. To address this issue, this paper proposes a novel graph-based approach for representing wastewater networks. In this approach, each component of the network, such as manholes, structures, pumps, etc., is represented as a node in the novel graph, while the pipes represent the connections between them. By adopting this graph-based representation, the true interconnected nature of the wastewater network can be effectively captured and visualized. This paper begins by discussing the limitations of the traditional separate databases approach in GIS for representing wastewater networks. Subsequently, the graph-based algorithm for transforming the GIS representation into a graph-based representation is presented. To validate the proposed approach, real-world wastewater network data is employed, demonstrating its ability to capture the network's connectivity and facilitate network analysis.

Keywords: Wastewater networks · Graph-based representation · Connectivity.

1 Introduction

Efficient wastewater management plays a vital role in ensuring public health and environmental sustainability. Geographic Information Systems (GIS) have become a popular tool for managing and analyzing geographic data associated with wastewater networks. However, the traditional approach of storing each component in separate databases can result in inconsistencies and inaccuracies, especially when it comes to representing the connections between these components.

In reality, wastewater networks are interconnected systems where various components work together to transport wastewater from its source to treatment plants. For example, manholes serve as access points to the pipes, facilitating maintenance and inspection activities. These connections between different components are crucial for the proper functioning of the network.

In GIS, each component of the wastewater network is typically stored in its own database, e.g. manholes in one database and pipes in another. While this approach allows for efficient management of individual components, it lacks the ability to capture and represent the connections between them. As a result, over time, as updates and modifications are made to the network, the databases become unsynchronized, leading to a loss of the true connectivity and relationships between the components.

To address the challenges associated with representing wastewater networks, a comprehensive and integrated approach is needed. A graph-based representation proves to be highly effective in capturing the interconnected nature of the network. By treating each component, such as manholes, pumps and structures as nodes, and representing the connections between them as edges which represents pipes in wastewater networks,

The Graph-Based Representation is a versatile method adept at illustrating data interrelationships in diverse domains. In [3], a pioneering graph-based spatial approach is introduced, employing Minimum Planar Graphs from Delaunay triangulations to assess habitat connectivity effectively. This innovation facilitates wildlife habitat network analysis and visualization, aiding in conservation decisions. In [4], the authors conduct a systematic survey of graph-based text representation, highlighting its superiority over the bag-of-words model in capturing structural and semantic information. The use of graph-based representation leads to enhanced performance in information retrieval, term weighting and ranking across various text applications. Additionally, in [5], a novel graph-based approach for multiview image representation is proposed. This method utilizes graph connections to describe pixel proximity in 3D space, achieving a compact and controllable representation that is well-suited for coding and reconstructing multiple views. In [1], a knowledge graph-based data representation is proposed for Industrial Internet of Things enabled manufacturing. [2] presents a global graph for modeling social interactions among pedestrians.

The paper is organized as follows. In Section 2, an overview of wastewater networks data is provided, and the limitations of representing them in a SIG using the traditional separate databases approach are discussed. Section 3 presents the graph-based algorithm, which transforms the GIS representation into a interconnected graph-based representation. In Section 4, experiments and results from applying the graph-based algorithm to real-world wastewater networks data are presented, with a focus on highlighting the algorithm's advantages and limitations. Finally, Section 5 concludes the paper by summarizing the benefits of the graph-based representation and its implications for wastewater network management.

2 Wastewater networks in GIS

Wastewater networks comprise a variety of interconnected components (see Figure 1) that collectively ensure the efficient and safe transportation of wastewater from its sources to treatment plants. Some of the essential components of wastewater networks include:

- **Pipes:** sewer pipes are the primary components of a wastewater network. They are used to transport wastewater from its source, such as residential, commercial or industrial buildings, to the treatment plants.
- **Manholes:** manholes are access points in the wastewater network that allow for maintenance and inspection activities. They are usually constructed at regular intervals along the sewer pipes. Manholes provide entry points for workers to access the sewer pipes and perform tasks like cleaning, repairs and inspections.
- **Pumps:** pumps are used to increase the pressure and flow of wastewater, especially in areas where gravity flow is not possible.
- **Fittings:** fittings are used to connect and join different sections of sewer pipes in wastewater networks.

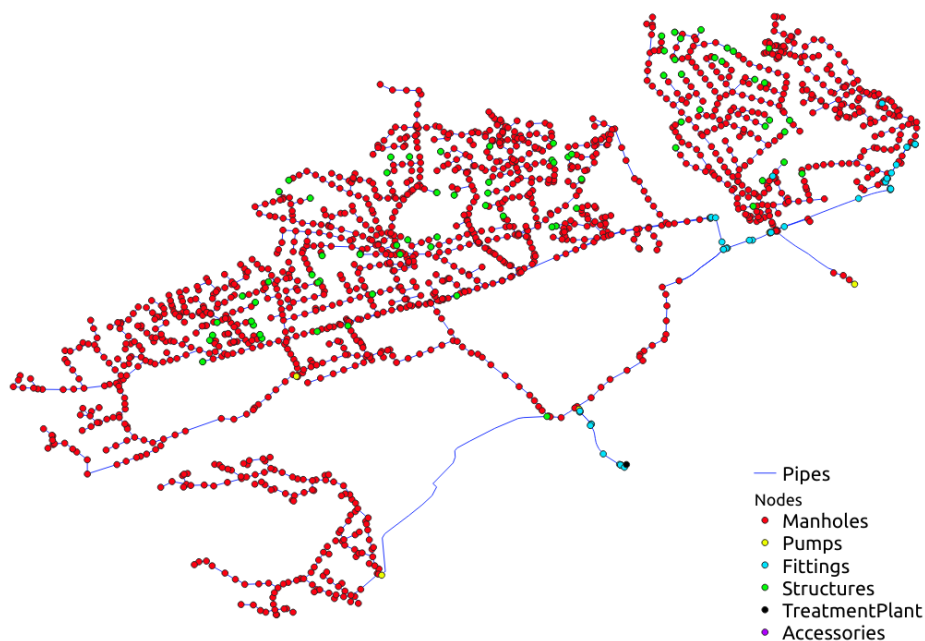


Fig. 1: Example of wastewater networks in GIS representation.

One characteristic of wastewater network data is the geographical information associated with each component, representing its position on the Earth. To

store, visualize and manage this spatial data, a Geographic Information System (GIS) is commonly employed. A GIS allows for the integration, analysis and interpretation of spatial data within the context of wastewater network management.

In the context of wastewater networks, two primary geometries are used to represent the network components in GIS (see Figure 1). The first geometry is the point geometry, which represents specific locations such as manholes, structures, pumps and other components. The second geometry is the line geometry, which represents the pipes that form the network infrastructure.

In addition to the geographical information, each component in the wastewater network is also characterized by attribute information. These attributes provide additional details and characteristics of the components, such as pipe diameter, material, flow capacity and other relevant data.

In a GIS system, the data related to each component of the wastewater network is typically stored in separate databases, as illustrated in Figure 1. Each component, such as pipes, manholes, pumps, fittings and structures, has its own database, allowing for easy management and analysis of the data.

However, in reality, these components are interconnected and have spatial relationships with each other; e.g, manholes serve as access points to pipes, and pumps are connected to specific pipes within the network. Therefore, it is crucial to establish and maintain the appropriate connections between these components.

As the amount of wastewater network data grows, it becomes increasingly important to update the nodes and pipes that represent these networks in the GIS. However, ensuring that these databases are properly synchronized with each other is challenging. Inconsistencies and inaccuracies in the representation of the wastewater network arise when the connections between the components are not accurately captured or updated. For example, Figure 2 illustrates some of the issues that can arise with this type of representation. In the figure, there are nodes that are not connected to any pipes, which is impossible in reality. Additionally, in some cases there are pipes that doesn't have a node at one of its extremities or both. There are many other potential problems that arise when the nodes and pipes of a wastewater network are not properly synchronized.

To address these issues, a graph-based representation can be employed. This representation ensures connectivity between different network components and eliminates inconsistencies. By using a graph structure, nodes representing manholes, structures, pumps and other components can be connected by edges representing pipes. This graph-based approach accurately reflects the interconnected nature of the wastewater network. Additionally, the graph-based representation helps to complete data by addressing missing nodes in the GIS representation, ensuring a more comprehensive view of the wastewater network. Moreover, the graph-based representation retains all information from the GIS representation, enabling the generation of GIS representations whenever needed. It also facilitates the integration of new information into the graph, making it a flexible and scalable solution for wastewater network management.

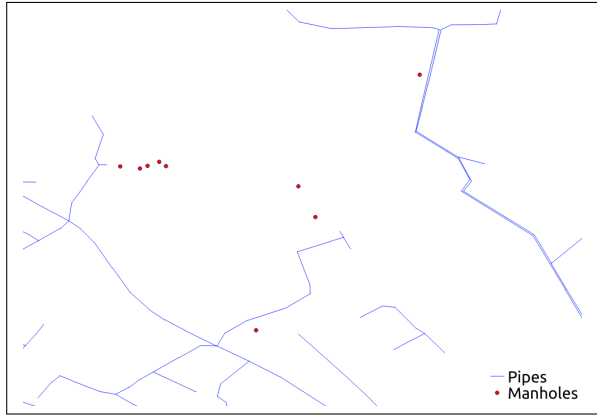


Fig. 2: Inconsistencies between databases

In the following section, the details of the graph-based representation and its benefits in maintaining the connectivity and integrity of wastewater network data will be presented.

3 Our graph-based representation

Figure 3 depicts the general process of constructing a graph-based representation for wastewater network data. The inputs comprise a database of pipes and databases of nodes, encompassing various components such as manholes, structures, pumps, fittings and more — all represented by node geometry.



Fig. 3: General process

The algorithm begins by extracting the intersections between each pipe in the pipes database and the nodes in the nodes database. This ensures that the connections between pipes and nodes are accurately identified. After detecting the intersections, the resulting nodes are connected or inserted into the graph. During this step, verification is performed to determine if the extremities of each pipe have intersections. If no intersections are found, dummy nodes are added to ensure connectivity within the graph, with edges representing the pipes.

Overall, the output of this process is a graph-based representation that accurately depicts the relationships between the pipes and nodes in the wastewater network.

3.1 Formal representation

The algorithm takes the pipes and nodes databases as inputs. These inputs can be formally represented as follows:

- The Pipes Database PD is a collection of line geometries that represent sewer pipes $PD = \{p_1, \dots, p_i, \dots, p_k\}$. Each pipe p_i is identified by a unique identifier i , and the total number of pipes in the database is denoted by k .
- The Nodes Database ND is a collection of point geometries that includes all the point type components: manholes, structures, pumps, fittings, accessories and treatment plants $ND = \{n_1, \dots, n_j, \dots, n_m\}$. Each element n_j has an attribute *type*, allowing us to identify the type of component. Each node n_j is identified by a unique identifier i , and the total number of nodes is denoted by m .

The algorithm generates a graph denoted as $G = (V, E)$ as its final output, where:

- V is a finite set of nodes that contains elements from ND and dummy nodes that will be added to ensure connectivity and complete missing nodes.
- $E \subseteq V \times V$ is a finite set of edges that represents connections between two nodes, indicating that these connections correspond to the pipes from PD . Each edge in the graph represents the relationship between two nodes, reflecting the pipes that connect them in the wastewater network representation.

3.2 Proposal algorithm

The graph construction algorithm (Algorithm 1) initializes an empty graph G with sets of nodes V and edges E . It iterates over each pipe p_i in the pipes database PD . For each pipe p_i , the algorithm follows these steps:

Algorithm 1 Graph Construction

```

1: procedure GRAPHCONSTRUCTION( $PD, ND, Distance$ )
2:    $V, E \leftarrow \{\}, \{\}$ 
3:    $G \leftarrow (V, E)$ 
4:   for  $i = 1$  to  $k$  do
5:      $Buffer \leftarrow CreateBuffer(p_i, Distance)$ 
6:      $Intersections \leftarrow ExtractIntersections(ND, Buffer)$ 
7:      $Ext1 \leftarrow RandomExtrimity(p_i)$ 
8:      $SortedInter \leftarrow DistanceAndSort(Intersections, Ext1)$ 
9:      $SortedInter \leftarrow AddDummyNodes(SortedInter)$ 
10:     $G \leftarrow ConnectionToGraph(SortedInter, G)$ 
11:  return  $G$ 

```

Step 1: Create a buffer. In GIS a "buffer zone" is defined by an area around a geographic feature that contains every points within a specified distance from that feature. In this step, a buffer is created around each pipe p_i by applying a distance that accounts for the physical dimensions of the pipe and any potential shifting of the nodes that are near to the pipe.

For example, in Figure 4, the blue line represents the pipe, and a buffer is created around it using a distance. The resulting yellow buffer represents the extended region around the pipe that encompasses its potential location within the given distance.

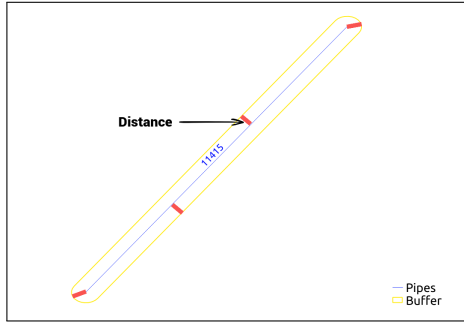


Fig. 4: Create a buffer

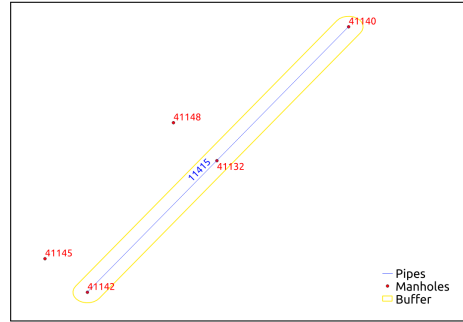


Fig. 5: Intersections detection

Step 2: Extract Intersections. In a real-world of wastewater networks, pipes are connected to nodes. This step aims to detect intersections between the buffer created around the pipe p_i and the nodes databases ND , to identify their connections.

The result is a set of nodes that must be connected to the corresponding pipe. For example, in Figure 5, the intersection between the buffer of pipe 11415 and the ND is detected, resulting in a set of nodes denoted as Intersections = $\{41132, 41142, 41140\}$. These nodes represent the set of nodes that must be connected to the pipe p_i , which is the pipe 11415 in this example.

Step 3: Calculate Distances and Sort Intersections. In this step of the algorithm, the Euclidean distance is calculated between one of the pipe's extremity $Extr1$ and each of the nodes intersecting the buffer (see algorithm 2). The Euclidean distance is the straight-line distance between two points in a two-dimensional space; it is suitable for cases where the distances between points are relatively small on surface Earth, which is the case in our situation with nodes that are near to each other in a wastewater network.

Using Euclidean distance, the proximity of each intersection node to one of the extremities of the pipe, labeled as $Extr1$, can be determined. Sorting

the intersection nodes based on their distances allows us to establish a logical order that reflects their positions relatively to the pipe’s extremities. This step is essential for checking the intersections with the extremities of the pipe.

Algorithm 2 Calculate Euclidean Distance and Sort Intersections

```

1: procedure DISTANCEANDSORT(Intersections, Extr1)
2:   distances  $\leftarrow$  {}
3:   for node in Intersections do
4:     x, y  $\leftarrow$  coordinates of node
5:     distance  $\leftarrow$   $\sqrt{(x - Extr1.x)^2 + (y - Extr1.y)^2}$ 
6:     Append distance to distances
7:   SortedInter  $\leftarrow$  Sort intersections based on distances
8:   return SortedInter

```

Step 4: Add dummy nodes. This step is crucial for ensuring the connectivity of wastewater networks (see algorithm 3). To construct graphs based on the intersections, it is essential to have a node near to each extremity of the pipe p_i . However, this condition is not always satisfied. It can occur that the set of intersections is empty ($|SortedInter| = 0$ in algorithm 3), or that certain nodes are present but not located at the extremities of the pipe p_i .

In such scenarios, it becomes necessary to add dummy nodes to the extremities of the pipe. These dummy nodes serve as virtual representations of the missing nodes, facilitating the connection of the pipe to the overall wastewater network.



Fig. 6: Example of a pipe with only one node intersecting the buffer area in manholes database, thus one missing extremity.

In Figure 6, it can be observed that there is only one node inside the buffer around pipe number 13360, which is near one extremity of the pipe. However, there is no node near the other extremity. To maintain connectivity with other

Algorithm 3 Adding Dummy Nodes

```

1: procedure ADDDUMMYNODES(SortedInter,  $p_i$ , Threshold)
2:   Ext1  $\leftarrow$  the first extremity of  $p_i$ 
3:   Ext2  $\leftarrow$  the second extremity of  $p_i$ 
4:   if  $|SortedInter| = 0$  then
5:     SortedInter  $\leftarrow$  AddFirst(Type = Dummy, coord = (Ext1.x, Ext1.y))
6:     SortedInter  $\leftarrow$  AddLast(Type = Dummy, coord = (Ext2.x, Ext2.y))
7:   else if  $|SortedInter| = 1$  then
8:     node  $\leftarrow$  node in SortedInter
9:     distance1  $\leftarrow$  EuclideanDistance(node, Ext1)
10:    distance2  $\leftarrow$  EuclideanDistance(node, Ext2)
11:    if distance1  $\geq$  Threshold and distance2  $\geq$  Threshold then
12:      SortedInter  $\leftarrow$  AddFirst(Type = Dummy, coord = (Ext1.x, Ext1.y))
13:      SortedInter  $\leftarrow$  AddLast(Type = Dummy, coord = (Ext2.x, Ext2.y))
14:    else if distance1  $\geq$  Threshold then
15:      SortedInter  $\leftarrow$  AddLast(Type = Dummy, coord = (Ext1.x, Ext1.y))
16:    else if distance2  $\geq$  Threshold then
17:      SortedInter  $\leftarrow$  AddFirst(Type = Dummy, coord = (Ext2.x, Ext2.y))
18:  else if  $|SortedInter| \geq 2$  then
19:    firstNode  $\leftarrow$  first node in SortedInter
20:    lastNode  $\leftarrow$  last node in SortedInter
21:    if EuclideanDistance(firstNode, Ext1.x, Ext1.y) > Threshold then
22:      SortedInter  $\leftarrow$  AddFirst(Type = Dummy, coord = (Ext1.x, Ext1.y))
23:    if EuclideanDistance(lastNode, Ext2.x, Ext2.y) > Threshold then
24:      SortedInter  $\leftarrow$  AddLast(Type = Dummy, coord = (Ext2.x, Ext2.y))
25:  return SortedInter

```

pipes, a dummy node is necessary for the missing extremity. The introduction of this dummy node serves the purpose of ensuring the connection of the pipe with the remainder of the network while preserving the integrity of the wastewater flow representation.

Step 5: Connecting to the graph The final step involves connecting the sorted intersections into the graph. After completing the step 4, the sorted intersections set is guaranteed to have at least two nodes that must be connected to the extremities of the pipe p_i . These nodes can either be from the original node set ND or dummy nodes that were added to ensure connectivity.

In this step the algorithm 4 begins by selecting the first node in the set of sorted intersections as the source node and adds it to the set of vertices in the graph G . Next, the algorithm iterates through the remaining nodes in this set. For each node, it creates an edge between the current source node and the destination node. Each edge in the graph represents a pipe in wastewater networks.

In the illustrated example (Figure 7), the buffer zone created around pipe number 11415 is taken into consideration. This buffer intersects three nodes,

Algorithm 4 Connecting to the graph

```

1: procedure CONNECTIONTOGRAPH(SortedInter, G)
2:   sourceNode  $\leftarrow$  first node in SortedInter
3:    $V \leftarrow V \cup \{sourceNode\}$ 
4:   for node in SortedInter excluding the first node do
5:     destinationNode  $\leftarrow$  node
6:      $E \leftarrow E \cup \{(sourceNode, destinationNode)\}$ 
7:     sourceNode  $\leftarrow$  destinationNode
8:   return G

```

namely $Intersections = \{41142, 41132, 41140\}$. These nodes represent the necessary connections to be established with pipe 11415. Before establishing these connections, a sorting process is performed and intersections with pipe extremities are verified. In this case, the intersections adequately cover the extremities, thus avoiding the need to introduce dummy nodes. Consequently, the graph can be constructed appropriately by connecting the sorted nodes to pipe 11415. It can be observed that a single pipe can result in the creation of two edges in the graph, as evidenced here with edges (41142, 41132) and (41132, 41140), owing to the connection of the middle node numbered 41132 to the pipe.

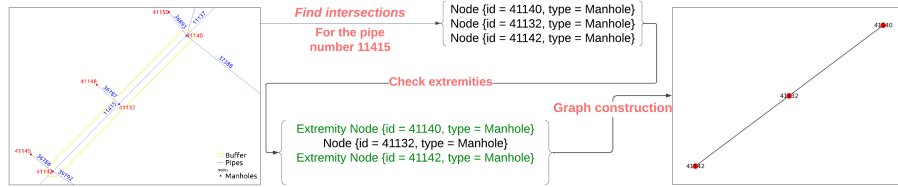


Fig. 7: Application of the algorithm: example with three nodes intersecting the buffer area including the pipe's extremities.

In the second example (Figure 8), involving pipe number 13360, a different situation arises. Only one intersection is found between this pipe and the nodes database, which happens to be located at one extremity. Unfortunately, no intersection exists at the other extremity. To ensure the proper connectivity of pipe 13360 within the graph, a dummy node with the label 107965 is introduced. This dummy node acts as a placeholder, allowing the inclusion of the missing extremity in the graph representation. Therefore, the resulting graph reflects the connection between node 39488 and the dummy node 107965, representing pipe 13360.

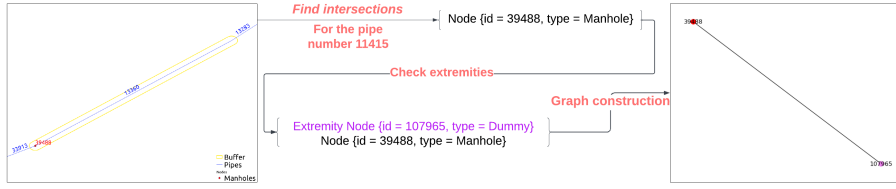


Fig. 8: Application of the algorithm: example with three nodes intersecting the buffer area including the pipe’s extremities.

4 Experiments

The primary objective of our study is to develop a graph-based representation derived from GIS datasets of wastewater networks. As shown in Table 1, the results of experiments conducted on four distinct real datasets of wastewater networks are presented. These datasets all represent the same geographical zone but originate from different providers. Each dataset consists of multiple components, and for each of these datasets, both the traditional GIS representation, also known as the separate databases representation, and the innovative graph-based representation derived from the GIS representation are available. Additionally, the difference in the number of objects between the SIG and graph-based representations is presented.

Datasets	Representation	Components							
		Pipes	Manholes	Pumps	Structures	Treatment plants	Accessories	Fittings	Dummy nodes
Dataset 1	GIS	43804	42007	217	760	14	0	451	0
	Graph	43803	41976	217	760	14	0	451	405
	Differences	-1	-31	0	0	0	0	0	+405
Dataset 2	GIS	47059	45411	245	801	14	427	0	0
	Graph	47064	45410	245	800	14	427	0	188
	Differences	+5	-1	0	-1	0	0	0	+188
Dataset 3	GIS	35535	34492	0	236	0	119	0	0
	Graph	35774	34434	0	141	0	27	0	1202
	Differences	+239	-58	0	-95	0	-92	0	+1202
Dataset 4	GIS	37330	37683	0	285	0	216	0	0
	Graph	37922	37518	0	182	0	129	0	38
	Differences	+592	-165	0	-103	0	-97	0	+38

Table 1: From GIS to graph: Comparison of components number

Concerning dataset 1, most of the components from the GIS representation are successfully preserved in the graph. However, there are very few components missing in the graph representation. This happens when the components are located at a considerable distance from the nearest pipes in the GIS representation, making it challenging to establish direct connections between them. Figure 2 provides a visual representation of this scenario, highlighting the manholes that are not in close proximity to any pipes. The algorithm ensures that each pipe in the GIS representation is represented as an edge in the graph-based representation. However, in certain cases, the number of edges in the graph may be less than the number of pipes in the GIS representation. This can happen when there are duplicated pipes in the GIS representation, where two pipes have the same extremities. In such cases, the algorithm inserts only one of these duplicated pipes into the graph, resulting in a difference in the number of edges. In Figure 9, it is evident that two pipes, namely 37974 and 16990, share the same extremities, specifically 32451 and 30017. As a result of removing these duplicated pipes during the graph-based representation, only one pipe with these extremities is inserted into the graph. This explains the negative difference in the number of edges compared to the pipes in the GIS representation for Dataset 1, where 21 duplicated pipes were encountered.

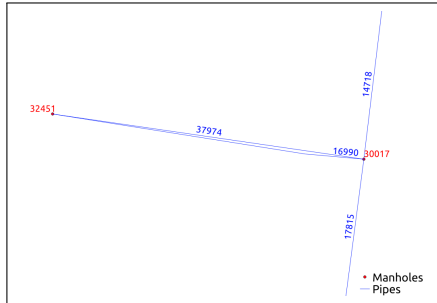


Fig. 9: Duplicated pipes in GIS

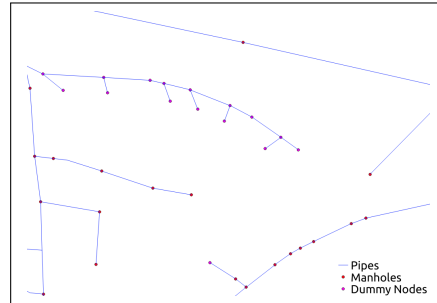


Fig. 10: Dummy nodes

Furthermore, a total of 405 dummy nodes were incorporated into the graph for dataset 1. Figure 10 illustrates an example of dummy nodes added to the graph. The nodes highlighted in rose color represent the dummy nodes. In Figure 10, the GIS representation of the wastewater network is depicted, revealing the absence of nodes at the extremities of certain pipes. Without the presence of manholes or other components, it is challenging to establish a complete branch in the network. Hence, the addition of dummy nodes becomes necessary to ensure connectivity and complete the representation of the wastewater network data.

Indeed, the number of connected components in the resulting graphs is an important criterion for evaluating the accuracy of the graph-based representation. In real-world wastewater networks (see the Figure 11), it is common to

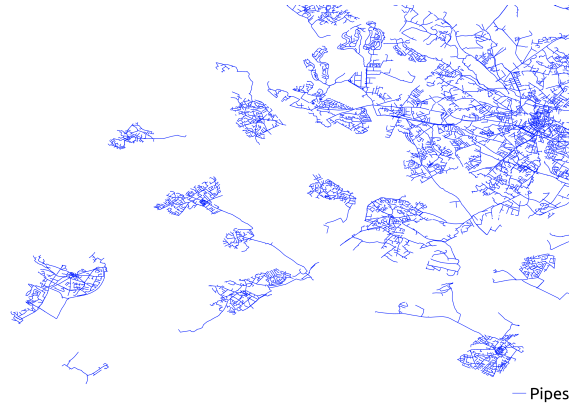


Fig. 11: Components in GIS representation

have disconnected components due to various factors such as the physical layout of the network.

In some cases, the disconnected components in the GIS representation may be the result of inserting errors or shifting errors during data collection or digitization processes. These errors can lead to gaps in the network representation, making it challenging to accurately assess the connectivity of the wastewater network.

Datasets	Representation	Connected components
Dataset 1	Pipes (GIS)	166
	graph	151
Dataset 2	Pipes (GIS)	475
	graph	168
Dataset 3	Pipes (GIS)	200
	graph	158
Dataset 4	Pipes (GIS)	177
	graph	81

Table 2: Connected components

In the graph-based representation, one of the main goals is to identify and correct errors present in the GIS databases, which may lead to disconnected components in the resulting graph. The small threshold of 0.1 meter used for the buffer distance is designed to capture and address these errors of small shifting or inserting errors.

By applying this threshold, the algorithm can identify nodes that are close to the pipes in the GIS representations but may not have a direct connection due to minor inaccuracies in their positions. The buffer distance helps in identifying these nearby nodes and ensures that they are connected by inserting appropriate

edges in the graph. As a result of this error correction process, the number of components in the resulting graph is often less than the number of components in the GIS representation (see Table 2). This reduction in the number of components indicates that the graph-based representation is successfully identifying and addressing the errors in the GIS representation, resulting in a more accurate and coherent representation of the wastewater network.

In Figure 12, an example of the graph-based representation of wastewater networks data is presented, where different colors of nodes symbolize specific types of components, with red indicating manholes. The edges represent the pipes, and their interconnections depict the real-world relationships found in wastewater networks. Furthermore, the graph-based representation retains all information about components from the GIS representation, encompassing both attribute and spatial details. This preservation enables us to generate GIS representations from the graph whenever required.

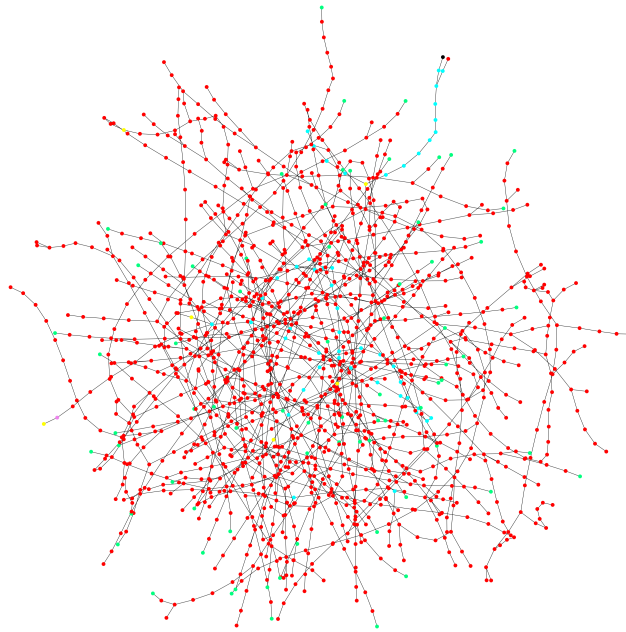


Fig. 12: Graph-based representation of wastewater networks

5 Conclusion

In this study, a graph-based approach for representing wastewater networks, derived from GIS datasets, has been proposed. Our graph-based representation accurately captures the interconnected nature of the network. By conducting experiments on real-world datasets, the efficiency and effectiveness of the method have been demonstrated. It showcases the ability to preserve most components from the GIS representation while resolving missing connections attributed to significant distances.

In future endeavors, our plans involve improving our graph-based representations by integrating water circulation within the pipes. This entails constructing oriented graph-based representations that reflect the flow direction and dynamics of wastewater through the network. Taking into account the directionality of pipes enables us to delve deeper into flow behavior analysis and optimize the network accordingly. Furthermore, since all the datasets we possess represent the same geographical zone, our objective is to explore the fusion of these oriented graphs. The fusion process will enable us to synthesize comprehensive representations that encompass the entire wastewater network for the given region.

Acknowledgements This research has received support from the European Union’s Horizon research and innovation programme under the MSCA-SE (Marie Skłodowska-Curie Actions Staff Exchange) grant agreement 101086252; Call: HORIZON-MSCA-2021-SE-01; Project title: STARWARS (STormwAteR and WastewAteR networkS heterogeneous data AI-driven management).

This research has also received support from the French national project ANR (Agence Nationale de la Recherche) CROQUIS (Collecte, représentation, complétion, fusion et interrogation de données hétérogènes et incertaines de réseaux d’eaux urbains).

References

1. LIU, Mingfei, LI, Xinyu, LI, Jie, et al. A knowledge graph-based data representation approach for IIoT-enabled cognitive manufacturing. *Advanced Engineering Informatics*, 2022, vol. 51, p. 101515.
2. ZHOU, Hao, YANG, Xu, FAN, Mingyu, et al. Static-dynamic global graph representation for pedestrian trajectory prediction. *Knowledge-Based Systems*, 2023, p. 110775.
3. Fall, A., Fortin, M.-J., Manseau, M., & O’Brien, D. (2007). Spatial Graphs: Principles and Applications for Habitat Connectivity. *Ecosystems*, 10(3), 448–461. doi:10.1007/s10021-007-9038-7
4. SONAWANE, Sheetal S. et KULKARNI, Parag A. Graph based representation and analysis of text document: A survey of techniques. *International Journal of Computer Applications*, 2014, vol. 96, no 19.
5. MAUGEY, Thomas, ORTEGA, Antonio, et FROSSARD, Pascal. Graph-based representation for multiview image geometry. *IEEE Transactions on Image Processing*, 2015, vol. 24, no 5, p. 1573-1586.