OPTIMAL ROUTING OF IRRIGATION CANAL PATHS USING A DIGITAL ELEVATION MODEL

Sai Chaitanya Reddy Ponnathota (1), Rajan K S (1)

[1] Lab for Spatial Informatics, International Institute of Information Technology, Gachibowli, Hyderabad, Telangana, 500032, India
Email: ponnathotasaichaitanya.reddy@research.iiit.ac.in; rajan@iiit.ac.in

KEY WORDS: Canals, DEM, Flow-routing algorithms, GIS, Terrain analysis

ABSTRACT: In developing the infrastructure facilities such as irrigation canals and road networks, topography acts as a significant enabler or constraint. Contour maps and low resolution DEMs have been used by Irrigation engineers and planners to assess the canal routing options, which is time consuming and requires repeated evaluations of the potential paths. So, there is a need to develop robust path planning algorithms, including least cost routing, that takes the topographic and engineering constraints while providing potential canal routing paths. Some recent works have attempted to develop algorithms on synthetic data sets but have not been scaled up on high-resolution data sets, limiting their practical use.
This work develops a generic algorithm to determine the least-cost flow path between two geo-locations, given the grid-based Digital Elevation Models (DEMs) and a unit cost of construction per length. From the numerous paths that are possible between the two points in any given topography, a distinct least-cost path is identified. The proposed approach is evaluated by computing canal routing paths over publicly available real-world datasets across two different resolutions of 1Km and 90-meter from different sources for Indian terrains. This is then compared against a global reference real-world dataset, Digital Chart of the World (DCW) at 1Km at a grid cell level. The results across multiple stretches shows an average accuracy of 82.09% as measured based on the path overlap against the DCW dataset, proving that the proposed algorithm can be useful in practice.

1. INTRODUCTION

The construction of canals and roads is one of the most commonly utilized utility infrastructure services in the world. Topography often acts as a significant constraint when planning potential routes for canals and roads. While traditionally contour based path planning was done, in the last few decades the availability of Digital Elevation Models has helped to understand the terrain features better and is well suited for computational models. A gridded Digital Elevation Model (DEM) is a common format for digital representation of terrain elevation. It is widely used to extract hydrological and geomorphological information for numerous purposes, such as flow direction, flow accumulation, and stream network delineation, etc. (David G. Tarboton, 1998, Erwin Weinmann, 1979, John Wilson et al., 2000, Tribe, 1992, Watson et al., 2011, Weihua Zhang et al., 1994)
There are two approaches for evaluating the flow path from the DEM: the single-flow direction method and the multiple-flow direction method. In the single-flow direction method, each cell only drains to one neighboring cell based on the principle of steepest slope. The D8 method, which uses the descent direction as a flow direction for a cell, is the most widely adopted single-flow direction method. In the multiple-flow direction method, each cell can flow to more than one neighboring cell that is at a lower elevation than the current cell. While the most common application of a DEM may be drainage and watershed modelling, the DEMs can be used for many other applications like viewshed analysis, cut and fill volume estimation

and other engineering works. One such engineering application of significance is the routing of canal to move the water for irrigation and other purposes. Irrigation canals are not just routing problems but also need to consider many other factors like reachability, land use, minor changes in elevation, irrespective of whether it is a gravitational flow based canal or lift-irrigation system with intermediate lifts over the path.
The manual task of drawing a path and visualizing its profile each time and then to determine the canal cost construction is time taking and requires repeated evaluations which is susceptible to errors. And moreover, due to the lack of any clear method, irrigation engineers and planners tend to adopt the conventional approach. Although a few studies addressed the least-cost path for canals, there are several challenges with the existing solutions that limit its applicability like (i) the existing algorithms are sensitive to the data, i.e., specific to the data, and not modeled to perform on real-world data sets; and (ii) are challenging to scale up to perform on high resolutions of data such as 90-meters, 30-meters, 10-meters, 5-meters.
Hence, in this paper, we develop a computational Gravitational Flow algorithm that evaluates the canal paths between a set of two geo-locations and shall report the least-cost canal path for construction. Between any two coordinates on the terrain, multiple trajectories are possible. The selected route should be a reasonable approximation of the one with the least cost and follows the principle of gravitational force. Here, we define a cost function based on the grid elevation and the distance to move from one cell to its neighboring cell.
We run the proposed algorithm on various real-world terrains with varying inputs belonging to different resolutions to test the robustness and performance of the algorithm. An experimental evaluation of the proposed algorithm is carried out for its correctness compared to real-world DCW (Digital Chart of the World) data, which showed an average accuracy of 82.09%. Our results also prove that the algorithm is applicable independent of the resolution and scales well as it is linearithmically based on the resolution of the data.
The rest of this paper presents - an overview of related work in Section 2; the proposed solution to the Gravitational flow model in Section 3; data used and results in Section 4; and an experimental evaluation of the results in Section 5. Finally, Section 6 discusses the outcomes of this research work.

2. RELATED WORK

In this section, we provide a detailed review of the current state of research related to flow routing algorithms.

Single flow-direction algorithms: The earliest and most straightforward method for specifying flow directions is to assign flow from each pixel to one of its eight neighbors, either adjacent or diagonal. The steepest descent algorithm is one of the most frequently used algorithms. After the calculation of the gradients between the central cell and all its neighboring cells, all flow is directed into the neighboring cell corresponding to the highest gradient. This method, designated D8 (eight flow directions), was introduced by O'Callaghan and Mark (O’Callaghan et al. 1984). Fairfield and Leymarie (John Fairfield et al., 1991) modified this algorithm to include a stochastic, quasi-random component. Gardner (Day et al., 1990) and Drayton calculated the aspect of direction using a surface fitting procedure, and the neighboring cell is selected based on the closest direction of the receiving cell. Lea (Lea et al., 1992). proposed an aspect-driven routing algorithm, whereby flow is moving kinematically along the aspect direction from the center of the source cell until it reaches a cell perimeter point. Once at the perimeter, flow is transferred to the coincident perimeter of the receiving cell. From there, it is routed to one of the other edges of the receiving cell, implying that the flow from a single cell generally follows a unique path to the outlet. The contributing area for a given grid cell can then be calculated as the number of flow lines passing through that cell multiplied by the grid-cell area. Scoging (Scoging, 1992) used the gradients of the four grid cell borders to calculate the resultant outflow direction. The cardinal neighbor corresponding as close as possible to this direction is selected as being the receiving cell.

Flow-routing algorithms: Earlier, network algorithms were adapted to solve the raster data structure type of problems. Douglas (Douglas, 1984) has proposed an algorithm for least-cost paths which is broken down into the computation of an accumulated cost surface integrated about a destination, and the generation of slope lines taking into consideration isotropic values. Lee (Lee et al., 1998) then proposed the least-cost paths by integrating viewshed information computed from digital elevation models. These algorithms are limited to situations where the cost of passage is the same for all movement directions. Walter et al., (2000) proposed a solution considering a function relating slope, distance, and cost. The algorithm is based on the accumulated cost. The run time complexity of the algorithm is $O(kn^2)$, where k is a number of repeated iterations to compute the result and is proportional to the path complexity, and n is the number of cells of the terrain. The evaluation of the algorithm is conducted on a grid of 60 x 70 nodes and artificial data sets. Because of this limited usage of the existing algorithm, we try to provide an optimal solution to determine the least-cost route model that is better in terms of time complexity, is independent of the resolution, and applicable to real-world data sets.

Canal Routing: Canal path across diverse areas depends on various factors, such as land structure and topography (Johnson 2008), as well as its morphology. Criteria for Design of Lined Canals and Guidance for Selection of Type of Lining (Bureau of Indian Standards, 2000) discusses the design of lining canals as an important feature of irrigation projects as it improves the flow characteristics.

Grid vs graph based routing: Dijkstra's algorithm (Michael et al., 1987) is to find the shortest path between nodes in a graph. Cost of an edge in Dijkstra's algorithm is one of the input parameters. However, for canal routing, edge cost won't be one of the input parameters. Elevation values are the input for our problem. As water flow is gravitational, it is necessary to make sure that the resultant canal path follows the gravitational flow. So, Dijkstra's algorithm cannot be applied directly as flow direction needs to be evaluated based on the elevation values to solve the canal routing problem.

3. GRAVITATIONAL FLOW MODEL ALGORITHM

According to the definition of gravitational flow, the flow of water is under the force of gravity. In simple terms, we define the flow to be always downstream. Consider the example shown in Figure 1; it is a hillshade which is a 3D representation of a terrain surface. As the gravitational flow is always downstream, the flow from current center cell, to its neighboring cell, is possible only when the elevation of the center cell's elevation is greater than or equal to its neighboring cell. Based on this analysis, we define the algorithm. In a typical graph data structure, the set of nodes and the edges are the inputs. In a grid data structure, each cell is a node. The edges of these nodes are its neighbouring cells. In total, for a node, only 8 edges are possible. Our solution is to find out the least cost path given a grid data structure, a start coordinate, an end coordinate and cost of construction per unit length. The edge weight for the center cell to its neighbouring cell would be the cost of construction from the center cell to the center of its neighboring cell considering the distance.
Multiple graph algorithms are currently in use. We will describe the type of algorithm that will be perfect for our use case .The Breadth First Search(BFS) and Depth First Search(DFS) algorithms are applicable when the weights are equal for all edges. For this problem, the edge cost isn't equal for all nodes. Because of this edge cost difference in the grid data structure, the

BFS and DFS algorithms can't be applied to the grid data structure. Dijkstra's algorithm (Michael et al., 1987) is used to find the shortest path between nodes in a graph. As Dijkstra's algorithm suits our approach, we will be implementing this approach in this problem. By using Fibonacci heap min-priority queue, we can further optimize the time complexity of the algorithm. After choosing the best fit algorithm, we need to modify the algorithm to suit the problem, as the data would be a grid data structure and not a graph data structure which depicts the cost from one node to another.

First, we set the cost of source to zero and for the remaining nodes to infinity. Keep track of all the nodes that are being visited. At the Initial step, all the nodes would be unvisited. Create a priority queue that will be used to traverse the nodes whose distance is the shortest from the current set of vertices in the queue and populate this queue with the start node. For the top element in the priority queue, mark the node as visited and calculate the cost from this center cell node to its 8 neighbouring nodes. A neighbouring node will be added to the queue when the path isn't deviating from the destination coordinate. The cost can be calculated by using the distance and cost of construction from the center cell to its neighbouring cell. If the cost from start coordinate to the current cell's neighbouring coordinate is less than the current cost, update the cost to the calculated cost from current cell to this neighbouring cell in addition to the cost from start coordinate to the center cell's coordinate and update the priority queue for the neighbouring node of this center cell's node such that the key value for the neighbouring cell's node is of high priority. Iterations will be continued until the priority queue is empty. And finally, we return the cost and path for the unconstrained Gravitational flow model.

As mentioned in the algorithm, the path traversed shouldn't be deviated from the destination coordinate. By definition the canal stream movement will always be forward. If the path deviates, then the result won't be a canal path. Several boundaries have been set up to determine the definition of deviated canal paths. The boundaries are set as follows:

- The path traversal from start coordinate should tend towards the end coordinate
- The path traversal shouldn't violate directional constraints
- The path length covered shouldn't exceed the threshold value

Algorithm 1 Gravitational Flow Path algorithm

```
procedure LEAST COST PATH(Terrain, Costofconstruction)
    Cost[src_node] = 0; Cost[rest_nodes] = ∞;
    create priority_queue<customData> Q; add start node to the queue;
    while Q is not empty do
        curr_center_node ← Get the minimum element from the Queue
        for each neighbour of curr_center_node do
            if neighbour node is in boundary then
                Calculate the cost from curr_center_node to this neighbouring node
                if Cost[curr_center_node]+distance from center cell to its neighboring cell < Cost[this_neighbour_node] then
                    Cost[this_neighbour_node] ← Cost[curr_center_node] + distance from center cell to its neighboring cell
                    Update the priority_queue Q
                end if
            end if
        end for
    end while
    Return the least cost path from start coordinate to end coordinate
end procedure
```

Algorithm 1: The Gravitational Flow Path Algorithm

100	99	101	97	100
100	98	102	98	99
99	104	97	96	101
98	95	101	95	94
97	96	96	93	92

Table 1: The table describes the elevation values

0	1	∞	∞	∞
1	$\sqrt{2}$	∞	∞	∞
∞	∞	∞	∞	∞
∞	∞	∞	∞	∞
∞	∞	∞	∞	∞

Table 2: Cost of construction after 3 iterations

0	1	∞	∞	∞
1	$\sqrt{2}$	∞	∞	∞
2	∞	$2\sqrt{2}$	∞	∞
∞	∞	∞	∞	∞
∞	∞	∞	∞	∞

Table 3: Cost of construction after 8 iterations

0 → ↓ ↘	1 ↓	∞	∞	∞
1 → ↓	$\sqrt{2}$ ↘	∞	∞	∞
2 ↓↘	∞	$2\sqrt{2}$ → ↙↘	$1+2\sqrt{2}$ ↓ ↘	∞
3 → ↓	$2+\sqrt{2}$ ↓	∞	$3\sqrt{2}$ → ↘	$1+3\sqrt{2}$ ↓
4 →	$3+\sqrt{2}$ →	$2+2\sqrt{2}$ →	$1+3\sqrt{2}$ →	$4\sqrt{2}$

Table 4: Final result

Algorithm demonstration

Using an example, we demonstrate how the algorithm works. Table 1 describes the elevation values. Table 2 represents the flow path from top left cell (0,0) to its neighbours (0,1), (1,1) and (1,0). As the elevation at (0,0) is either greater than or equal to that of its neighbours, the flow path is possible. The table 3 represents the flow path from cell (1,1), (0,1) and (0,1) to its neighbours (0,2), (1,2), (2,2), (2,1) and (2,0). As the elevation at (0,1) is less than the elevation of (0,2), no gravitational flow is possible from (0,1) to (0,2). Similarly, (1,1) is also a neighbor of (0,2). As elevation at (1,1) is less than the elevation of (0,2), no gravitational flow is possible in this case as well. Similarly for other cells as well. The table 4 describes the cost of construction for canals from top left (0,0) to bottom right (4,4) along with the water flow direction. The arrow indicates the flow directions that are possible. Consider the cell (3,1) highlighted in yellow, there are three possible paths to it from (2,0), (2,2) and (3,0). Among all these, the least-cost path is chosen and so, the canal path to (3,1) is from (2,0). Considering all the different possibilities, the flow direction from (0,0) to (4,4) is (0,0), (1,1), (2,2), (3,3) and (4,4).

Considering all the different functions that are defined, in the worst-case when the algorithm has to traverse all the cells, the gravitational flow model runs in O(n logn), where n is the total number of cells in the terrain. The Gravitational Flow Model algorithm is better than Walter et al., (2000) solution in terms of time complexity.

5. DATA AND RESULTS

In this section, we first describe the data which is used for terrain analysis. Then, we describe the models using a variety of metrics, followed by analyses of data attributes. Later we apply the algorithms on these data and discuss the results.

Datasets

The data set we used is of 1KM resolution belonging to the terrain of Krishna river basin within India covering roughly 1500KM by 2000KM (data courtesy of USGS). The terrain model consists of about 28.8 million grid points and has a wide variety of topographies including mountains, riverbeds, flatlands, and highlands, as shown in Figure 1.

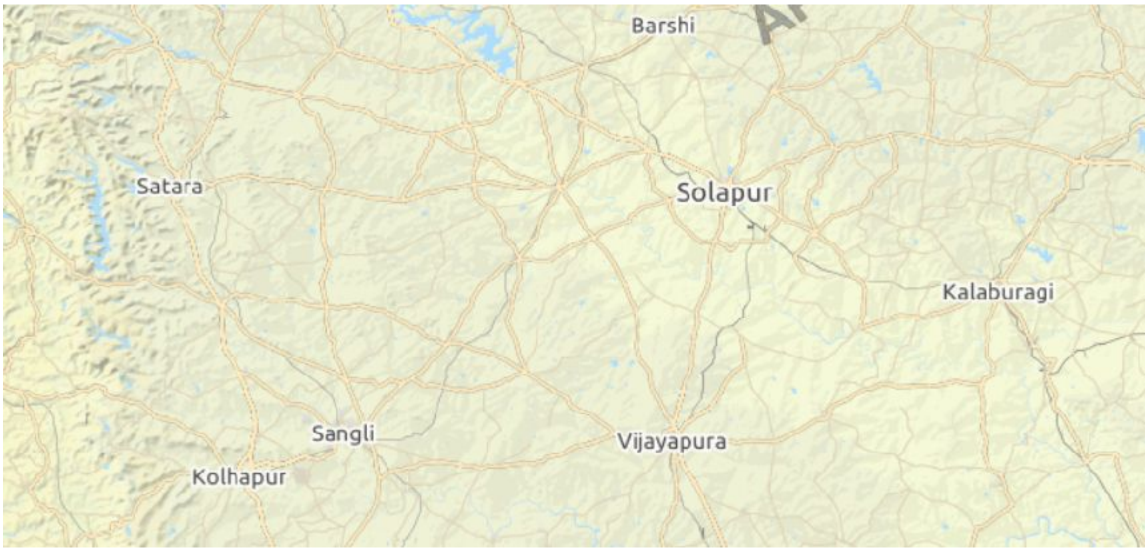

Figure 1: Part of Krishna river basin terrain

Study Area

Table 5 describes the different regions that are considered for evaluating the algorithm. Figure 2 shows the hillshade area of the NSP Right Canal which is discussed in detail in the next seb-section. Figure 3 describes the origination of the NSP Right Canal and its path flow from the origin.

Data Region	Map Extent (UL, LR)	Number of Grids	Length of the Canal
Part of the river along Krishna River	(16.4371, 75.5885), (16.1892, 76.3044)	90	87 Kms
Canal along Ganga basin	(20.5737, 78.2699), (20.29971, 78.7992)	72	78 Kms
Canal along Bembla Dam	(15.2583, 76.3283), (15.5943, 76.8328)	67	72 Kms

Canal along Krishna Sagar Dam	(12.4226, 76.629), (12.2152, 76.9129)	44	42 Kms
Canal along Nagarjuna Dam (NSP Right Canal)	(16.5696, 79.3107), (16.4761, 79.5242)	26	23 Kms

Table 5: Different regions of the study area used to test the algorithm

Results

We present the results obtained from the proposed gravitational flow algorithm. To illustrate the applicability and suitability of the proposed algorithm, NSP right canal will be presented and discussed here. Taking the start and end point of the NSP right canal, Figure 4 shows the height profile for the euclidean path between these points, while Figure 5 shows the height profile for the obtained path based on our algorithm.

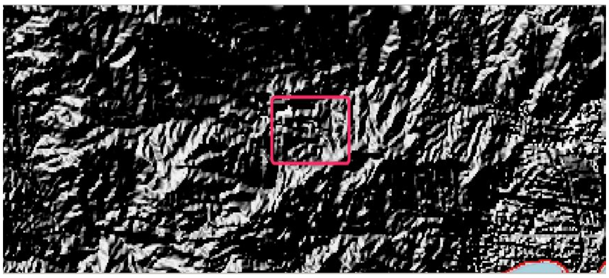

Figure 2: Hillshade data of part of Krishna river basin. The red bounded region is one of the study areas belonging to NSP Right Canal

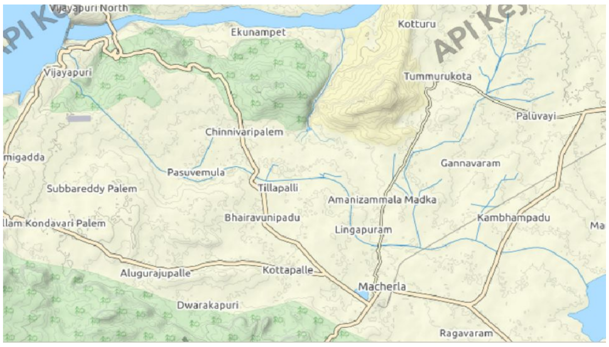

Figure 3: The NSP Right Canal originating from Nagarjuna Sagar Dam

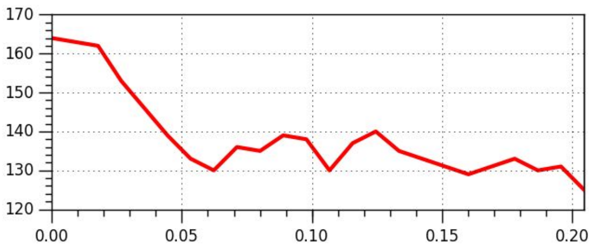

Figure 4: Height(Meters) vs Distance(100x Kms) profile for the euclidean path

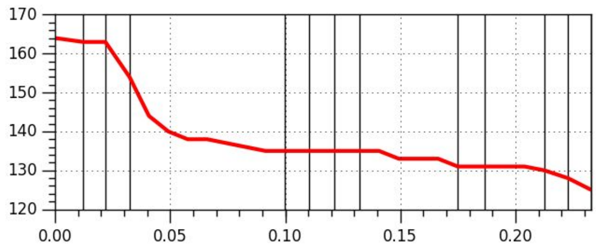

Figure 5: Height(Meters) vs Distance(100x Kms) profile on applying the algorithm

If the Euclidean path is considered as the canal path, the distance travelled will be minimum but the cost will vary depending on the rise and fall that the flow passes through. So, this shortest Euclidean path leads to a high cost as the height variation is irregular, as shown in Figure 4. Considering the elevations in Figure 4, the gravitational flow won't be possible for the canal construction. The only feasible solution is to clear the path such that the elevation profile from the start point to the end point will always be lower from a cell to its succeeding cell. To follow the above procedure it will be expensive as the elevation clearance increases. On the other hand, on applying the proposed gravitational flow algorithm, the distance increases a bit but the cost of construction would be minimal as the flow of water will always be governed by gravitational force and the elevation profile will be such that it supports such gravitational flow. For the NSP right canal, the elevation values range from 164 to 125. The total distance covered by the gravitational path is 23KM, a small increase over the euclidean path of around 20 KM. But the tradeoff of increased path length against the much lesser cost of construction shows that this approach does provide valuable outputs. Figure 6, shows the resultant path obtained by applying the algorithm as grids in black colour and it agrees well with the actual canal path that exists in the real-world data represented by the turquoise vector line. It is overlaid over the DEM data, where the darker colors indicate lower elevations and light color higher elevation values in the region.

Validation of the Results

For validating the results and checking its applicability to the real world datasets, we use the river and canal data from DCW (DMA and USGS, 1996), as rivers follow the gravitational path and the canals are currently operational. The DCW is a standard reference for the real-world dataset generated at 1KM, and used here after rasterizing the vectors. As the grid size is larger than than the original vector contours the positional grid accuracy for rivers and channels is good to a grid/pixel. It is the most comprehensive GIS global database that is freely available. The data contains different thematic layers, of which we are most interested in the hydrography and drainage system.
We define correctness as the percentage of the number of resultant path cells that overlapped with the DCW data to the total number of cells that are present from start point to end point in the DCW data. By using this definition, we calculate the correctness of the algorithm. For the corresponding data listed in Table 5, the accuracy for each path is reported in Table 6. Overall the average accuracy obtained is 82.09% which is reasonably good at this data resolution. It proves that the algorithm scales linearithmically according to the data resolution and also is efficient in terms of time complexity. It can also be seen from Table 6 that as the number of cells in the canal path decreases the accuracy increases, while the accuracy is lower for the longer paths. The computational reason for lower correctness in the later part may be due to multiple paths possible from a grid and the choice may affect the result. On the other hand, the dataset being coarse may not capture the true height changes across the region or between neighbours and affect the algorithm. This may be corrected if the dataset resolution improves.

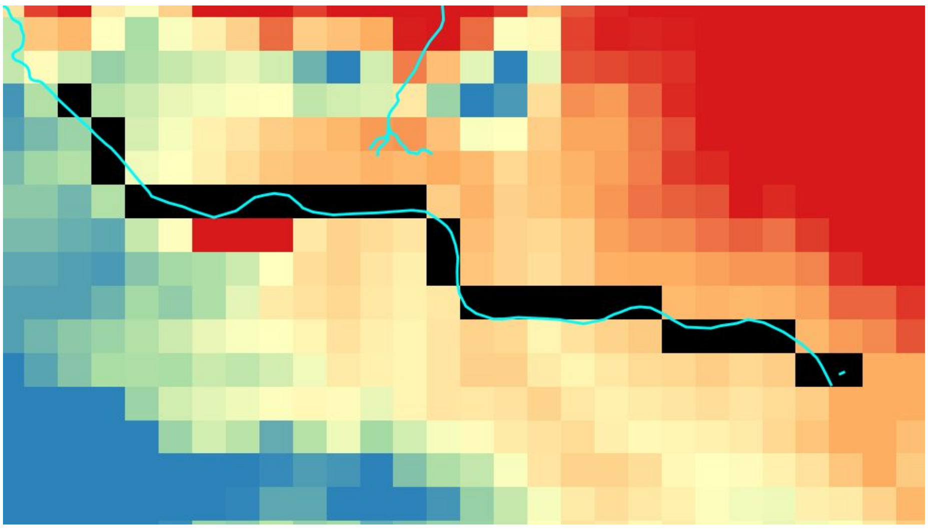

Figure 6: The resultant path with reference to DCW canal data

Region	Overlapped Grids	DCW Grids	Accuracy / Correctness
Part of the river along Krishna River	62	90	68.88%
Canal along Ganga basin	50	72	69.44%
Canal along Bembla Dam	51	67	76.11%
Canal along Krishna Sagar Dam	42	44	95.45%
Canal along Nagarjuna Dam (NSP right canal)	26	26	100%
Average Overall Accuracy			***82.09%***

Table 6: OC: Number of overlapped cells visited with respect to DCW data along the resultant path; **RC:** Number of cells of the DCW data along the path

6. CONCLUSION

As Contour maps and low resolution DEMs are time consuming, a computational algorithm based on the gravitational principle is proposed for canal routing over the DEMs. Also since DEMs of high spatial resolutions are becoming available and the data size will increase enormously, the proposed method can help in better planning and implementation of the canal construction without encountering surprises during the execution of such engineering works. Another advantage of this computational approach is that apart from providing robust canal routing paths, including least cost routing, it allows for taking into account both the topographic and engineering constraints. The analysis of the path generated in addition to its comparison with real-world canals or rivers from DCW data, shows the algorithm can be scaled linearithmically with the data resolutions. Also, the time complexity of the proposed algorithm is better than the existing algorithm. The average accuracy of 82.09% proves that the method can be used in practice.

Further work is needed to assess the impact of multiple resolutions on the algorithm's performance and to include spatial constraints like no-go regions or polygons to be avoided along the path while computing the flow routing. The cost function in the algorithm can be extended to integrate several new parameters based on the land cover/use that it will pass over. One more interesting problem that can be considered is the applicability of the proposed algorithm on designing and planning of different utility infrastructures such as roads.

References:

Bureau of Indian Standards, 2000, Criteria for Design of Lined Canals and Guidance for Selection of Type of Lining.

Criado J., Lopez J. C. and Lopez C., "CANAL: an efficient program for channel routing," Proceedings. Electrotechnical Conference Integrating Research, Industry and Education in Energy and Communication Engineering", 1989, pp. 320-323

Day R. L., T. W. Gardner, K. C. Sasowsky, 1990. Automated extraction of geomorphometric properties from digital elevation data. Zeitschrift fur Geomorphologie, Supplementband, pp. 57–68.

DMA and USGS, 1996. 30" DEM from Digital Chart of the World, from https://www.ngdc.noaa.gov/mgg/topo/report/s5/s5Avii.html

Douglas D.H., 1994. Least cost path in GIS using an accumulated cost surface and slope lines. Cartographica, 31(3), pp. 37–51.
Erwin Weinmann P. and Eric M. Laurenson, 1979. Approximate Flood Routing Methods: A Review, 105(12), pp. 1521-1536.
Fairfield John and Pierre Leymarie, 1991. Drainage networks from grid digital elevation models.Water Resources Research, 27(5), pp. 709-717.
John P. Wilson and John C. Gallant. 2000. Terrain Analysis: Principles and Applications.
Johnson, L. E., 2008. "GIS for water-supply and irrigation systems." Geographic Information Systems in Water Resources Engineering, pp. 137–162.
Lea, N. L., Parsons A. J. and A. D. Abrahams, 1992. An aspect driven kinematic routing algorithm. InOverland flow: Hydraulics and Erosion Mechanics. edited byA. J. Parsons & A. D. Abrahams, pp. 147–175.
Lee J. and Stucky D., 1998. On applying viewshed analysis for determining least-cost paths on Digital Elevation Models. International Journal of Geographical Information Science, pp. 891–905.
Michael L Fredman and Robert Endre Tarjan, 1987. Fibonacci heaps and their uses in improved network optimization algorithms. J. ACM, 12(8), pp. 596–615.
O'Callaghan J. F. and D. M. Mark. 1984. The extraction of drainage networks from digital elevation data.Computer vision, graphics, and image processing, 28(3), pp. 323-344.
Scoging H., 1992. Modelling overland-flow hydrology for dynamic hydraulics. InOverland flow, Hydraulics and Erosion Mechanics. edited by A. J. Parsons & A. D.Abrahams, pp. 89–103.
Smith, J., 1989. Architectural Photogrammetry. Elsevier, Amsterdam, pp. 321-332.
Tarboton David G., 1998. A new method for the determination of flow directions and upslope areas in grid digital elevation models.Water Resources Research, 33(2), pp. 309–319.
Thomas, L., 1990. Close-range camera calibration. In: Industrial Photogrammetry, edited by Smith, J., John Wiley & Sons, New York, pp. 403-445.
Tribe A., 1992, Automated recognition of valley lines and drainage networks from grid digital elevation models: a review and a new method. Journal of Hydrology, 139, pp. 263–293.
Walter Collischonn and Jorge Victor Pilar, 2000. A direction dependent least-cost-path algorithm for roads and canals. International Journal of Geographical Information Science, 14(4), pp. 397–406.
Watson D., D. Tarboton, K. Schreuders and M. Baker., 2011. Generalized terrain-based flow analysis of digital elevation models. InProceedings of the 18th WorldIMACS Congress and MODSIM09 International Congress on Modelling and Sim-ulation, Modelling and Simulation Society of Australia and New Zealand andInternational Association for Mathematics and Computers in Simulation, pp. 2000–2006.
Weihua Zhang, David R. Montgomery, 1994. Digital elevation model grid size,landscape representation, and hydrologic simulations. Water Resources Research, 30(4), pp. 1019–1028