



A Depth-First Search Algorithm for Optimizing the Gravity Pipe Networks Layout

Gustavo Paiva Weyne Rodrigues¹  · Luis Henrique Magalhães Costa¹ ·
Guilherme Marques Farias² · Marco Aurélio Holanda de Castro³

Received: 24 March 2019 / Accepted: 2 October 2019 /

Published online: 15 November 2019

© Springer Nature B.V. 2019

Abstract

The layout is displayed in one of the most complex tasks in the gravity pipe network project because there are several factors to consider and often a choice of unassociated or smaller layout. Currently, the designer's experience is needed so different layout alternatives be analyzed to reduce the depths of the network. Generally, this operation is manual and does not ensure the best result. For this research, a depth-first search algorithm was presented, which allows the optimization of the layout of a gravity pipe network, assessing the topographic conditions of the manholes (nodes), leading to a layout that has the sum of lower unfavorable slopes. A hypothetical and a real network were used. The computational time required was considered negligible. The results showed a robust model, which works for the complete layout of any network, of any size, resulting in the lowest possible depths.

Keywords Gravity Pipe networks · Optimal layout · Depth-first search algorithm · Search algorithms

✉ Gustavo Paiva Weyne Rodrigues
gustavo_weyne@uvanet.br

Luis Henrique Magalhães Costa
luishenrique.uva@gmail.com

Guilherme Marques Farias
guilhermemf15@gmail.com

Marco Aurélio Holanda de Castro
marco@ufc.br

¹ Department of Civil Engineering, Vale do Acaraú State University, Sobral, Ceará, Brazil

² Department of Civil Engineering, Federal Institute of Education, Science and Technology of Ceará, Fortaleza, Ceará, Brazil

³ Department of Civil Engineering, Federal University of Ceará, Fortaleza, Ceará, Brazil

1 Introduction

The layout is one of the most complex tasks in a gravity pipe network design. There are several factors that must be taken into consideration and often the choice of the layout does not result in the lowest cost. The execution of gravity pipe networks is one of the costliest items in the execution of hydraulic infrastructure works. A typical gravity pipe network system requires the design from street layout and terrain topography, which ends at an end point. In this type of project, in the phase of layout creation and subsequent designing of the network, special attention must be paid to achieve the minimum cost.

According to Diogo and Graveto (2006), in relation to computational aspects, the optimization of a gravity pipe network layout, regarding complexity, can be directed as follows: it starts with an undirected initial base node graph and a set of feasible solutions in order to find the ideal overall configuration, composed of directed tree forest. In each network there are several directed trees, represented by manholes (nodes) and pipes, culminating in a forest.

Although, in principle, the determination of the ideal path selection can be performed deterministically by a simple cost comparison on all paths that can be traversed in the network layout, according to Mai and Evans (1984), the optimization enumerating the branches of a network had a complexity that was evident in larger networks because of the vast number of feasible paths.

Researches on the optimization of the layout of gravity outflow networks are scarcer. On the other hand, several computational methods have been proposed to optimize the design. These methods include linear programming (LP), nonlinear programming (NLP), dynamic programming (DP), search algorithms (SA), genetic algorithms (GA), differential evolution (DE), particle swarming optimization (PSO), evolutionary algorithms (EA), ant colony optimization algorithm (ACO) etc.

In the 1960s, the first researches on network sizing optimization began with the works of Deininger (1966), Holland (1966) and Liebman (1967), the first two using NLP and the third with heuristic procedures.

In the following decade, Dajani (1971) and Velon (1971) used LP. Dajani et al. (1972), using NLP, proposed a model for the determination of the optimal longitudinal profile for a given network layout plan.

Walsh and Brown (1973), based on the concepts presented by Haith (1966), developed a DP model that considers more precise cost functions and allowed that the outflow occurred in partially filled pipes. Mays and Yen (1975) developed two computational models to optimize sewage collection networks using discrete differential dynamic programming (DDDP).

With the advancement of technology in the 1980s, because of computer processors, several researches were carried out in DP, as in Kulkarni and Khanna (1985) and Nzewi et al. (1985). The GAs gained notoriety and were the subject of study by Heaney et al. (2000), Afshar (2006), Pan and Kao (2009) and Haghghi and Bakhshipour (2012). Walters and Lohbeck (1993) and Walters and Smith (1995) proposed the use of the genetic algorithm and an evolutionary strategy for layout optimization for various tree graph applications, such as sewage, irrigation and drainage. Other relevant works referring to the sewage collection network layout were those developed by Diogo and Graveto (2006) and Steele et al. (2016).

2 Problem Definition

The optimization of the gravity networks layout involves choosing the arrangement of the network always seeking for minimizing the excavation volume, that is, trying to follow the natural slope of the ground. In practice, due to various aspects, this is not always possible. Gravity pipes shall follow the slope of the ground surface and follow a direct route to the discharge point as far as topography and layout of streets allow that so. The main goal of this research was to develop a depth-first search (DFS) algorithm to optimize the gravity flow network layout, adopting a direction for the flow in which there is less influence of the slope in the pipes whose ground elevations are not favorable. The specific goals are that, from the imposition of one or more outlet nodes, the proposed model determines the optimal path regarding the slope of the ground, dividing the network in a number of basins equal to the number of previously established outlet nodes.

The results of this research do not refer to the designing, but to the optimization of gravity pipe networks layout, unlike some research cited above. Initially, generating the layout manually may seem like a simple task; however, choosing the ideal layout requires thorough attention since the slopes have a relevant aspect regarding the reduction of depths.

The innovation of this research is to present a model that has the following characteristics for case studies:

- The decision variable is the slope of the ground;
- All streets have pipes;
- The topography may be irregular; that is, it will not always be favorable to the natural inclination of the ground;
- The designer has the freedom to perform the layout anyway and will always have the ideal solution for the entire network;
- It does not depend on the number of pipes and outlet nodes;
- If the network has more than one outlet node, the model automatically divides the basins.

3 Methodology

3.1 Graph Theory - Basic Concepts

For a better understanding of the proposed methodology, some basic definitions of graph theory are presented. The notions of vertex, edge, adjacent edges, path, connections, loop and trees are properly defined in Boaventura Netto and Jurkiewicz (2009).

According to Villas et al. (1993), graph theory is the part of mathematics dedicated to studying the relationships between entities (objects), which have relevant characteristics. A graph, for Boaventura Netto and Jurkiewicz (2009), is a mathematical structure formed by two objects: sets of vertices and edges. The graph can be directed and non-directed. The directed ones are those whose edges indicate direction of the connection and the non-directed ones are those whose arcs indicate a double connection.

Regarding minimum paths, the values that determine path lengths are associated with edges, not vertices. The value that is associated with each edge is called weight. A directed graph with weight at the edges is called a directed weighted graph. According to Cormen (2013), weight is a generic term associated with edges with values and the

weight of a path is the sum of edge weights. These graphs represent situations where there is distance, time, flow or capacity.

3.2 Depth-First Search Algorithm Concept

According to Cormen et al. (2009), search algorithms are those that perform a pathway to explore a graph by examining all vertices and edges. Those algorithms explore edges from the most recently discovered vertex from which unexplored edges still emerge. DFS is used to go through or fetch items within data structures, graphs, or trees, whose basic characteristic is to go through all child nodes to the root node as deep as possible and then perform a backtracking. Russell and Norvig (2010) mention that this type of search can be implemented in several ways, such as going through the graph or tree while there are unvisited children. An illustration of a DFS algorithm directed graph is shown in Fig. 1.

3.3 Programming Language

The programming language used to create the layout optimization algorithm was C# and the IDE (Integrated Development Environment) utilized was Microsoft Visual Studio. The C# language was chosen because it is a clear and easy-to-understand language, it has object-oriented programming features, and does not present compatibility issues with the Windows environment.

The computer used in the design and model's execution has an Intel® Core™ i7-5500u processor, 2.40GHz, 4 MB cache, two cores, four processors, and 8GB of RAM memory.

4 Problem Statement

4.1 Development of the Model

From a drawn network, that is, with established topography, numbering of the beginning and ending nodes (manholes), as well as the extensions and numbering of the pipes, the research

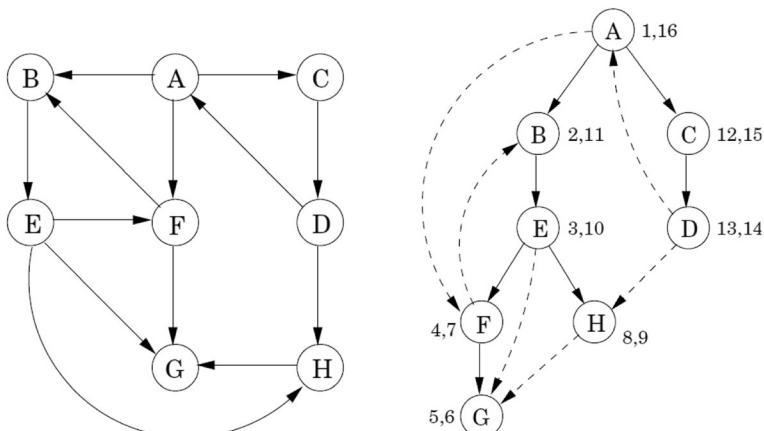


Fig. 1 DFS on a directed graph. Source: Dasgupta et al. (2008)

model can be started. There is the need for two text files with input data containing the features of the pipes and nodes mentioned above. The first file contains the pipe data and the second file the node data. It is important to highlight that there is not always the condition for the flow to be conducted through the natural inclination, as, for example, in a pipe where there is no upstream flow contribution, that is, where there is only one outlet pipe and none of entrance to a given node (these pipes necessarily have to conduct the flow even against the natural topographic slope).

Figure 2 illustrates the geometry of two consecutive pipes. All dimensions are always measured in relation to the same reference level and N_i , D_i , UPE_i , DPE_i , NE_i , are node, pipe diameter, upstream pipe elevation, downstream pipe elevation and node elevation, respectively.

The calculation of the slope of any pipe is given by Eq. 1.

$$S = (NE_i - NE_{i+1}) / L \tag{1}$$

Where S is ground slope [L/L]; NE_i is upstream node elevation [L]; NE_{i+1} is downstream node elevation [L]; L is pipe length [L].

There are three situations to be assessed in relation to the slope gradient at the layout of a pipe in a gravity network:

1. The ground level of the upstream node is larger than the downstream one: favorable ground (positive slope, $S > 0$).
2. The ground level of the upstream node is equal to the downstream one: flat terrain (zero slope, $S = 0$).
3. The ground of the upstream node is smaller than the downstream one: unfavorable terrain (negative slope, $S < 0$).

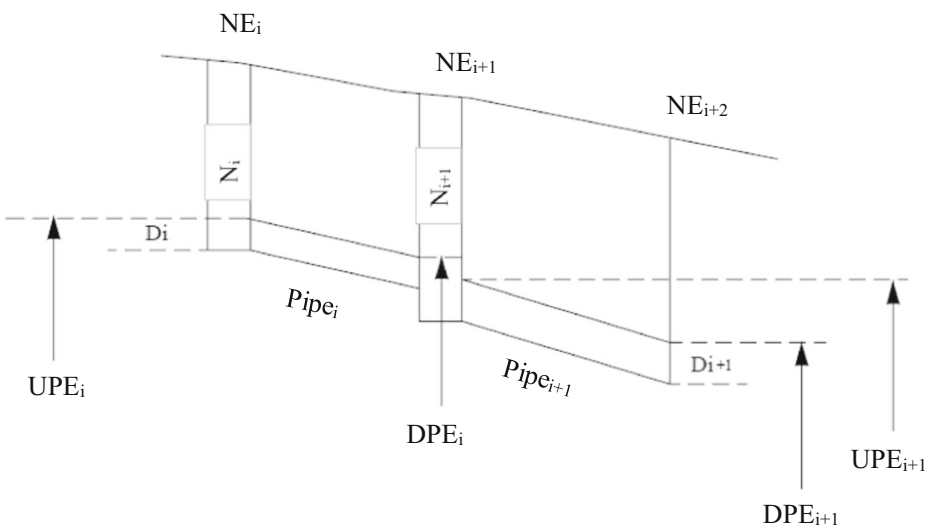


Fig. 2 Geometry of two consecutive pipes in a gravity network. Source: Adapted from Gameiro (2003)

Pipes that have null or negative slope are the main target of the research. Each pipe whose slope is null or negative will have its respective unfavorable area calculated, presented in Eq. 2. If the slope is null then the value of Eq. 2 becomes 0, in case negative is the resulting value itself. This area will represent the weight of each edge of a weighted directed graph as explained below.

$$A = (NE_i - NE_{i+1}) \cdot L \tag{2}$$

The objective function of the algorithm is to minimize the sum of unfavorable areas of the pipes, using Eq. 3.

$$\text{Minimize } A_{\text{total}} = \sum_{i=1}^{SP} A \tag{3}$$

Where A_{total} is the sum of unfavorable areas [L^2]; A is unfavorable area [L^2]; SP is the number of pipes on the path up to the stopping point.

The model was divided into two phases: the first, for data reading and initial verification of network parameters; the second is the beginning of the depth-first search algorithm methodology. In the first phase, from reading the data of the pipes and nodes contained in the two input data files, the model starts the procedures for the complete network search. As a first step, the algorithm excludes outlet nodes from the network.

The second step of the algorithm imposes the direction of flow in all parts of the network by the natural slope of the ground, without a thorough analysis of the consequences. From this imposition, the third step is to determine the amount of outflow stream and inflow stream pipes for each node. After this accounting, the fourth step consists in searching for nodes that do not have outflow stream pipes and that only one is inflow stream pipe. The pipes of those nodes will have the reverse flow direction. The number of inflow stream and outflow stream pipes for each node is counted again, which configures the fifth step. Following the above instructions, it may occur that at least one node has two or more inflow stream pipes, but no outflow stream ones. This situation is a hindering factor in the hydraulics of a network, as there would be no downstream flow from this node. This type of case was called a confluent node, as shown in Fig. 3.

In the second phase, all confluent nodes of the network must be corrected. The model performs the search, generating n graphs, whose initial node (root) corresponds to each confluent node, where n is the number of confluent nodes in the network. From each root node, branching in the graph occurs according to the pipes arriving at this node. Then, the upstream nodes of these inflow stream pipes to the root node are analyzed. If there is only one outflow stream and one inflow stream pipe in those upstream nodes, the branch proceeds -

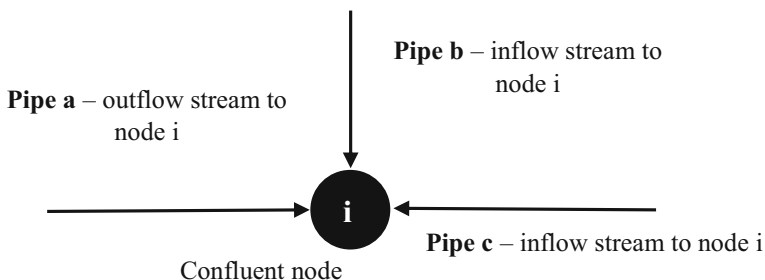


Fig. 3 Confluent node example

which forms a typical tree-like configuration. The criteria for stopping tree growth occurs when the branch reaches a node that has at least two outflow stream pipes or two inflow stream pipes from that same node, as shown in Fig. 4, where node *a* represents the stopping point.

The search algorithm, starting from the root node, explores all paths of the generated graph by calculating the unfavorable area, through Eq. 2, of each upstream pipe of the confluent node. This calculation corresponds to the weight of each edge. Then all those areas are summed up until the stopping point is reached (Eq. 3). Thus, each path will have an unfavorable area accumulated value, as in the example of Fig. 5, which has 3 paths. The weights between nodes 0–1, 0–2, 0–3, and 3–4 are 10, 15, 20, and 25, respectively, and therefore the path between nodes 0 and 1 is the one of lowest value. The higher this accumulated unfavorable area value, the greater will be the adverse topography which the network will have to traverse, and, consequently, the cost of excavation due to the high depths.

An important tool developed in this study was to prevent the adopted solution from creating a circuit (loop), that is, that there was no outlet for the flow in the network (Fig. 6). If a scenario like the one in Fig. 6 were adopted by the algorithm, it could result in a lower cost layout, but would not be hydraulically solved.

For this peculiar case, the code requires that all paths generated from each confluent node be traversed by performing the following check: the outlet node of each path cannot return to itself, as that would form a cyclic graph.

Assuming node 3 of Fig. 6 as the confluent, in order not to form a circuit, the generated paths whose final node is 4 must be analyzed, for example. In this case, mandatorily, from node 4, following the direction of the adopted flow, no path could reach node 4 itself. This tool works as a filter on the number of feasible solutions. All these paths must be eliminated as they are not hydraulically correct.

Figure 7 represents the graph growth flowchart (second phase of the algorithm) of each root node. The path represented by the solution of each confluent node reverses all directions of the traffic flow belonging to that node's graph. Subsequently, the pipes of this path can be

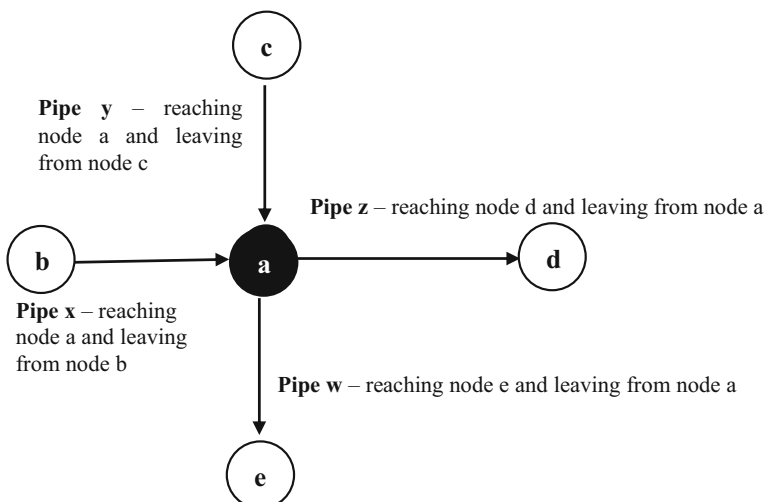
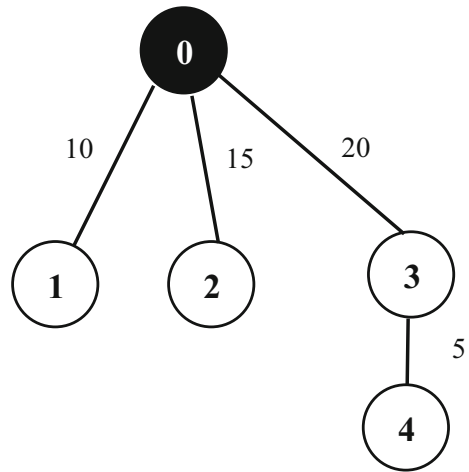


Fig. 4 Examples of branch stopping point

Fig. 5 Example of weighted directed graph



attributed to a boolean variable as true, since they can no longer vary in the continuation of the network analysis.

The strategy of the proposed model is to determine the best in the graph branches. The optimal solution is the one where the sum of the unfavorable slope areas has the lowest value for each confluent node (Fig. 7).

5 Results and Discussions

5.1 Case Study 1 - Hypothetical Network

In order to present the graphs derived from a gravity network layout, as well as to apply the proposed computational model, it is presented a case study containing a hypothetical 12-pipe

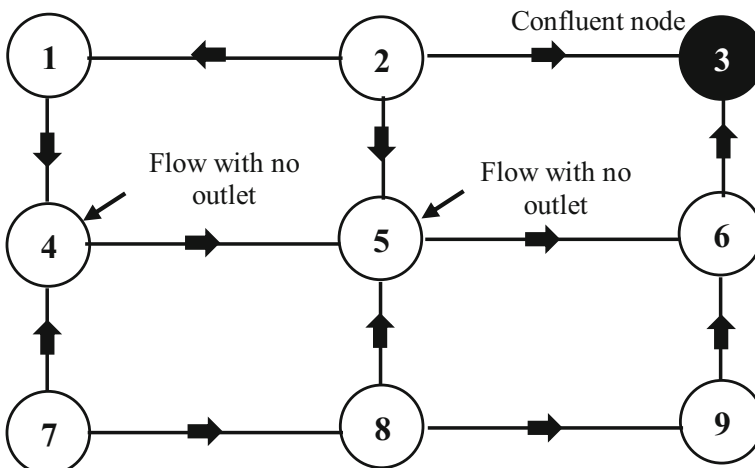


Fig. 6 Example of flow with no outlet

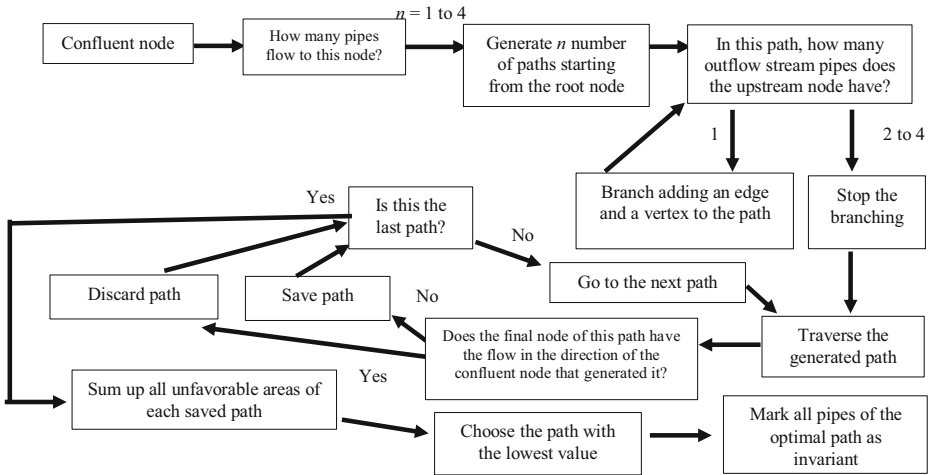


Fig. 7 Graph growth flowchart (second phase of the algorithm)

and 9-node network, adapted from Moeini and Afshar (2017), illustrated in Fig. 8. For model simulation, node 2 was considered to be the final one.

The same extensions of the mentioned authors were maintained; however, the node elevations (Table 1) were altered as a means of analyzing the algorithm. It should be noted that the order of nodes and their respective topographic elevations do not express the direction of the flow, since this will be defined by the model.

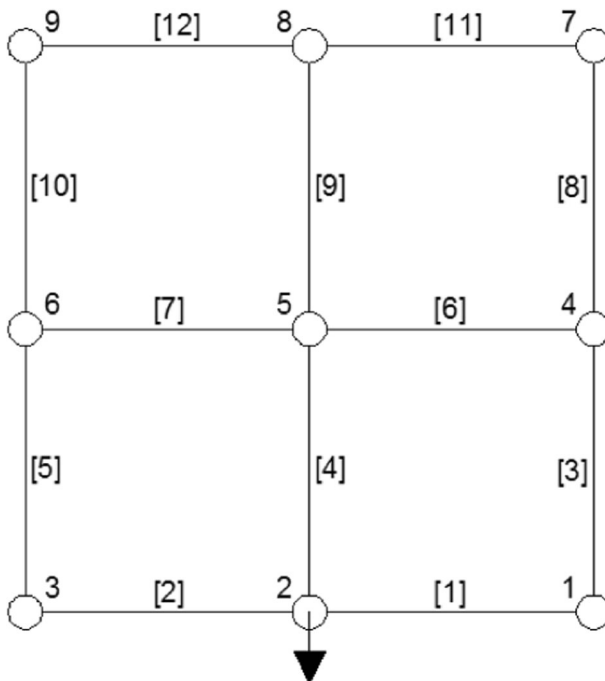


Fig. 8 Case Study 1 - Hypothetical Network. Adapted from Moeini and Afshar (2017)

Table 1 Characteristics of the hypothetical network pipes

Pipe	Node 1	Node 2	NE_i (m)	$N_{e_{i+1}}$ (m)	L (m)
1	2	1	9.45	9.60	100
2	3	2	9.20	9.45	100
3	4	1	9.40	9.60	100
4	5	2	9.80	9.45	100
5	6	3	10.00	9.20	100
6	5	4	9.80	9.40	100
7	6	5	10.00	9.80	100
8	7	4	10.20	9.40	100
9	8	5	10.40	9.80	100
10	9	6	10.60	10.00	100
11	8	7	10.40	10.20	100
12	8	9	10.40	10.60	100

Since the network has only one final node, there will be only one basin in the network. After the execution of the model, in the first phase, all the flow directions were determined. Two confluent nodes were located: 3 and 4 (Fig. 9). The arrows indicate the orientation of the flow obtained in the first phase.

In the second phase, the algorithm created the graphs from the confluent nodes and branched to the stopping points that were nodes 2 and 6. In this network two graphs were generated, which are illustrated in Fig. 10. For nodes 3 and 4, there are two and three possible paths, respectively.

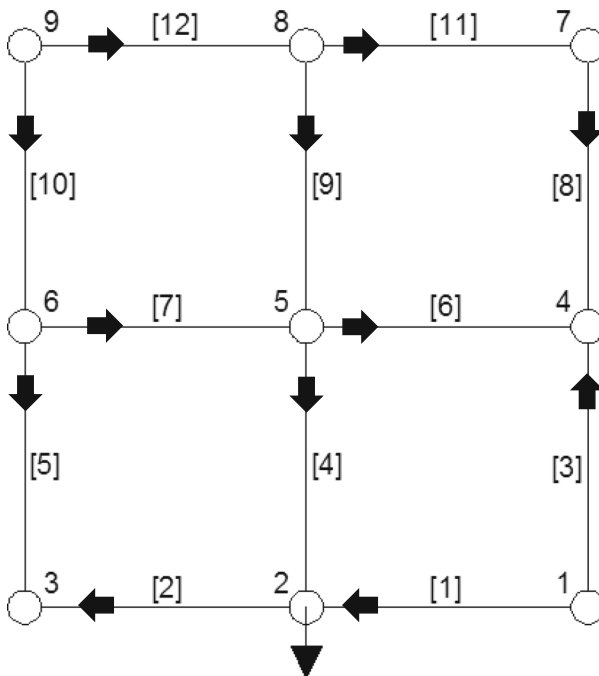


Fig. 9 Flow directions after the execution of the first phase of the algorithm for case study 1 - hypothetical network. Adapted from Moeini and Afshar (2017)

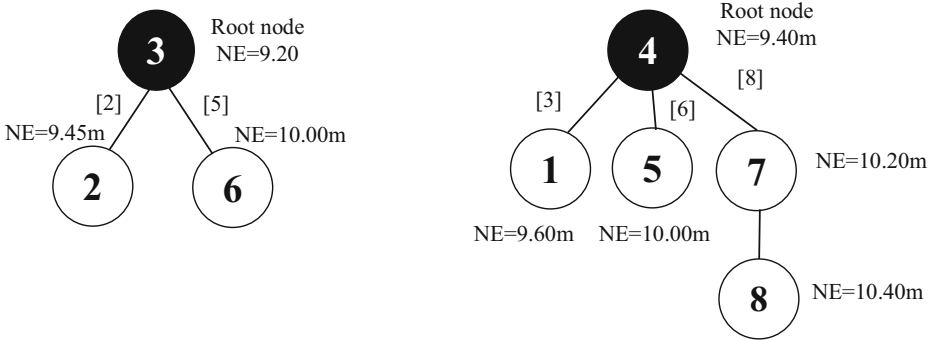


Fig. 10 Graphs with the paths to be traversed from case study 1

Attributing unfavorable area values to each edge of the graph results in Fig. 11, in which those values are presented as weights of a weighted directed graph.

The algorithm explores all graph paths by summing the weights of each edge. For confluent node 3, there are two paths: node 3 to 2 and node 3 to 6 with edge weight values of 25 and 80, respectively. Confluent node 4 has three paths: Nodes 4 to 1, Node 4 to 5, and Node 4 to 8, where the quantitative of the accumulated weights, in due order, are equal to 20, 40, and 100.

For each branch, the value of the accumulated weights was found, thus defining the path of least weight. For node 3, the flow direction of the in pipe 2 (node 3 to 2) was reversed. Regarding node 4, the flow direction in pipe 3 (node 4 to 1) was altered. The algorithm determined the optimal tracing (Fig. 12), attributing the new flow directions to the network, which would not be achieved if other paths had been chosen.

5.2 Case Study 2 – Real Network – Scenarios 1 and 2

To demonstrate the applicability of the proposed algorithm, it was used a real network (Fig. 13) from the municipality of Teodoro Sampaio in the state of São Paulo/Brazil. This network has 125 sections and 74 nodes. The decision to choose this network was that, due to the layout of

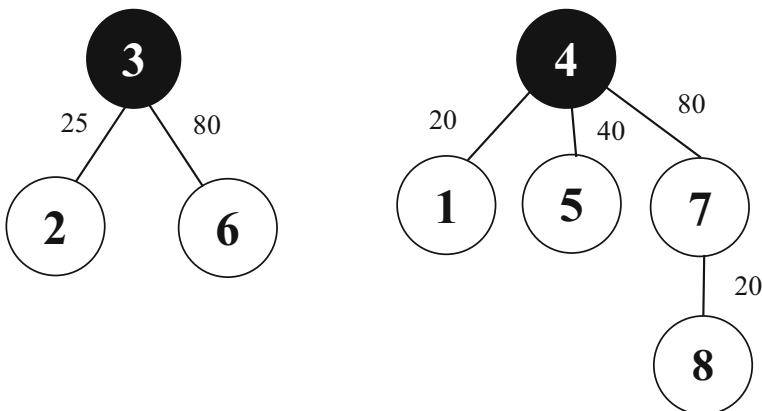


Fig. 11 Directed weighted graphs with the paths and respective weights to be summed from each root node - case study 1

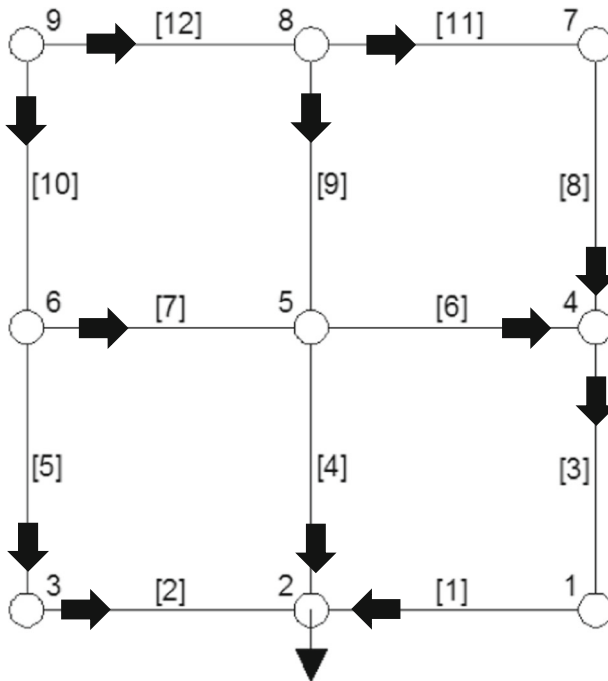


Fig. 12 Flow directions in the hypothetical network after the execution of the second phase of the algorithm. Adapted from Moeni and Afshar (2017)

the streets and uneven surface elevation contours, the network must have some pipes that should be designed against the natural slope of the ground. This will result in confluent nodes and their graphs, for which the accumulated weights must be determined and then the lowest value paths selected.

Two different scenarios were created for the application of the model. In the first scenario there is one outlet node: 51. In this configuration, for the flow to reach node 51, several pipes must have been designed against the natural slope of the ground.

In the second scenario two outlet nodes were arranged: 41 and 51. The goal of this proposition was to present the automatic division of basins that the model performs, determining the layouts with the lowest unfavorable slopes possible. The algorithm verifies each pipe and, according to the slopes and least cost path analysis, determines which basin it belongs to. For this scenario the pipes were divided into 2 basins.

The nodes elevations of the network are presented in Table 2.

In scenario 1, after the execution of the model, the network had 4 confluent nodes: 15, 41, 70 and 72. Obviously, for the entire flow to reach node 51, some pipes belonging to the aforementioned nodes must have the flow direction reversed.

Table 3 presents the confluent nodes with the respective adopted feasible paths. The proposed algorithm should analyze and determine which path for each confluent node will lead to the smallest sum of null or unfavorable slopes (column 6 of Table 3), which is the strategy of the present study.

In scenario 2, the algorithm automatically divided the two basins, determining the path with the lowest sum of slopes contrary to the natural ground. As it can be seen in Table 4, the only

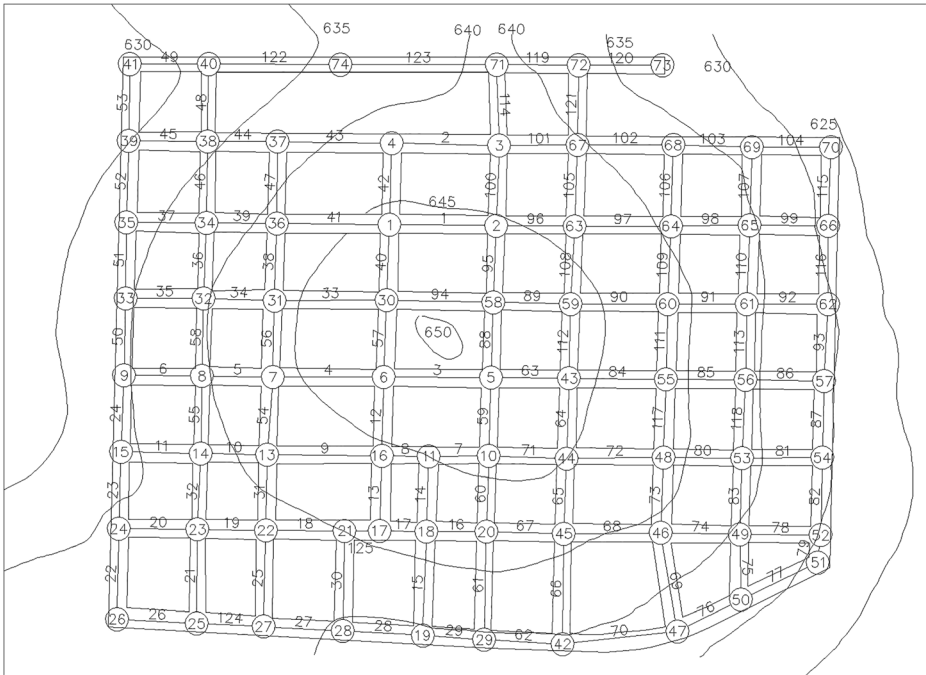


Fig. 13 Real network of the case study 2

difference between the two scenarios refers to node 41, since it is one of the outlet nodes of scenario 2, the model did not present a proposal to change the direction of flow against the slope of the ground, considering the slopes of pipes 49, 122 and 123 are favorable. The number of pipes of the basins of outlet nodes 41 and 51 were, respectively, 72 and 53.

Table 2 Node elevations of the real network - case study 2

Node	Elevation (m)	Node	Elevation (m)	Node	Elevation (m)	Node	Elevation (m)
1	645,97	20	642,08	39	624,53	58	648,14
2	645,40	21	640,00	40	628,37	59	646,09
3	641,77	22	639,39	41	621,64	60	641,36
4	641,54	23	638,13	42	633,83	61	635,89
5	648,85	24	636,44	43	646,36	62	630,52
6	647,89	25	639,00	44	644,94	63	644,19
7	643,09	26	639,00	45	641,39	64	639,19
8	639,50	27	637,28	46	638,67	65	635,49
9	633,96	28	634,79	47	632,06	66	629,44
10	645,61	29	634,10	48	641,49	67	640,23
11	644,91	30	648,45	49	634,38	68	636,51
12	644,15	31	643,42	50	631,24	69	633,19
13	640,92	32	639,61	51	629,88	70	626,44
14	638,01	33	633,79	52	636,54	71	640,00
15	633,30	34	637,90	53	630,59	72	637,07
16	644,15	35	631,49	54	641,76	73	633,07
17	640,71	36	641,11	55	636,09	74	636,73
18	641,49	37	637,78	56	630,94		
19	634,27	38	633,77	57	624,53		

Table 3 Chosen paths to the real network confluent nodes from case study 2 –scenario 1

Confluent node	Possible path number	Number of not feasible paths due to loop formation	Number of feasible paths	Pipe(s) of the feasible path(s)	Sum of the feasible path weight(s)
15	6	2	4	11 and 10	471.496
				24	41.910
				11 and 55	407.694
				23 and 22	391.286
41	6	5	1	49, 122 and 123	1772.191
70	6	4	2	104, 103 and 102	950.912
72	2	1	1	115, 116 and 93	291.386
				121	209.287

It is important to report that for the case studies of the hypothetical network, real network scenario 1 and 2, the computational time for model execution was 0.15, 0.30 and 0.35 s, in the proper order.

6 Conclusions

In this research, an innovative depth-first search algorithm is presented for optimization in gravity pipe network layout. The decision of the model strategy always refers to what depth value it is possible to reduce on flat grounds or with topography unfavorable to gravity hydraulic conduction.

Results of this nature can help designers, leading to a more economical layout solution. It is possible to observe that, using the algorithm, the layout of any network can be led to a minimal cost solution.

The proposed methodology was validated in a hypothetical network presented in the specialized literature. As means of demonstrating the applicability of the model, the algorithm was utilized to a real network in two different scenarios. The research model found all possible solutions, discarding those that are not hydraulically feasible. To the best knowledge of the authors, there is no other exact comparative method for determining the optimal layout available in literature.

Table 4 Chosen paths to the real network confluent nodes from case study 2 –scenario 2

Confluent node	Possible path number	Number of not feasible paths due to loop formation	Number of feasible paths	Pipes of feasible path(s)	Feasible path weight (sum of weights of the edges)
15	6	2	4	11 and 10	471.496
				24	41.910
				11 and 55	407.694
				23 and 22	391.286
70	6	4	2	104, 103 and 102	950.912
72	2	1	1	115, 116 and 93	291.386
				121	209.287

The execution of the model in C# language took less than half a second for the solution in all case studies, which confirms the robustness of the proposed approach.

The innovation of this work is based on the analysis of the complete layout of any network, of any size, resulting in the lowest possible depths, avoiding unsolved hydraulic circuits and also with insignificant computational time. It should be noted that if the network has more than one outlet node, the layout is optimized and automatically divided into a number of basins that will have the same value as the number of outlet nodes.

In short, the strategy of the proposed algorithm is which directions of flow should be reversed in the pipes in order to obtain the minimum slopes and, consequently, the lowest depths in a gravity pipe network, thus preventing the designer to make the decisions by trials or manually.

References

- Afshar MH (2006) Application of a genetic algorithm to storm sewer network optimization. *Sci Iran* 13(3):234–244
- Boaventura Netto PO, Jurkiewicz S (2009) *Grafos: introdução e prática*. Blucher, São Paulo
- Cormen TH (2013) *Algorithms unlocked*. MIT Press, Cambridge
- Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) *Introduction to algorithm*. MIT Press, Cambridge
- Dajani JS (1971) Network evaluation of wastewater collection economics. PhD thesis. Northwestern University
- Dajani JS, Gemmel RS, Morlok EK (1972) Optimal design of urban wastewater collection networks. *J Sanit Eng Div Am Soc Civ Eng* 98(6):853–867
- Dasgupta S, Papadimitriou CH, Vazirani U (2008) *Algorithms*. McGraw-Hill Higher Education, New York
- Deininger RA (1966) Computer aided design of waste collection and treatment systems. Proc., 2nd American Water Resource Conf., Univ. of Chicago, Chicago
- Diogo AF, Graveto VM (2006) Optimal layout of sewer systems: a deterministic versus a stochastic model. *J Water Res Plan Man* 132(9):927–943
- Gameiro LFS (2003) Dimensionamento otimizado de redes de esgotos sanitários com a utilização de algoritmos genéticos. MSc dissertation. Mato Grosso do Sul Federal University
- Haghighi A, Bakhshpour AE (2012) Optimization of sewer networks using an adaptive genetic algorithm. *Water Resour Manag* 26:3441–3456
- Haith D (1966) Vertical alignment of sewers and drainage systems by dynamic programming. MSc dissertation. Massachusetts Institute of Technology
- Heaney JP, Pitt R, Field R (2000) Innovative urban wet-weather flow management systems. EPA/600/R-99/029, U.S. Environmental Protection Agency, Cincinnati, USA
- Holland ME (1966) Computer models of wastewater collection systems. PhD thesis. Harvard University, Cambridge
- Kulkarni VS, Khanna P (1985) Pumped wastewater collection systems optimization. *J Environ Eng Div (Am Soc Civ Eng)* 111(5):589–601
- Liebman JC (1967) A heuristic aid for the design of sewer networks. *J Sanit Eng Div Am Soc Civ Eng* 93(4):81–90
- Mai SW, Evans DJ (1984) A parallel algorithm for the enumeration of the spanning trees of a graph. *Parallel Comput* 1:275–286
- Mays LW, Yen BC (1975) Optimal design of branched sewer system. *Water Resour Res* 11(1):37–47
- Mocini R, Afshar MH (2017) Arc based ant colony optimization algorithm for optimal design of gravitational sewer networks. *Ain Shams Engineering J* 8(2):207–223. <https://doi.org/10.1016/j.asej.2016.03.003>
- Nzewi EU, Gray DD, Houck MH (1985) Optimal design program for gravity sanitary sewers. *Civ Eng Syst* 2: 132–141
- Pan TC, Kao JJ (2009) GA-QP model to optimize sewer system design. *J Environ Eng* 135(1):17–24
- Russell SJ, Norvig P (2010) *Artificial intelligence: a modern approach*, 3rd edn. Prentice Hall, Upper Saddle River, NJ
- Steele JC, Mahoney K, Karovic O, Mays LW (2016) Heuristic optimization model for the optimal layout and pipe design of sewer systems. *Water Resour Manag* 30:1605–1620
- Velon JP (1971) Sewer cost – estimation model: an application. MSc dissertation. Northwestern University

- Villas MV, Ferreira AGM, Leroy PG (1993) Estruturas de dados: conceitos e técnicas de implementação. Ed. Campus, Rio de Janeiro
- Walsh S, Brown LC (1973) Least cost method for sewer design. *J Environ Eng Div* 99(3):333–345
- Walters GA, Lohbeck T (1993) Optimal layout of tree networks using genetic algorithms. *Eng Optim* 22(1):27–48
- Walters GA, Smith DK (1995) Evolutionary design algorithm for optimal layout of tree networks. *Eng Optim* 24(4):261–281

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.