Operational Research (2026) 26:60
https://doi.org/10.1007/s12351-026-01054-z

ORIGINAL PAPER

Optimizing irrigation network system design through canal capacity analysis

Dimitra Alexiou[1] · Dimitra Vagiona[1] · Angelo Sifaleras[2]

Received: 14 December 2024 / Revised: 15 March 2026 / Accepted: 9 April 2026

Abstract

This study addresses the challenge of efficiently distributing water from a central source to distinct demand points through an irrigation canal system. In this network design, the demand locations serve as network nodes, interconnected by arcs that represent the canals to be built. The building expense for each canal is influenced by various factors, including land structure and topography. The rate of water flow along a canal path from the main source to a specific demand location is restricted by the minimum capacity of the canal along that route. The primary objective is to identify a subset of canals that can maximize water delivery per unit of time to each demand location while minimizing construction costs and satisfying constraints due to canal capacities. The proposed graph-theoretic approach is validated using a semi-real irrigation case study with an irregular, free-form, topology.

Keywords Optimization · Irrigation network · Bottleneck problem

---

✉ Dimitra Alexiou
dimitraalexiou@plandevel.auth.gr

Dimitra Vagiona
dimvag@plandevel.auth.gr

Angelo Sifaleras
sifalera@uom.gr

[1] Department of Spatial Planning and Development, Engineering School, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

[2] Department of Applied Informatics, School of Information Sciences, University of Macedonia, 156 Egnatia Str., 54636 Thessaloniki, Greece

1 Introduction

The supply of water to an irrigation system is closely intertwined with its impact on agricultural productivity. A crucial determinant of an irrigation system's efficiency lies in the configuration of its canal network, particularly in addressing the diverse water requirements across different locations. The capacity of a canal dictates the volume of water it can transport, and this capacity varies based on several factors, including the structure of the land, the topography (Johnson 2008), and the morphology of the canal. Recent works on irrigation network optimization include the studies (Gajghate and Mirajkar 2020; Gajghate et al. 2021).

In this context, our focus is on the examination of a stationary and continuous-flow irrigation system. Within this system, a comprehensive canal water distribution network has been meticulously established. Its core objective revolves around receiving water from a primary source and facilitating the transportation of the maximum attainable volume of water per unit of time to different geographical points. This endeavor is undertaken with a dual perspective: optimize water distribution and simultaneously minimize construction costs, while meticulously accounting for the inherent limitations posed by canal capacity. To approach this problem systematically, we employ a graph theoretical approach (Griffin 2023), which allows us to represent the irrigation canal network, denoted as N, as a weighted, connected, Directed Acyclic Graph (DAG) denoted as $G = (V, E, C)$. Within this graph representation, individual nodes in $V = (v_0, v_1, ..., v_n)$ signify discrete water demand locations, which can include inter-farm distributors responsible for delivering water to farms or specific crop locations. The set of arcs, denoted $E = [(v_i, v_j], v_i, v_j \in V)$, represents the potential canals to be constructed. The elements of $C = [c(v_i, v_j)]$ correspond to the capacities of the canals $(v_i, v_j) \in E$. In cases where there are preexisting canals or pipelines, it is assumed that the construction cost associated with the respective arcs is zero.

A simple path P_{ij} within the context of the set N can be defined as an ordered sequence represented by $P_{ij} = (v_i, v_{i+1}, \ldots, v_{i+k} = v_j)$. This path must adhere to specific criteria, where v_i and v_j belong to the set of edges E, and i ranges from 1 to $k - 1$. Importantly, this sequence requires that each node appear only once within it. For our analysis, we exclusively consider simple paths. For any given pair of nodes (v_i, v_j) within the set of edges E, v_i is designated as the predecessor of v_j, while v_j assumes the role of the successor of v_i. Consequently, the predecessors and successors of a specific node v_i within the set V are fully represented by the elements contained within the sets denoted $\Gamma^-(v_i)$ and $\Gamma^+(v_i)$, respectively.

This work extends previous work by Alexiou and Tsouros (2017), where the proposed algorithm was initially applied on a network with a highly regular topology and a grid-like structure. In addition, the connections of the previous case study were uniform and predictable. In this new work, we validate the proposed algorithm using a semi-real irrigation case study with an irregular free-form topology, regarding the irrigation canal network in the Axios river Delta, located in central Macedonia, Greece. Thus, the connection pattern is now varied and non-uniform. Such topology differences usually have an impact on the solution process, since bottlenecks might be more distributed in the grid network due to the more combinatorial but structured search space. On the other hand, irregular networks might have fewer, but more complex, paths to analyze.

Overall, this work has three subsequent contributions. First, it proposes a problem formulation that jointly optimizes bottleneck flow and construction cost for single-path canal systems. Second, it proposes a deterministic algorithm that avoids the stochastic overhead of other metaheuristic approaches used in the literature. Third, the validation is carried out using a semi-real irrigation case study with irregular topology, representative of real-world irrigation infrastructure.

The remainder of this paper is organized as follows. Sections 2 and 3 present the proposed methodology and a semi-real irrigation case study, respectively. Finally, the conclusions and future work are discussed in Sect. 4.

2 Algorithmic approach for the bottleneck problem

To formulate a water irrigation network that aligns with the described problem, a systematic algorithmic approach is used to determine the maximum water conveyance achievable to each demand location using a single canal route (Kaibel and Peinhardt 2006; Renjie et al. 2012; Punnen and Ruonan 2009). The highest volume of water that can be transported along a canal path P_{ij} connecting location v_i to v_j is determined by the smallest canal capacity contained in P_{ij}. Consequently, the central challenge revolves around the identification of bottleneck scenarios, which require the determination of the utmost flow possible from v_i to v_j via a P_{ij} pathway.

2.1 Notation and definitions

Table 1 summarizes the notation used throughout the algorithmic formulation.

2.2 Algorithmic approach for bottleneck trees

The proposed solution methodology proceeds in two sequential phases. Phase 1 determines the maximum bottleneck flow from the water source to every demand location. Phase 2 constructs a minimum-cost bottleneck tree that realizes these maximum flows using the fewest and cheapest canals.

The goal of the first phase is to compute, for each demand node $v_i \in V$, the value $L(v_i)$, representing the maximum volume of water per unit time that can be delivered from the source v_0 to v_i through any single canal path. The fundamental observation is that the flow achievable along any path P_{0i} from v_0 to v_i is constrained by the canal with the smallest capacity on that path (i.e., the bottleneck canal). The algorithm employs a label-setting strategy inspired by Dijkstra's shortest-path algorithm, adapted to maximize the minimum arc capacity rather than minimize cumulative distance.

Table 1 Summary of notation and symbols

Symbol	Description
N	The irrigation canal network
$G = (V, E, C)$	Weighted, connected, directed acyclic graph (DAG) representing N
$V = \{v_0, v_1, \ldots, v_n\}$	Set of nodes (demand locations); v_0 denotes the water source
n	Total number of demand locations (excluding the source)
$E = \{(v_i, v_j) \mid v_i, v_j \in V\}$	Set of directed arcs representing potential canals
$C = [c(v_i, v_j)]$	Capacity matrix; $c(v_i, v_j)$ is the maximum water flow rate of canal (v_i, v_j)
P_{ij}	A simple (non-repeating) path from node v_i to node v_j
$\Gamma^-(v_i)$	Set of predecessor nodes of v_i (nodes with arcs directed toward v_i)
$\Gamma^+(v_i)$	Set of successor nodes of v_i (nodes reachable from v_i via a single arc)
$L(v_i)$	Label of node v_i: the maximum bottleneck flow from source v_0 to v_i
Q	Set of nodes whose labels are currently temporary
p	The most recently processed (permanently labeled) node
P_d^k	The k-th alternative bottleneck path (B-path) to demand location d
z_d	Number of alternative B-paths to demand location d
B	Set of arcs currently included in the bottleneck tree (BT)
$G_{v_i}^k$	Set of new arcs added to B when selecting the k-th B-path to v_i; defined as $G_{v_i}^k = P_{v_i}^k \setminus (B \cap P_{v_i}^k)$
$\mathrm{Cost}(G_{v_i}^k)$	Total construction cost of the arc set $G_{v_i}^k$
$S = (s_1, s_2, \ldots, s_k)$	Stack data structure storing nodes during DFS-based path enumeration
BT	The bottleneck tree: the optimal subset of canals connecting all demand locations to the source

With the bottleneck flow values $L(v_i)$ determined, the second phase identifies the set of canals to construct (i.e., forming a directed tree rooted at v_0) that achieves these flows at the lowest total construction cost. Demand locations are processed sequentially, from the nearest to the source to the most distant.

The incremental, greedy nature of this strategy ensures that canals shared across Bottleneck paths (B-paths) to multiple destinations are constructed only once, substantially reducing redundant expenditure. If $P_{v_i}^q \subseteq P_{v_{i-1}}^q$, then $G_{v_i}^q = \emptyset$ and the tree remains unaltered, since all required canals were already included in a previous iteration.

Concerning the bottleneck problem, an algorithmic methodology has been devised, utilizing a pertinent adaptation of Dijkstra's algorithm (Christofides 1975; Goodrich et al. 2013). More precisely, a label denoted as $L(v_i)$ is linked to each node $v_i \in V$. This label takes on a numerical value and, at a specific point in the procedure, is designated as either permanent or temporary. A label denoted as permanent, symbolized by $L(v_i)$, signifies the maximum flow achievable through a single canal path from the source node v_0 to the destination node v_i. The set Q comprises nodes from V with labels that are currently designated as temporary within the ongoing algorithmic process. The algorithm concludes its execution once all nodes in V have their labels transitioned to permanent status, specifically, i.e., $Q = \emptyset$. The variable p represents the most recently processed node, the label of which has been permanently set. The pseudocode of the solution procedure is provided in Algorithm 1.

Algorithm 1 The algorithmic procedure

```
procedure SOLUTION_ALGORITHM($V, Q, P, C, L$)
    Set $Q \leftarrow V - v_0$, $L(v_0) \leftarrow \infty$, $p \leftarrow v_0$
    $\forall v \in Q$, set $L(v) \leftarrow 0$
    while $Q \neq \emptyset$ do
        for every $v \in (\Gamma^+(p) \cap Q) \neq \emptyset, p \in (V - Q)$ do
            $L(v) \leftarrow max\,\{L(v), min\{L(p), c(p, v)\}\}$
            $L(x) \leftarrow max\{L(y), y \in Q\}$
            $p \leftarrow x$ and $Q \leftarrow Q - x$
        end for
    end while
    return $P, L$
end procedure
```

A B-path, denoted as $P_k = (v_0 = i_1, \ldots, i_q = v_k = T)$, leading from the source node s to the destination location T, must satisfy the following conditions: $L(i_j) \geq L(T)$ and $c(i_{j-1}) \geq L(T)$ $\forall j = q, q-1, \ldots, 2$. To select the most suitable B-path for each demand location $v_i \in V$, where $i = 1, 2, \ldots, n$, we employ a Depth-First Search (DFS)-type algorithm (Cormen et al. 2022). The search is conducted sequentially for each demand location, starting from the closest nodes to the source and progressing toward the more distant ones. The collection of derived B-paths forms a bottleneck tree, denoted as BT.

The primary objective is to identify the BT tree with the lowest total cost for its arcs. During a specific step in the execution of the algorithm, let $P_{v_i}^k = [(v_0, p_1), (p_1, p_2), \ldots, (p_{q-1}, p_q = v_i)]$ represent the k^{th} alternative B-path, where $k = 1, 2, \ldots, z_{v_i}$, signifying the number of alternative B-paths leading to location v_i. Additionally, $\mathrm{B} = \{(p_{j1}, p_{j2}), (p_{j3}, p_{j4}), \ldots\}$ is the set containing all arcs included in the bottleneck tree at a previous phase of the algorithm. Once a $P_{v_i}^k$ has been identified, B is augmented by a set of arcs $G_{v_i}^k$, where $G_{v_i}^k = P_{v_i}^k - (\mathrm{B} \cap P_{v_i}^k)$. The selection of $G_{v_i}^k$, to expand B, involves finding an arc set $G_{v_i}^k$ with the minimum total cost for its arcs, expressed as:

$$Cost(G_{v_i}^k) = \min\{Cost(G_{v_i}^k) \mid k = 1, 2, \ldots, z_{v_i}\} \tag{1}$$

Algorithm 2 provides the pseudocode of the Bottleneck Tree algorithmic procedure.

Algorithm 2 The bottleneck tree algorithmic procedure

```
procedure BottleneckTree_Solution_Algorithm($n, \Gamma^-(i)$)
    $d \leftarrow 1$
    while $d \leq n$ do
        $i \leftarrow d,\ k \leftarrow 1,\ z \leftarrow 0,\ A^-(i) \leftarrow \Gamma^-(i),\ S_k \leftarrow d$
        $TerminationFlag = false$
        while $TerminationFlag = false$ do
            while $A^-(i) \neq \emptyset$ do
                Select $j \in A^-(i)$ (j is a predecessor of i)
                if $L(j) \geq L(d)$ and $c(j,i) \geq L(d)$ then
                    $k \leftarrow k+1,\ S_k \leftarrow S_k + j,\ A^-(i) \leftarrow A^-(i) - j$
                    if $j = s$ then
                        $z \leftarrow z+1,\ P_d^z \leftarrow S_k,\ i \leftarrow j$
                        Exit loop ($A^-(i) \neq \emptyset$)
                    end if
                    $i \leftarrow j$
                end if
            end while
            $A^- \leftarrow \Gamma^-(i)$
            $k \leftarrow k-1$
            if $k < 0$ then
                $TerminationFlag = true$
            else
                $i \leftarrow j$
            end if
        end while
        $k \leftarrow 1$
        while $k \leq z$ do
            $G_d^k \leftarrow P_d^k - (\mathbf{B} \cap P_d^k)$
            $Cost(G_d^k) \leftarrow \sum_{(y_{j+1}, y_j) \in G_d^k} c(y_{j+1}, y_j)$
            $k \leftarrow k+1$
        end while
        Select $G_d^q$ based on Equation 1
        Update Bottleneck Tree: $B \leftarrow B + G_d^k$
        $d \leftarrow d+1$
    end while
    return $B$
end procedure
```

Let $P_{v_i}^q$ represent the B-path corresponding to $G_{v_i}^q$. If $P_{v_i}^q$ is a subset of $P_{v_{i-1}}^q$, then $G_{v_i}^q = \emptyset$. In such cases, the Bottleneck Tree (BT) remains unchanged, since all the canals that serve these paths were already included in the BT during a previous stage.

The rationale presented in the preceding paragraphs is an integral part of the Bottleneck Tree algorithmic procedure applied to each demand location $d \in V$. The search is conducted sequentially, starting from the smallest to the largest demand location *d*. Additionally, the Depth-First Search (DFS)-based algorithm systematically generates all alternative B-paths $P_d^k, k = 1, 2, \ldots, z_d$ for each location. This process progressively augments the Bottleneck Tree BT with the canals contained in $G_{v_i}^q$. The elements of a B-path are generated and stored within a stack denoted as $S = (d = s_1, s_2, \ldots, s_k = v_0)$. Furthermore, the values of $L(i), i = 1, 2, \ldots, n$, for a bottleneck path P_d^k have already been determined by the algorithm. For each demand location *d*, the application of the algorithm triggers the formation of a DFS tree.

3 Case study: irrigation canal network in the Axios River Delta, Central Macedonia, Greece

To demonstrate the practical applicability of the proposed methodology, we consider a semi-real irrigation network inspired by the agricultural landscape of the Axios (Vardar) River delta and its surrounding plain in Central Macedonia, Greece. This region, situated west of Thessaloniki, constitutes one of the most productive agricultural zones in southeastern Europe. The area encompasses approximately 22,000 hectares of intensively cultivated land, supporting rice paddies along the lower delta, cotton and maize fields on the central plain, and fruit orchards (primarily peach and kiwi) on slightly elevated terrain to the west. The irrigation infrastructure is managed by the Greek Regulatory Authority for Energy, Waste & Water (RAAEY) which operate an extensive canal system sourcing water from the Axios River and, in part, from the Aliakmonas River via the interconnecting canal of Aliakmonas–Axios.

The network topology presented herein reflects the irregular, non-uniform connectivity patterns that arise in practice due to: (i) terrain variability between the flat alluvial delta and the gently elevated western hinterland; (ii) fragmented land parcel ownership requiring canal routing along cadastral boundaries and existing road infrastructure; (iii) the presence of natural obstacles such as drainage channels, seasonal waterways, and ecologically protected wetland zones (the Axios Delta is a Ramsar and Natura 2000 site, imposing routing constraints on new canal construction); and (iv) heterogeneous soil conditions ranging from heavy alluvial clay in the delta to sandy loam on higher ground.

The network N comprises $n = 19$ demand locations (nodes v_1 through v_{19}) and a single source node v_0, representing the main water intake from the Axios River at the delta's head, as shown in Fig. 1. The directed acyclic graph $G = (V, E, C)$ contains $|V| = 20$ nodes and $|E| = 38$ directed arcs. Each arc represents a potential canal whose construction is subject to site-specific costs determined by distance, terrain, soil type, required earthworks, and the need for ancillary structures (culverts, siphons, road crossings).

Canal capacities $c(v_i, v_j)$ are expressed in m^3/min and have been assigned within the range $[10, 40]$ m^3/min, consistent with flow rates typical of secondary and tertiary irrigation canals in the region. To simplify the analysis and without loss of generality, the construction cost of each canal is assumed to be numerically equal to its capacity.

The numeric values enclosed within squares adjacent to each node v_i signify the computed value $L(v_i)$ achieved by applying the solution algorithm. This specific metric denotes the maximum volume of flow that a single canal path, originating from the source node v_0 and ending at the corresponding demand location, can effectively transport.

Consider a path denoted as $P_k = \{v_0, v_1, \ldots, v_k\}$ that leads to the values $L(v_k)$. We refer to P_k as a *bottleneck path* from v_0 to v_k. For instance, when $v_k = 13$, a value of $L(13) = 20$ can be achieved through three distinct alternative bottleneck paths: $P_{13}^1 = [0, 2, 5, 9, 8, 13]$, $P_{13}^2 = [0, 1, 2, 5, 9, 8, 13]$, and $P_{13}^3 = [0, 1, 4, 10, 9, 8, 13]$. The accompanying Table 2 provides the count of alternative bottleneck paths associated with each node within the analyzed network.

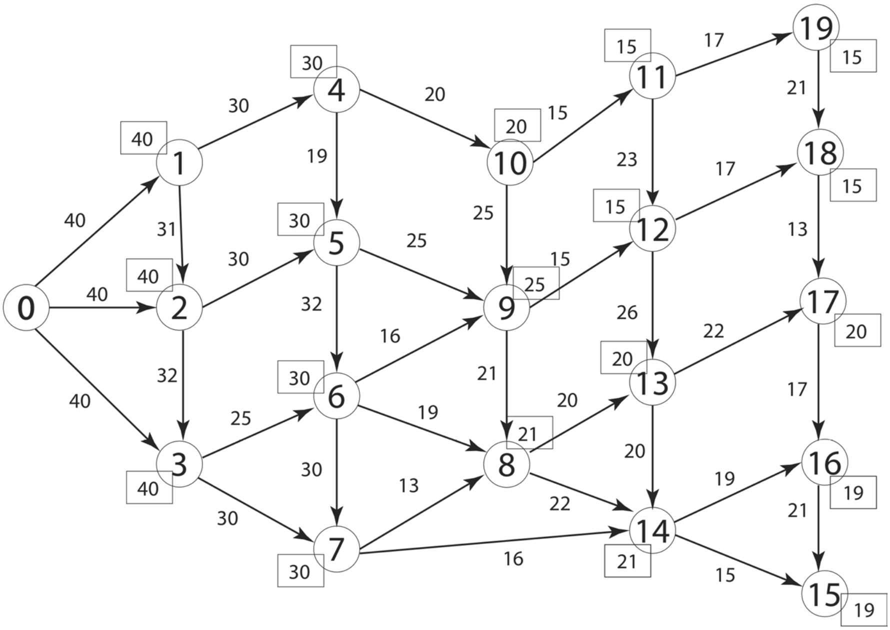


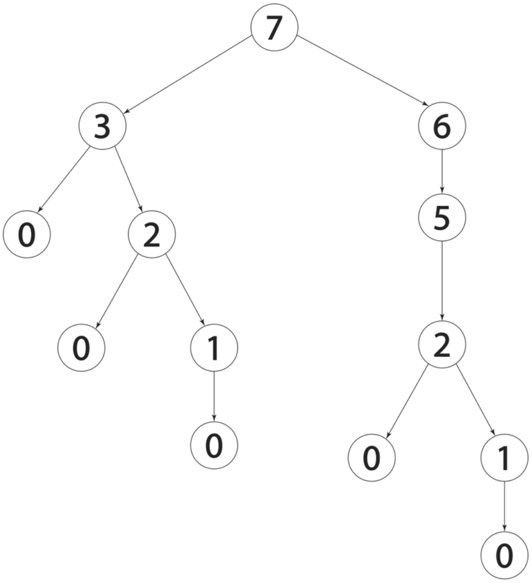
Fig. 1 A network of a semi-real irrigation case study with an irregular topology

Table 2 Count of alternative bottleneck paths for each node

Nodes	# Alternative paths	Nodes	# Alternative paths
0	–	10	1
1	1	11	1
2	1	12	10
3	1	13	3
4	1	14	2
5	2	15	17
6	2	16	17
7	5	17	3
8	2	18	11
9	2	19	1

Evidently, the presence of multiple alternative bottleneck paths allows for the creation of numerous irrigation design systems. The primary aim in this context is to identify the specific system that minimizes the overall construction cost of the canals while ensuring that the quantity of water delivered to each demand location v_i matches $L(v_i)$. To simplify the analysis and without loss of generality, we assume that the construction cost of each canal is numerically equivalent to its capacity. As previously noted, the DFS-based algorithmic procedure has the capacity to produce alternative canals within the bottleneck tree. For instance, Fig. 2 provides a visual representation of the alternative B-paths for demand location $v = 7$.

Fig. 2 Alternative B-paths for demand location $v = 7$



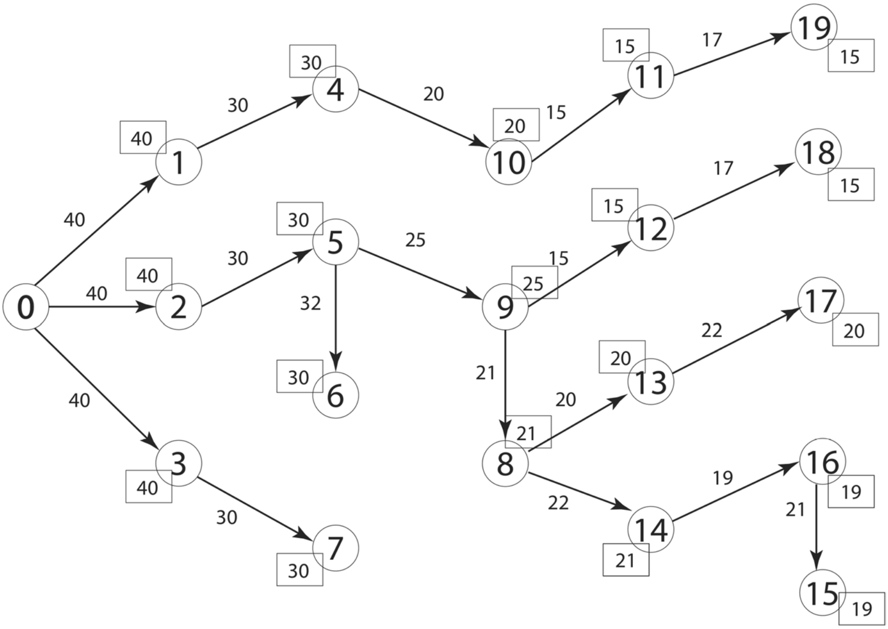


Fig. 3 The cost-efficient bottleneck tree

Figure 3 presents the minimum-cost bottleneck tree derived using the applied algorithmic approach for the analyzed network. The arcs contained in the bottleneck tree in Fig. 3 represent the canals to be built so that the cost of supplying water to the demand locations is minimized.

Table 3 Count of alternative bottleneck paths for each node

Nodes	Max flow	Nodes	Max flow
0	∞	10	20
1	40	11	15
2	40	12	15
3	40	13	20
4	30	14	21
5	30	15	19
6	30	16	19
7	30	17	20
8	21	18	15
9	25	19	15

Table 3 provides a summary of the maximum water flow capacity achievable for each demand location within the optimal irrigation network configuration.

4 Conclusions and future work

The approach presented in this study represents a heuristic procedure capable of efficiently addressing scenarios involving large-scale networks, specifically those with numerous demand locations, as the time complexity of the bottleneck tree construction algorithm is a low-degree polynomial function denoted as $O(n^2)$, where n signifies the count of demand locations. Furthermore, the proposed method can be easily applied to pipeline drainage systems, as comprehensive data on pipe cost and capacity are available on the market.[1]

Real-world irrigation systems are influenced by a myriad of factors, which include variables such as water flow velocity and precise water requirements at specific locations. In select cases, the framework of the presented method can be extended to address these complex factors. In essence, a bottleneck tree within an irrigation canal network represents the minimal set of canals needed to connect all demand locations to the water source. Naturally, numerous bottleneck tree configurations can be devised. However, this study applies a methodology to determine the bottleneck tree that incurs the lowest feasible cost of canal construction. This approach ensures that the canals within an irrigation or pipeline drainage system are strategically arranged to maximize the supply of water to the demand locations during each time unit.

The proposed approach could be possibly integrated with GIS and hydraulic simulation tools (e.g., EPANET, WaterGEMS). In addition, a computational analysis and a sensitivity analysis using other cost functions would be very interesting as research extensions of this study. Finally, another interesting future research work would be an experimental comparison between the proposed algorithmic method and other solution approaches found in the literature, using other real-world datasets.

Funding Open access funding provided by HEAL-Link Greece.

---

[1] http://www.alibaba.com/showroom/irrigation-pipe-price.html.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Alexiou D, Tsouros C (2017) Design of an irrigation network system in terms of canal capacity using graph theory. J Irrig Drain Eng 143(6):06017002

Christofides N (1975) Graph theory, an algorithmic approach. Academic Press, Cambridge

Cormen TH, Leiserson CE, Rivest RL, Stein C (2022) Introduction to algorithms, 4th edn. MIT press, Cambridge

Gajghate PW, Mirajkar A (2020) Irrigation pipe network planning at tertiary level: an Indian case study. KSCE J Civ Eng 24(1):322–335

Gajghate PW, Mirajkar A, Shaikh U, Bokde ND, Yaseen ZM (2021) Optimization of layout and pipe sizes for irrigation pipe distribution network using Steiner point concept. Math Probl Eng 2021(1):6657459

Goodrich MT, Tamassia R, Goldwasser MH (2013) Data structures and algorithms in python. Wiley, Hoboken

Griffin CH (2023) Applied graph theory: an introduction with graph optimization and algebraic graph theory. World Scientific, Singapore

Johnson LE (2008) GIS for water-supply and irrigation systems Geographic information systems in water resources engineering. CRC Press, Boca Raton, pp 137–162

Kaibel V, Peinhardt MAF (2006) On the bottleneck shortest path problem (Tech. Rep. No. 06-22). Konrad-Zuse-Zentrum für Informationstechnik

Punnen AP, Ruonan Z (2009) Bottleneck flows in unit capacity networks. Inf Process Lett 109(6):334–338

Renjie S, Qinggang J, Yanyan L, Jing Z (2012) Identify the bottleneck of water network by using graph theory. Adv Mater Res 433–440:4794–4797