Topology Aligned Least Cost Routing Model for Canals

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in ***Computer Science and Engineering*** *by Research*

by

PONNATHOTA SAI CHAITANYA REDDY
201102202
ponnathotasaichaitanya.reddy@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
JUNE 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Topology Aligned Least Cost Routing Model for Canals” by Ponnathota Sai Chaitanya Reddy, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Dr. K. S. Rajan

To my advisor Dr. K. S. Rajan

Acknowledgments

I am deeply grateful to my advisor, Dr. K. S. Rajan, for his immense support and patience during my research career. He has provided valuable insights during discussions for further improvements, offered opportunities to collaborate and explore, and consistently supported my research efforts.

I want to express my gratitude to the institution for providing a wonderful environment that has helped me learn and grow both academically and personally. I want to thank my family for their support and understanding throughout my academic journey. I also would like to thank my friends for whom I am immensely grateful for all the support they have provided during the journey.

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. 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 article discusses the problem of canal route analysis and proposes a least-cost route model (LCRM) for canals between two points, given the grid-based Digital Elevation Models (DEMs), a unit cost of construction per length, cost of lift to raise water along the surface of the terrain up to a height of 10 meters, set of coordinates the resultant flow needs to pass-through.

This work develops two generic models namely, Gravitational Flow Model (GFM) where the flow is unidirectional and under the force of gravity; and Lift Based Flow Model (LBFM) where anti-gravitational force or pumping is used to lift the water along the surface of the terrain. We present an optimised version of Lift based flow model as well to reduce the number of computations. By implementing and verifying the algorithms on real-world data sets, the correctness of these algorithms will be based on the Digital Chart of the World (DCW) data sets. The algorithm correctness values for 1KM resolution stand at 82.10%, whereas for 90 meters resolution stands at 82.08%. The average value stands out to be 82.09% proving that both the GFM and LBFM algorithms are very effective in practice.

The study also advances the problem of LCRM by introducing a more complex problem that balances cost and distance considerations that is a trade-off between the cost and the distance. Multiple use-cases will be examined to explore this problem and establish its boundaries. The results of the trade-off between the cost and the distance show that for various cases of the terrain and the spatial scale of the data, the patterns tend towards the elliptical bounds, though they can have different patterns at some other parametric combinations.

Contents

List of Figures

List of Tables

Chapter 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. [3, 21, 23, 24, 15].

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.

Consider the Figure 1.1, which describes the NSP Right canal area. Figure 1.2 depicting the profile cross-section of the NSP Right Canal path. This figure provides a two-dimensional view of the terrain's elevation along a specific line, illustrating how the elevation changes over distance. It is evident that the elevation decreases progressively along this 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

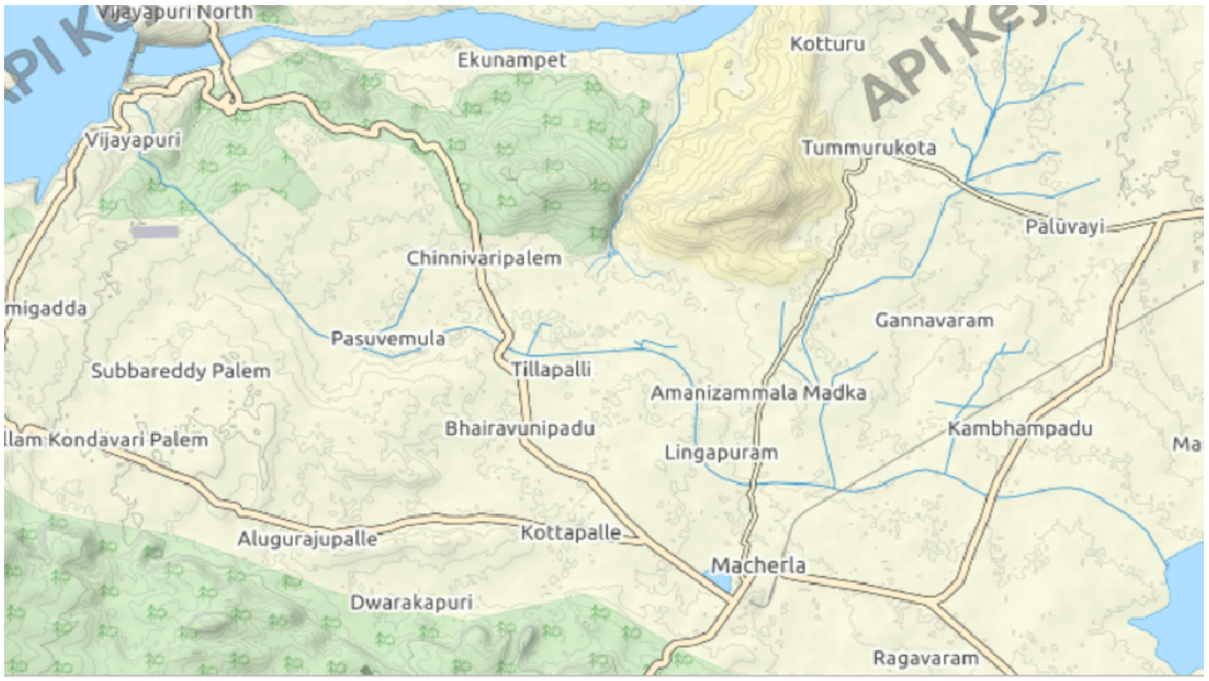


Figure 1.1 NSP Right Canal

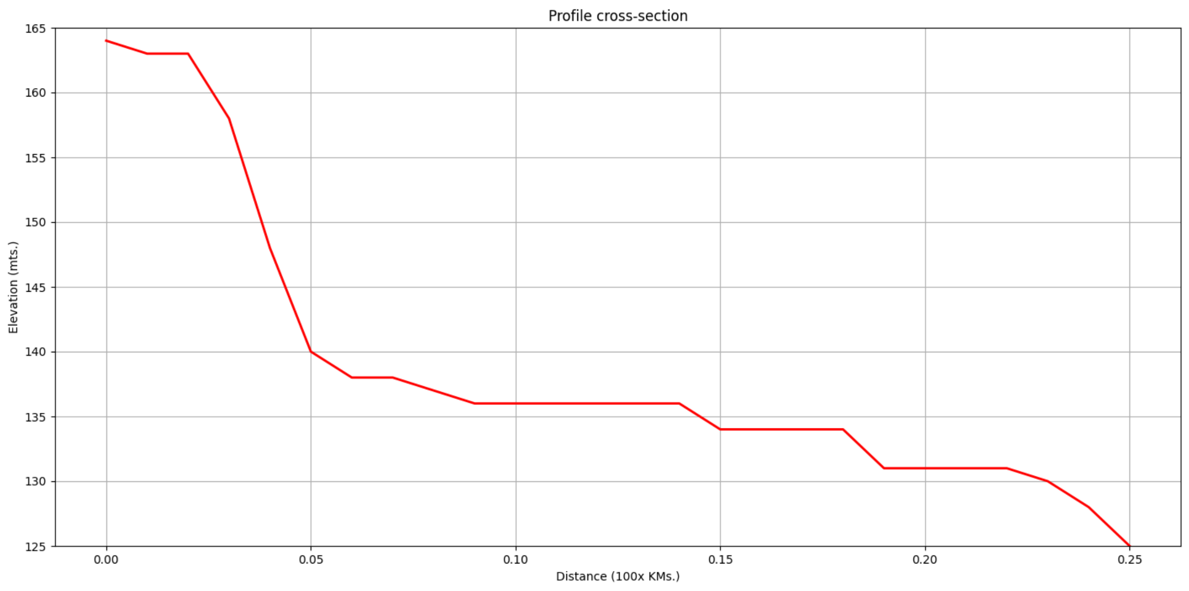


Figure 1.2 Elevation profile of NSP Right Canal area

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.

1.1 Research Question

As mentioned earlier, method for determining canal paths involves a manual process that necessitates repetitive profile visualizations, consuming considerable effort and being susceptible to errors. Despite some studies attempting to devise models for identifying the least-cost path for canals, these efforts have predominantly utilized synthetic datasets rather than real-world data. Consequently, the applicability and accuracy of such models in practical canal routing scenarios remain uncertain. Therefore, there is a need to focus on developing canal routing models considering real-world data to enhance reliability and applicability in diverse geographical contexts. As we move forward, the following questions will be answered:

- Identify the least-cost path for canals.
- Given a cost parameter, identify the possible paths and vice-versa

1.2 Research Objectives

Objectives of the study are as follows:

- Propose computational models to determine the least-cost path for canals
- The proposed solution should be independent of the data resolution
- The proposed solution should demonstrate consistent and dependable performance
- Evaluate the trade-off between cost and distance problem

1.3 Thesis Overview

We define two terminologies, gravitational irrigation, and lift irrigation. The gravitational irrigation is a method of irrigation where the movement of water is under the influence of gravity. Lift irrigation is a method of irrigation where the water is lifted along the surface of the terrain by using external force. Considering the example as shown in Figure 1.3, the path from α to β_1 is defined as gravitational flow where the elevation difference between any cell, ρ and its next neighboring cell, ω is greater than or

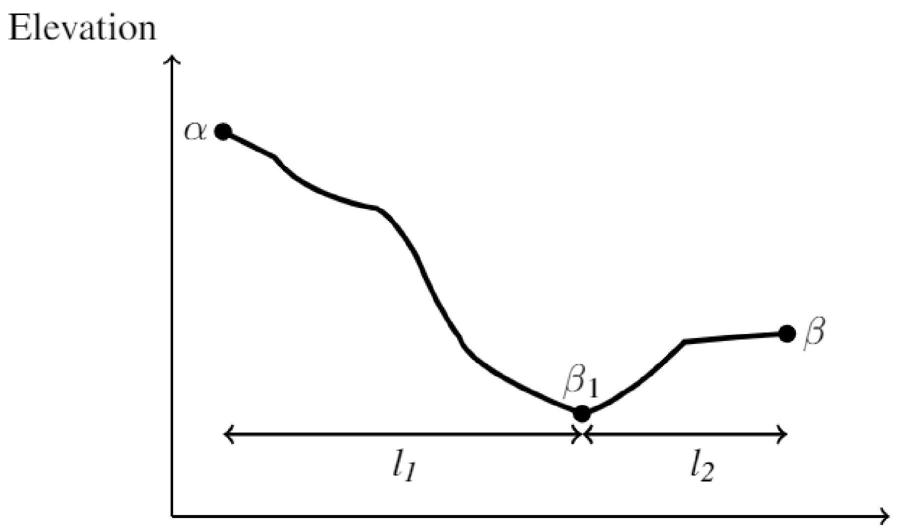


Figure 1.3 Elevation profile of a path

equal to zero, i.e., $\Delta H(\rho, \omega) \geq 0$. The path from β_1 to β is defined as lift based flow where at least one cell exists such that elevation difference between the current cell, ρ and its next neighboring cell, ω is less than zero, i.e., $\Delta H(\rho, \omega) < 0$. In other terms, we define a gravitational flow where the length from α to β_1, $l_1 \geq 0$ and $l_2 = 0$ and the elevation difference between any cell, ρ and its next neighboring cell, ω i.e., $\Delta H(\rho, \omega) \geq 0$. Similarly, lift based flow is defined where $l_1 \geq 0$ and the length from β_1 to β, $l_2 > 0$ and there exists at least one cell such that elevation difference between the current cell, ρ and its next neighboring cell, ω i.e., $\Delta H(\rho, \omega) < 0$. Similarly, you can see in Figure 1.4, a hillshade of a DEM terrain with grey-scale indicating relative elevation, with darker areas, i.e., the black color being lower. The path from α to β_1 indicating gravitational flow path. The path from β_1 to β indicating lift based flow path due to green polygonal region indicating higher elevation.

In this study, we propose two algorithms to determine the least cost route model for canals. 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. In the first algorithm, we discuss the gravitational flow path where the flow movement is based entirely on the gravitational force. In this algorithm, we define a cost function that relates elevation and distance to move from one cell to its neighboring cell. Later we discuss the lift based flow model where the water is lifted along the surface of the terrain. In this lift based flow model algorithm, we first define a lift elevation function that is used to define the elevation that needs to be raised at a cell. We use this lift elevation function to define the cost function that relates elevation and distance. As we now consider lifting the water along the surface, a vast number of combinations are possible between two coordinates. With several such combinations, we try to narrow down the combinations by identifying scenarios that cause the least number of computations and combinations. We also discuss the methodology that is necessary to describe the possible coordinates where the lifts are considered. Considering all the different functions that are defined, in a worst-case scenario, the lift based route model runs in $O(nm^2 \ log \ n)$ time, where m is the set of coordinates the resultant path has to flow through. To the best of our knowledge, this is the first study that proposes the lift irrigation method for the LCRM for Canals problem.

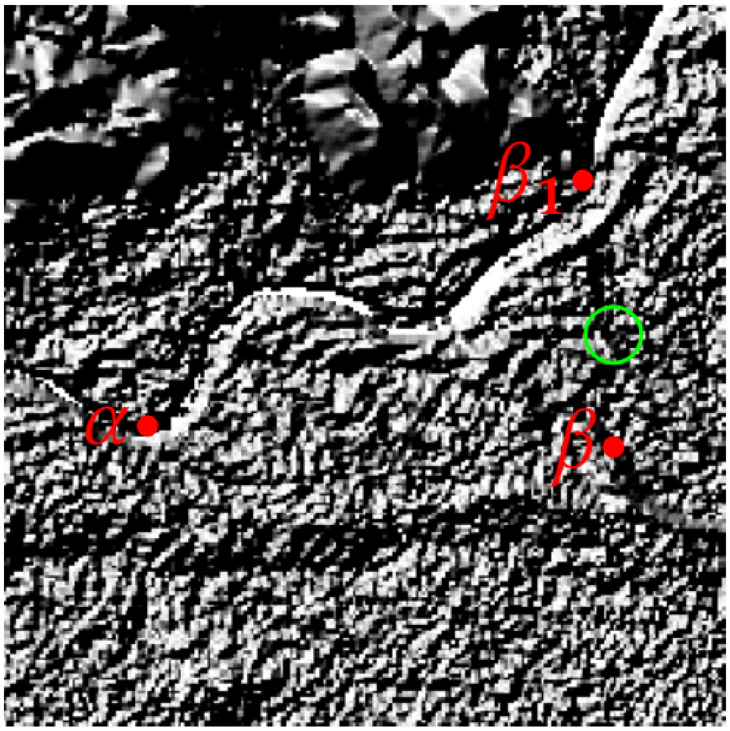


Figure 1.4 Hillshade of a DEM terrain

We run both the algorithms on various real-world terrains with varying inputs belonging to different resolutions and discuss the results. We then conduct an experimental evaluation on both the algorithms, discussing the execution time of the algorithm. We also verify the algorithm correctness using the real-world DCW (Digital Chart of the World) data. Our results also prove that the algorithm is applicable independent of the resolution and scales linearithmically based on the resolution data.

1.3.1 Thesis Structure

The thesis is structured as follows:

- An overview of related work is explained in Section 2.
- Defining the Least Cost Route Model problem in Section 3.
- Solutions to the Gravitational flow model, the lift based flow model and the trade-off between two parameters will be presented in Section 4

- Data and results in Section 5
- Conclusions in Section 6

Chapter 2

Related work

2.1 Types of flow directions

2.1.1 Single flow-direction

Single Flow Direction defines that the total amount of flow should be received by a single neighbouring cell which has the maximum downhill slope with the current cell.

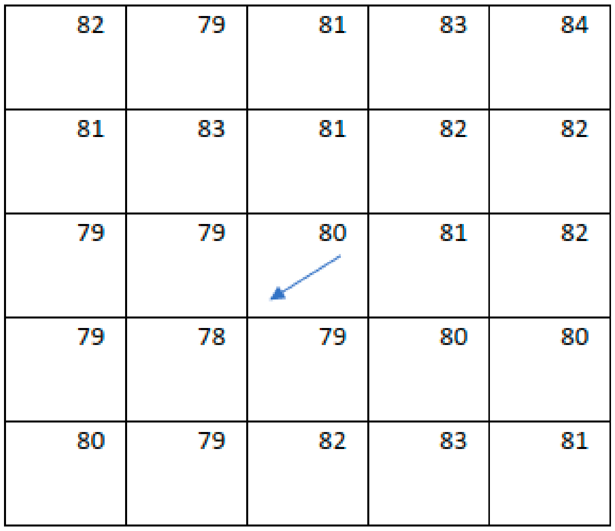
2.1.2 Multi flow-direction

Multiple Flow Direction (MFD) algorithms defines that the flow from the current cell should be distributed to all lower neighbouring cells according to a predetermined rule.

2.2 Applications of flow-direction algorithms

2.2.1 Single flow-direction algorithms

1. **Deterministic eight-node algorithm**: Deterministic eight-node (D8) algorithm was reported by O'Callaghan and Mark in 1984 [17]. On a 3 x 3 local window of a DEM, the slopes between the centre cell and its eight neighbouring cells are computed. The flow direction of the centre cell points to the centre of the neighbouring cell with the maximum downhill slope, and that single neighbour will receive all flow accumulated in the centre cell. Figure 2.1 describes the single flow direction. The algorithm is the most popular one, particularly in commercial GIS software, because of its simple and efficient computation, and strong capability in dealing with local depressions and flat areas (Tarboton 1997 [21]). One major weakness of the D8 algorithm is that although the centre cell can receive upstream flow from several sources, the downstream flow can only be in one direction. Thus it is not suitable for areas where divergent flow occur, such as convex slopes and ridges (Costa-Cabral and Burges 1994 [2, 22], Wilson and Gallant 2000 [24])



82	79	81	83	84
81	83	81	82	82
79	79	80	81	82
79	78	79	80	80
80	79	82	83	81

Figure 2.1 The single flow direction for D8

2. **Random eight-node algorithm**: Random eight-node (Rho8) algorithm (Fairfield and Leymarie 1991 [6]) recognises that the flow over a grid DEM can be arbitrary thus introduced randomness into D8. It is a stochastic version of D8 that aims to break up parallel flow paths that may be resulted from D8, and produces a mean flow direction equal to the aspect. However, it still cannot model flow dispersion (Wilson and Gallant 2000 [24]).

2.2.2 Multi flow-direction algorithms

Multiple flow direction (MFD) algorithms recognise flow divergence over a natural landscape. Thus on a 3 x 3 local window, the flow from the centre cell dose not necessarily point to a single neighbouring cell, but rather, it may flow into all or part of downstream neighbours refer figure 2.2. Based on this principle, a number of algorithms have been developed with various ways of distributing flow proportionally.

1. **QMFD**: Quinn et al. (1991) [18] proposed a flow distribution equation according to slope and contour length (flow width).

$$F_i = \frac{L_i \tan \beta_i}{\sum\limits_{i=1}^{n} L_i \tan \beta_i} (\tan \beta > 0, n \leq 8)$$

where F_i is the proportional flow to be distributed into the i^{th} neighbouring cell; L_i is the flow width; and $\tan \beta_i$ denotes the slope between centre cell and the i^{th} neighbouring cell. On a grid DEM, $L_i = \frac{\sqrt{2}}{4} \triangle_d$ $(\triangle_d = cellsize)$ for diagonal cells, while $L_i = \frac{1}{2}\triangle_d$ for others.

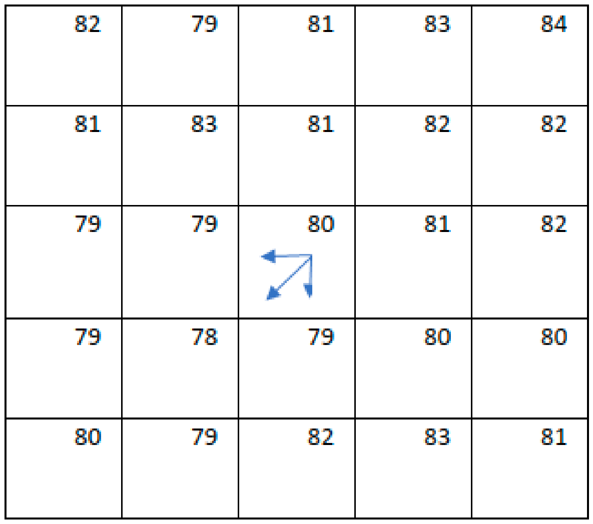


Figure 2.2 The multiple flow direction

2. **FMFD**: Freeman(1991) [8] took a similar approach but did not consider the flow width.

$$F_i = \frac{(\tan \beta_i)^\rho}{\sum_{i=1}^{n} (\tan \beta_i)^\rho} (\tan \beta > 0, n \leq 8)$$

   Based on the test on a conical surface, Freeman found that $\rho = 1.1$ produced the most accurate results. This model was later tested by Pilesjö and Zhou (1997) on a spherical surface and it was found that $\rho = 1.0$ would be more appropriate for the spherical surface case.

3. **DEMON**: In Digital Elevation Model Networks (DEMON) [2], flow is generated at each cell (source cell) and routed down a stream tube until the edge of the DEM or a local depression is reached. The stream tubes are formed from the points of intersections of a line drawn in the aspect and a cell edge. The amount of flow, expressed as a fraction of the area of the source cell, is added to the flow accumulation value of the downstream cell, thus the source cell will provide an impact value on each of the cell along the stream tube. After flow has been generated on all cells and its impact on each of its 'stream tube' cells has been added, the canal flow accumulation value is the total upslope area contributing runoff to each cell, i.e. Specific Catchment Area.

2.3 Grid vs Graph algorithms

In a graph structure, there will be a set of nodes namely the vertices, and the nodes are connected by edges which represent the relationship between the nodes. The graph algorithms are widely used in network routing, transportation planning, and telecommunications, where the relationships between entities are dynamic and require efficient routing decisions based on various criteria such as distance, cost, or capacity. In case of grid structure, each cell represents a node. And each cell has 8 neighbours representing 8 edges with weight equivalent to the distance between the nodes. Grid-based algorithms simplify complex environments into a structured grid, allowing for efficient traversal and path finding using algorithms like A* or Dijkstra's algorithm.

2.4 Least-cost path algorithms for canals

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 grid 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 [17]. Fairfield and Leymarie [6] modified this algorithm to include a stochastic, quasi-random component. Gardner [20] 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 [13] 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 [9] 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 [4] 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 of isotropic values. Lee [14] 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 Collischonn and Jorge Victor Pilar [1] 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 $60x70$ 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.

Chapter 3

Problem Statement and Proposals

3.1 LCRM Problem Statement

In this section, we will define the problem statement for the Least-cost route model. We will describe the definitions and notations that will be necessary as we go proceed.

3.1.1 Terrain

Definition 1. A terrain τ is modeled by a tuple (B_L, T_R, R, C, h) where

- $B_L \rightarrow$ Bottom left coordinates
- $T_R \rightarrow$ Top right coordinates
- R $\rightarrow$ Number of rows
- C $\rightarrow$ Number of columns
- $h \rightarrow$ Elevation of a cell, given a row and a column

Definition 1 describes about the data attributes that will be used as one of the inputs for this problem. This input is a terrain data that is represented as a series of points along X-Y axis generally termed as longitude and latitude respectively. A Z-axis is associated to this X-Y axis determining the elevation at a given point. The data is represented in the form of a 2D-matrix where rows corresponds to the longitudes and the columns corresponds to the latitudes. The value of a particular row and column determines the elevation at that coordinate.

3.1.2 What is LCRM problem

Definition 2. The LCRM problem is defined by the tuple (τ, S_C, E_C, γ, C_l, ξ, ψ) where

- $\tau \rightarrow$ Terrain

- $S_C \rightarrow$ Start coordinates
- $E_C \rightarrow$ End coordinates
- $\gamma \rightarrow$ Cost of construction per unit length
- $C_l \rightarrow$ Cost of raise to lift water to an elevation of 10 meters
- $\xi \rightarrow$ Set of coordinates the resulting flow needs to pass through
- $\psi \rightarrow$ Set of polygonal areas where flow is restricted

Definition 2 formally defines the Least-cost route model. The problem is to find a least-cost path from start coordinate to end coordinate given the parameters for a terrain as described in Definition 1, the cost of construction over the terrain for a unit length being γ, the cost of raise to lift the water along the surface of the terrain up to an elevation of 10 meters being C_l, the resultant flow path passing through the defined set of coordinates ξ and the resultant path that the flow needs to be avoided is ψ.

3.1.3 Solution to the LCRM problem

Definition 3. The solution to the LCRM problem (τ, S_C, E_C, γ, C_l, ξ, ψ) is defined by a sequence of coordinates $\langle P_0,P_1,P_2,....,P_k\rangle$ where

- $P_i \in \tau$, $\forall$ i={0,1,2,3,....,k}
- $\langle P_{i_1}, P_{i_2},....,P_{i_{|\xi|}}\rangle = \xi \in \tau$
- $P_0 = S_C$
- $P_k = E_C$

1. In a gravitational flow path, the sequence of coordinates would be such that
   $h(P_0) \geq h(P_1) \geq h(P_2) \geq .... \geq h(P_k)$
2. In a non-gravitational path, i.e., lift based flow path, the sequence of coordinates would be such that
   $\langle P_0,P_1,P_2,....,P_{s_0},....,P_{e_0},....,P_{s_1},....,P_{e_1},....,P_{s_j},....,P_{e_j},....,P_k\rangle$ where
   - $h(P_0) \geq h(P_1) \geq h(P_2) \geq .... \geq h(P_{s_0})$
   - $h(P_{e_j}) \geq h(P_{e_j+1}) \geq h(P_{e_j+2}) \geq .... \geq h(P_k)$
   - $\forall x$, $h(P_{e_x}) \geq h(P_{e_x+1}) \geq h(P_{e_x+2}) \geq .... \geq h(P_{s_{x+1}})$
     where x={0,1,2,....,j-1}
   - $h(P_{s_0}) < h(P_{s_0+1}) < h(P_{s_0+2}) < .... < h(P_{e_0})$

- $\forall y$, $h(P_{s_y}) < h(P_{s_y+1}) < h(P_{s_y+2}) < .... < h(P_{e_y})$
  where y={0,1,2,....,j}

  $\langle P_{s_0}, P_{s_1}, P_{s_2}, ..., P_{s_j} \rangle$ are the set of coordinates where a lift occurs. $\langle P_{e_0}, P_{e_1}, P_{e_2}, ..., P_{e_j} \rangle$ are the set of coordinates from where a gravitational flow occurs.

Definition 3 defines the solution to the LCRM problem. The solution to this LCRM problem would be a sequence of coordinates $\langle P_0, P_1, P_2, ..., P_k \rangle$ where each coordinate belongs to the terrain. The solution also contains the set of coordinates that the resultant flow must pass through i.e., $\langle P_{i_1}, P_{i_2}, ..., P_{i_{|\xi|}} \rangle = \xi \in \tau$.

As we have described earlier that we would be proposing two new methodologies to solve this LCRM problem namely Gravitational Flow path and Lift Based Flow path. In the gravitational flow path solution, the sequence of coordinates of solution would be such that the elevation of the current cell is always greater than or equal to its next neighbouring cell i.e., $h(P_i) \geq h(P_j)$ where $i < j$ and $\mathrm{P_i}$ represents the point in the resultant flow path. The resultant flow path for the gravitational flow path would be of sequence $h(P_0) \geq h(P_1) \geq h(P_2) \geq .... \geq h(P_k)$ where P_0 corresponds to the start coordinate and P_k corresponds to the end coordinate.

In the Lift Based Flow path, the solution isn't always guaranteed to be as that of Gravitational Flow path. In this method, the solution would be a combination of several intermediate gravitational flow paths as well as several intermediate lift paths. Combining these intermediate paths would result in the lift based flow cost. The path can be described as $\langle P_0, P_1, P_2, ..., P_{s_0}, ..., P_{e_0}, ..., P_{s_1}, ..., P_{e_1}, ..., P_{s_j}, ..., P_{e_j}, ..., P_k \rangle$. Here the paths from P_0 to P_{s_0} and P_{e_j} to P_k determines the gravitational flow paths. Its corresponding height elevation difference would be described as $h(P_0) \geq h(P_1) \geq h(P_2) \geq .... \geq h(P_{s_0})$ and $h(P_{e_j}) \geq h(P_{e_j+1}) \geq h(P_{e_j+2}) \geq .... \geq h(P_k)$. There are certain intermediate paths in the resultant flow path that determines the gravitational flow paths such that $\forall x$, $h(P_{e_x}) \geq h(P_{e_x+1}) \geq h(P_{e_x+2}) \geq .... \geq h(P_{s_{x+1}})$ where x={0,1,2,....,j-1} where the elevation of the current cell is always greater than or equal to its neighbouring cell. The remaining paths determine the lifts that are necessary for the flow from start coordinate to end coordinate. The height elevation of these paths are of the sequence $h(P_{s_0}) < h(P_{s_0+1}) < h(P_{s_0+2}) < .... < h(P_{e_0})$ and $\forall y$, $h(P_{s_y}) < h(P_{s_y+1}) < h(P_{s_y+2}) < .... < h(P_{e_y})$ where y={0,1,2,....,j}. The sequence describes that the elevation at the current cell is always less than its subsequent neighbouring cell. As a whole, the resultant path in a lift based flow model would be a combination where gravitational flow as well as lifting the water along the surface of the terrain occurs. The solution also conveys a pattern describing the coordinates where the lift occurs. The set of coordinates $\langle P_{s_0}, P_{s_1}, P_{s_2}, ..., P_{s_j} \rangle$ describes where the lift occurs since its preceding cell's elevation is greater than or equal to that of the current cell's elevation which is less than its succeeding neighbouring cell's elevation. It also conveys from where the gravitational flow occurs as well. The set of coordinates $\langle P_{e_0}, P_{e_1}, P_{e_2}, ..., P_{e_j} \rangle$ describes where the gravitational flow occurs since its preceding cell's elevation is less than that of the current cell's elevation which is greater than its succeeding neighbouring cell's elevation.

3.1.4 Optimal solution to the LCRM problem

Definition 4. The optimal solution to the LCRM problem (τ, S_C, E_C, γ, C_l, ξ, ψ) is by minimizing the below function:

$$Sol(LCRM) = GFMC(LCRM) + LFMC(LCRM)$$

where GFMC(LCRM) is the gravitational flow path cost and LFMC(LCRM) is the lift based flow path cost.

$$GFMC(LCRM) = \mu * \gamma \tag{3.1}$$

where μ is the total flow path length covered between S_C and E_C.

$$LFMC(LCRM) = \sum_{u=0}^{j} \left\lceil \frac{|h(P_{s_u}) - h(P_{e_u})|}{10} \right\rceil * C_l$$

After defining the resultant path in Definition 3, we now discuss about the optimal to the resultant path. The cost of the LCRM problem would be defined as the sum of gravitational flow path cost and the lift based flow path cost. In order to determine the optimal path, the solution would be formed by minimizing the function, $Sol(LCRM) = GFMC(LCRM) + LFMC(LCRM)$. The cost for the gravitational flow is determined by the total distanced covered to the times i.e., $GFMC(LCRM) = \mu * \gamma$, where μ is the total flow path length covered from start coordinate and end coordinate. As the elevation of current cell is always greater than or equal to its succeeding neighbouring cell, no external force is required for the water to flow and so the cost function for the gravitational flow is based on the distance and the cost of construction.

The cost for lift based flow model is based on the number of lifts present in the resultant flow path times the cost of lift for a single lift of up to 10 meters. The number of lifts is calculated based on the elevation difference along the resultant flow path. As described in Definition 3, the lift based flow paths would be of sequence $\forall y$, $h(P_{s_y}) < h(P_{s_y+1}) < h(P_{s_y+2}) < .... < h(P_{e_y})$ where y={0,1,2,....,j}. Using this sequence we determine the elevation difference. The elevation difference would be $h(P_{e_y})$ - $h(P_{s_y})$ for y={0,1,2,....,j}. So, the number of lifts would be the ceil of difference of the elevation for lift based paths. Since number of lifts will be an integer, it is necessary to round it to the nearest integer. The number of lifts is based on the difference between the start point where the lift occurs and the end point where the lift ends. Number of lifts isn't based on its subsequent neighbouring cells as an external force can lift the water along the surface of the terrain up to 10 meters. Due to this external force, we divide the elevation difference by 10 to determine the lifts that are necessary from start to the end. The lift is then multiplied by the cost of lift. So, this gives us the cost of lift from P_{s_y} to P_{e_y}. Since multiple such lifts can occur in the resultant path, summation is applied for all the paths where a lift is necessary and so, the lift based cost function is defined as

$$LFMC(LCRM) = \sum_{u=0}^{j} \left\lceil \frac{|h(P_{s_u}) - h(P_{e_u})|}{10} \right\rceil * C_l$$

The optimal solution to the LCRM problem can be obtained by minimizing both the functions $GFMC(LCRM)$ and $LFMC(LCRM)$.

3.2 Parameterized Paths problem

In this section, we will define the problem statement for the Parameterized Paths problem. We will describe the definitions and notations that will be necessary as we go proceed.

3.2.1 What is parameterized paths problem

Definition 5. The parameterized paths problem is defined by the tuple (τ, $\mathrm{S_C}$, $\mathrm{E_C}$, γ, C_l, ξ, ψ, N_l) where

- $N_l \rightarrow$ Number of lifts

Definition 5 formally defines one of the variation of the parameterized paths problem. The problem is to find different resultant flow paths where the number of lifts along the path lifts equivalent to N_l from start coordinate to end coordinate given the parameters a terrain as described in Definition 1, the cost of construction over the terrain for a unit length being γ, the cost of raise to lift the water along the surface of the terrain up to an elevation of 10 meters being C_l, the resultant flow path passing through the defined set of coordinates ξ and the resultant path that the flow needs to be avoided is ψ.

Definition 6. The parameterized paths problem is defined by the tuple (τ, $\mathrm{S_C}$, $\mathrm{E_C}$, γ, C_l, ξ, ψ, D_t) where

- $D_t \rightarrow$ Distance travelled

Definition 6 defines the other variation of the parameterized paths problem. The problem is to find different resultant flow paths where the distance travelled by the resultant flow path is equivalent to D_t from start coordinate to end coordinate given the parameters a terrain as described in Definition 1, the cost of construction over the terrain for a unit length being γ, the cost of raise to lift the water along the surface of the terrain up to an elevation of 10 meters being C_l, the resultant flow path passing through the defined set of coordinates ξ and the resultant path that the flow needs to be avoided is ψ.

3.2.2 Solution to the parameterized paths problem

Solution to the parameterized paths problem would be a set of paths from start coordinate $\mathrm{S_C}$ to end coordinate $\mathrm{E_C}$. Contrary to the LCRM problem, the parameterized paths problem doesn't consider the cost as one of the important factors. In the paramterized paths problem, if the parameter is the number of lifts, then for the set of resultant paths from $\mathrm{S_C}$ to $\mathrm{E_C}$, the range of distances would be from $dist_{min}$ to $dist_{max}$ where $dist_{min}$ corresponds to minimum distance travelled for one of the resultant paths. Similarly, $dist_{max}$ corresponds to maximum distance travelled for one of the resultant paths. The distance ranges would be of $dist_{euc} \leq dist_{min} \leq dist_{max} \leq dist_{gra}$; where $dist_{euc}$ is the Euclidean flow path distance from $\mathrm{S_C}$ to $\mathrm{E_C}$. And $dist_{gra}$ is the minimal gravitational flow path distance from $\mathrm{S_C}$ to $\mathrm{E_C}$.

If the parameter is the distance travelled, then for the set of resultant paths from $\mathrm{S_C}$ to $\mathrm{E_C}$, the range of lifts would be from $lift_{min}$ to $lift_{max}$ where $lift_{min}$ corresponds to minimum number of lifts required for one of the resultant paths. Similarly, $lift_{max}$ corresponds to maximum number of lifts required for one of the resultant paths. The lift ranges would be of $lift_{euc} \leq lift_{min} \leq lift_{max} \leq lift_{gra}$; where $lift_{euc}$ is the Euclidean flow path lifts from S_C to E_C. And $lift_{gra}$ is the minimal gravitational flow path lifts from S_C to E_C.

3.3 Proposals

In this section we describe the proposals. To solve LCRM problem, we will be proposing two new methodologies namely Gravitational Flow Model and Lift Based Flow Model.

3.3.1 Gravitational Flow Model

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. For the resultant path to be a downstream flow, the elevation difference between the current cell and its succeeding neighbouring cell should always be greater than or equal to zero. Let C_C be the current cell, and N_C be the succeeding neighbouring cell in the resultant flow. For the flow to be gravitational, the elevation difference between C_C and N_C should be such that $h(C_C) \geq h(N_C)$. So, at any given point P_i in the resultant flow path, the elevation difference would be of sequence $h(P_{i-1}) \geq h(P_i) \geq h(P_{i+1})$.

3.3.2 Lift Based Flow Model

In the gravitational flow model, water flow will always be a downstream flow. If a gravitational flow cannot be found from start coordinate to end coordinate, we propose a new methodology named Lift Based Flow Model. In this model, water is lifted is along the surface of the terrain by using an external force such as lift pumps. Due to elevation being higher in its succeeding neighbour, the water has to flow upstream. By gravitational flow definition, water flow is always downstream. In order for the water to flow upstream, lift pumps would be used to lift the water along the surface of the terrain that can lift the water up to a height of 10 meters. A lift is necessary when elevation difference between C_C and N_C is such that $h(C_C) < h(N_C)$. There would always be at least one cell whose elevation is lower than its succeeding neighbour in case of lift based flow model.

Chapter 4

Flow Models

In this chapter we will discuss the solutions to both the proposals mentioned earlier. We will first discuss the base model solution which doesn't contain any constraints that are applicable to the model and then we extend this base model to consider the constraints.

4.1 Unconstrained flow

In this unconstrained flow, we will discuss how to solve the model by considering only certain parameters as inputs such as start coordinate, end coordinate and the cost of construction per unit length. Discussions on both Gravitational Flow Model as well as Lift Based Flow Model would be discussed.

Assumption: In the resultant flow path, a node once visited cannot be revisited. The resultant flow path won't be a meandering flow.

4.1.1 Gravitational Flow Model

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 1.4; it is a hillshade which is a 3D representation of a terrain surface. The path from α to β_1 can be termed as a gravitational path as the surface is smooth. The surface of the terrain is smooth from α to β_1, and the path identification can be a gravitational flow as no irregularities are present in between.

As the gravitational flow is always downstream, the flow from current cell, C_C to its neighboring cell, N_C is possible only when $h(C_C) \geq h(N_C)$; i.e., an edge exists between C_C and N_C only when $h(C_C) \geq h(N_C)$. Let (ϕ_1, ϕ_2) be the latitudes of C_C and N_C respectively. Similarly, (λ_1, λ_2) be the longitudes of C_C and N_C respectively. The distance between C_C and N_C is calculated by using the Haversine formula [11], defined by $d(C_C, N_C)$ is described as

$$2 * R * \sin^{-1}\left(\sqrt{\sin^2\left(\frac{\phi_1 - \phi_2}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\phi_1 - \phi_2}{2}\right)}\right)$$

where, R is the radius of the Earth. The distance from center cell to its neighbouring cell isn't constant. It varies based on the coordinates from which the flow tries to pass through.

We now determine the cost function that relates elevation and distance for the gravitational flow to move from C_C to N_C as

$$\zeta(C_C, N_C) = \begin{cases} d(C_C, N_C) * \gamma, & \textit{if } h(C_C) \geq h(N_C) \\ NA, & otherwise \end{cases}$$

The cost function is calculated based on the elevation difference from C_C to N_C. The flow is possible only when the center cell's elevation is greater than or equal to its neighbouring cell.

As mentioned earlier, in a Gravitational Flow Model, the resultant path would be such that the preceding cell's elevation should always be greater than or equal to the current cell's elevation which is greater than or equal to the succeeding cell's elevation. Based on this analysis, we will 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 i.e., the cell surrounding the nodes are its edges. 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 a center cell to its neighbouring cell would e the cost fucntion that we have described earlier i.e., $\zeta(C_C, N_C)$.

Multiple graph algorithms are currently in use. We will describe the type of algorithm that will be prefect 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. In the LCRM 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 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 LCRM 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 gird data structure and not a graph data structure which depicts the cost from one node to another. Fist, we define 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 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 queue when the path isn't deviating from the destination coordinate. The cost can be calculated by using the cost function defined earlier as $\zeta(C_C, N_C)$ from 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 with 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 is deviated, then the result won't be a canal path [16]. Several boundaries have been setup 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 limited threshold value

The pseudo code of the algorithm explained is described in Algorithm 1

Algorithm 1 Gravitational Flow Path algorithm

```
1: procedure LEAST COST PATH($terrain, costOfConstruction$)
2:     $Cost[src\_node] = 0; Cost[rest\_nodes] = \infty;$
3:     create priority_queue<customData> Q; add start node to the queue;
4:     while Q is not empty do
5:         $curr\_center\_node \leftarrow$ Get the minimum element from the Queue
6:         for each neighbour of curr_center_node do
7:             if neighbour node is in boundary then
8:                 Calculate the cost from curr_center_node to this neighbouring node
9:                 if $Cost[curr\_center\_node] + \zeta(C_C, N_C) < Cost[this\_neighbour\_node]$ then
10:                    $Cost[this\_neighbour\_node] \leftarrow Cost[curr\_center\_node] + \zeta(C_C, N_C)$
11:                    Update the priority_queue Q
12:                end if
13:            end if
14:        end for
15:    end while
16:    Return the least cost path from $S_C$ to $E_C$
17: end procedure
```

After applying the Dijkstra's algorithm using priority queue min-heap, the resultant coordinates would be of sequence $\langle P_0, P_1, P_2, ..., P_k \rangle$ and the relation between the points can be described as $h(P_0) \geq h(P_1) \geq h(P_2) \geq .... \geq h(P_k)$ where P_0 corresponds to the start coordinate S_C, P_k corresponds to the end coordinate. The set of other points are the coordinates of how path should flow from one point to the another.

The time complexity of the algorithm would be $O(nlogn)$, where n is the total number of cells/nodes in the terrain.

4.1.2 Lift Based Flow Model

It is not necessary to have gravitational flow from start coordinate to end coordinate. The gravitational flow path cannot be achieved when there are coordinates along the path from the start coordinate whose elevation is less that that of its neighbouring nodes i.e., *$h(P_m)$ < elevation of neighbours of P_m*.

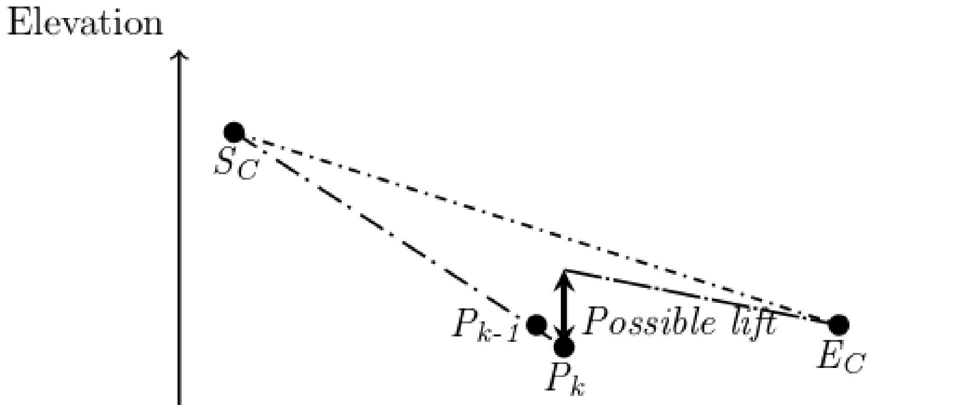


Figure 4.1 Lift flow case1

In order to identify the path model for canals when gravitational flow path cannot be achieved, the concept of lift irrigation, i.e., lifting the water along the surface of the terrain, is considered. In other terms, there exists at least one coordinate in the resultant flow path where $h(C_C) < h(N_C)$. Considering the example shown in Figure 1.4, the path from β_1 to β can be termed as a lift based path as the surface isn't smooth throughout. Along the path, there exists an area where the surface has a higher elevation, meaning that an external force or pumping is necessary for the flow to reach β.

Four different use cases were identified to determine the lifts for lift based flow model

1. Gravitational path till the elevation of E_C and then use a lift
2. Gravitational path to the possible extent and then use a lift
3. Gravitational path above the elevation of E_C and then use a lift
4. Lift at the start i.e., S_C

4.1.2.1 Case1

Considering the example shown in Figure 4.1, there is a gravitational path from S_C to P_k. We define the point P_k where the elevation of its previous point $P_{k\text{-}1}$ is equivalent to the elevation of the

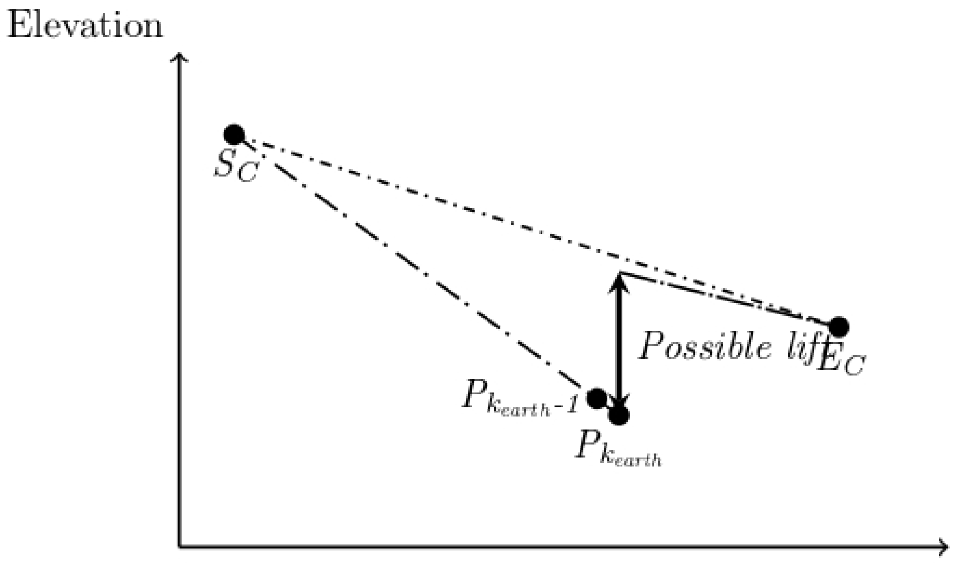


Figure 4.2 Lift flow case2

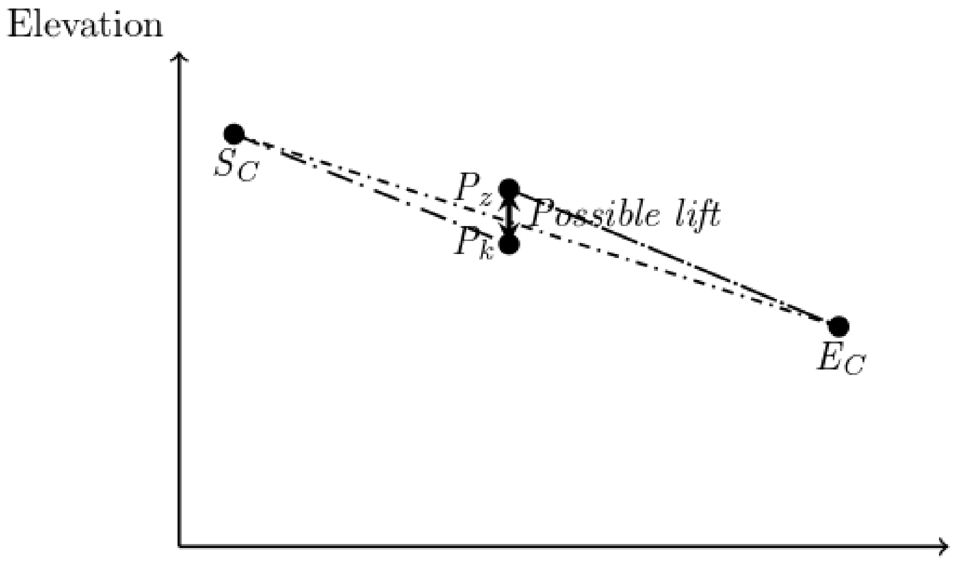


Figure 4.3 Lift flow case3

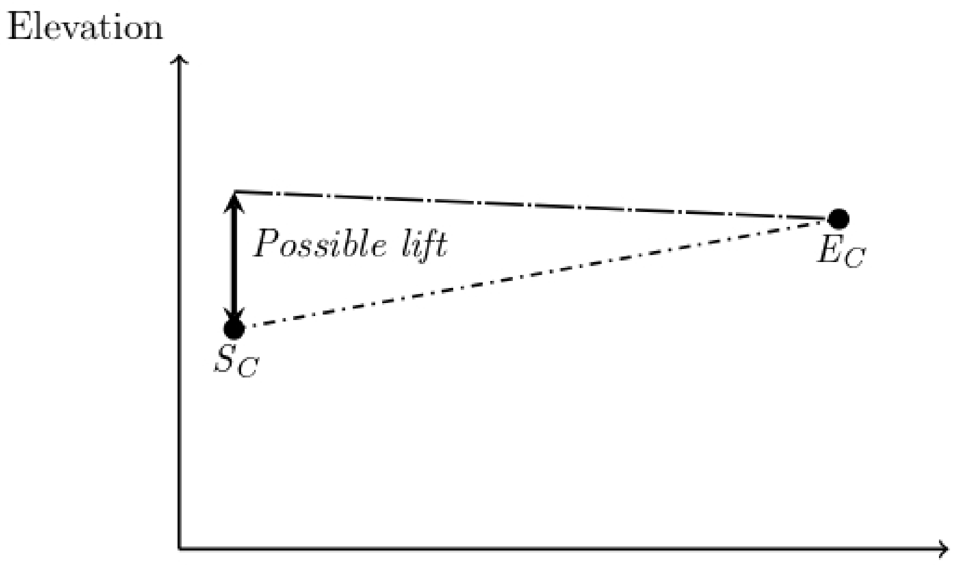


Figure 4.4 Lift flow case4

end coordinate E_C i.e., $h(P_{k-1}) = h(E_C)$. In this case, we consider the lift possibility at P_k so that, we won't continue further. There are 2 points that are possible from this case. One of them is that we are restricting the elevation to not lower below P_k and the other is depending on the terrain, it may or may not require multiple lifts. So, we will be using lift model at P_k and from there once again traverse through using the gravitational path model.

4.1.2.2 Case2

Considering the example shown in Figure 4.2, there is a gravitational path from S_C to $P_{k_{earth}}$. We define the point $P_{k_{earth}}$ where the gravitational path exists through out. In this case, we consider the lift possibility at $P_{k_{earth}}$ as there is no other possibility. The possibility of this use case is that, the path may be stretched due to the gravitational path to the extent that there is no possibility to flow further. So, we will be using lift model at $P_{k_{earth}}$ and from there once again traverse through using the gravitational path model.

4.1.2.3 Case3

Considering the example shown in Figure 4.3, there is a gravitational path from S_C to P_k. We define the point P_k where the gravitational path exists through out. And moreover, the elevation of P_k is higher that the elevation of E_C i.e., $h(P_k) > h(E_C)$. In this case, we consider the lift possibility at P_k to consider a different path. The necessity of this use case is that, a lift at this point may results in a better path which may not require a lift from here on and will most likely be a least-cost path. So, we will be using lift model at P_k and from there once again traverse through using the gravitational path model.

4.1.2.4 Case4

Considering the example shown in Figure 4.4, there is no gravitational path from S_C. Here, the elevation of S_C is less that the elevation of E_C i.e., $h(S_C) < h(E_C)$. In this case, the only possibility is to consider a lift at the start itself. This is a case where no gravitational flow path exists. So, we will be using lift model at S_C and from there once again traverse through using the gravitational path model.

We now define the lift elevation function that determines the elevation difference that needs to be lifted from C_C to N_C in order for the flow to pass. The lift elevation function for lift based flow model from C_C to N_C is:

$$\delta(C_C, N_C) = \begin{cases} \mid h(C_C) \text{ - } h(N_C) \mid + \delta(N_C', C_C), & \textit{if } h(C_C) < h(N_C) \\ NA, & otherwise \end{cases}$$

where $\delta(N_C', C_C)$ is the lift elevation from N_C' to C_C and N_C' is one of the neighbouring cells of C_C and $N_C' \neq N_C$.

The lift elevation function is calculated based on the elevation difference from the center cell to its neighbouring cell. If the elevation of the current cell is greater than or equal to its neighbouring cell, then it isn't necessary to lift the water as the water flow can occur due to gravitational force where flow will be a downstream flow. If the elevation of the current cell is less than its neighbouring cell, then the gravitational flow wouldn't work as the elevation is higher and in order for the water flow to continue through this neighbouring cell, it is necessary to lift the water. And so, the lift raise for this center cell to its neighbouring cell would be its height difference i.e., $\mid h(C_C) \text{ - } h(N_C) \mid$. The cumulative heights lifted for this neighbour cell would be the summation of the center cell's lift elevation. So, the lift elevation function from center cell to its neighbour cell would be $\mid h(C_C) \text{ - } h(N_C) \mid + \delta(N_C', C_C)$. As mentioned earlier, we use the lift technique method that can lift the water up to a height of 10 meters. Considering the scenario explained, this cumulative summation is necessary to determine the lifts.

We now determine the cost function for the lift based flow to move from C_C to N_C. This cost function relates the lift elevation function, elevation and distance from C_C to N_C as

$$\zeta(C_C, N_C) = \begin{cases} \left\lceil \frac{\delta(C_C, N_C) - \delta(N_C', C_C)}{10} \right\rceil * C_l + d(C_C, N_C) * \gamma, & \textit{if } \left\lceil \frac{\delta(N_C', C_C) + \delta(C_C, N_C)}{10} \right\rceil > \left\lceil \frac{\delta(N_C', C_C)}{10} \right\rceil \\ d(C_C, N_C) * \gamma, & \textit{if } \left\lceil \frac{\delta(N_C', C_C) + \delta(C_C, N_C)}{10} \right\rceil = \left\lceil \frac{\delta(N_C', C_C)}{10} \right\rceil \\ d(C_C, N_C) * \gamma, & \textit{if } h(C_C) \geq h(N_C) \end{cases}$$

where $\delta(N_C', C_C)$ is the lift elevation from N_C' to C_C and N_C' is one of the neighbouring cells of C_C and $N_C' \neq N_C$

The primary attributes required to calculate the cost function for the lift based model is based on lift elevation function and the height elevation. If the elevation of the current cell is greater than or equal to its neighbouring cell, then the lift elevation function isn't applicable and in this scenario, the cost would be the equivalent to the distance travelled times the cost of canal construction which is $d(C_{\mathrm{C}}, N_{\mathrm{C}}) * \gamma$. If the elevation of the current cell is less than its neighbouring cell, then a lift is necessary in this case. However, the necessity of one more lift is dependent on the lift elevation function reason being that the external force used to lift the water along the surface of the terrain can lift up to a height of 10 meters. So, it isn't necessary to lift the water along the surface at each cell. The lift technique will consider this lift height mechanism. To determine if an additional lift is necessary, we compare the number of lifts using lift elevation from the current cell to its neighbouring cell to the number of lifts at the center cell which are $\left\lceil \frac{\delta(N_{\mathrm{C}}', C_{\mathrm{C}}) + \delta(C_{\mathrm{C}}, N_{\mathrm{C}})}{10} \right\rceil$ and $\left\lceil \frac{\delta(N_{\mathrm{C}}', C_{\mathrm{C}})}{10} \right\rceil$. If no additional lift is necessary from current cell to its neighbouring cell, then the cost would be equivalent to the distance travelled times the cost of canal construction which is $d(C_{\mathrm{C}}, N_{\mathrm{C}}) * \gamma$. If additional lift is necessary then the cost function would be the summation of the cost for the number of lifts and the cost of canal construction. So, the cost for the lifts would be the number of lifts times the cost of each lift. The cost function if a lift is necessary is evaluated as $\left\lceil \frac{\delta(C_{\mathrm{C}}, N_{\mathrm{C}}) - \delta(N_{\mathrm{C}}', C_{\mathrm{C}})}{10} \right\rceil * C_l + d(C_{\mathrm{C}}, N_{\mathrm{C}}) * \gamma$.

Remark. *The number of lifts is the primary constraint for the cost calculation rather than the summation of lifted heights. (For Ex. Lifts of (5,5) are considered over lifts of (2,3,4). Since 5+5 requires two different lifts whereas 2+3+4 requires three different lifts)*

For a lift to occur, the condition $h(C_C) < h(N_C)$ needs to be satisfied. There will be a huge number of cases where the condition could be satisfied. We now discuss different scenarios that are possible for lifting the water for coordinates from σ_{i} to $\sigma_{\mathrm{i+1}}$ which results in determining least-cost path.

- Let $P_{k,E}$ be the gravitational path from σ_{i} to P_k. We define a coordinate P_k, where $h(P_k)$ is just less than $h(\sigma_{i+1})$ i.e., $h(P_k) < h(\sigma_{i+1})$ and $h(P_{k-1}) \geq h(\sigma_{i+1})$. We consider this coordinate, P_k, suitable for lifting along the surface of the terrain.

- Let $P_{k_{earth},E}$ be the gravitational path from σ_{i} to $P_{k_{earth}}$. We define a coordinate $P_{k_{earth}}$, such that $h(P_{k_{earth}})$ is less than all of its neighboring cells i.e., there is no such neighbouring coordinate P_z of $P_{k_{earth}}$ where a gravitational flow from $P_{k_{earth}}$ to P_z exists; i.e., $h(P_{k_{earth}-1}) < h(\sigma_{i+1})$, $h(P_{k_{earth}}) < h(\sigma_{i+1})$ and $h(P_{k_{earth}-1}) \geq h(P_{k_{earth}})$. We consider this coordinate, $P_{k_{earth}}$, suitable for lifting along the surface of the terrain.

- Let $P_{k,E}$ be the gravitational path from σ_{i} to P_k. We define a coordinate P_k, such that $h(P_k)$ is less than all of its neighboring cells but greater than the elevation of σ_{i+1} i.e., there is no such neighbouring coordinate P_z of P_k where a gravitational flow from P_k to P_z exists; i.e., $h(P_{k-1}) \geq h(\sigma_{i+1})$, $h(P_k) \geq h(\sigma_{i+1})$ and $h(P_{k-1}) \geq h(P_k)$. We consider this coordinate, P_k, suitable for lifting along the surface of the terrain.

- If $h(\sigma_i) < h(\sigma_{i+1})$, then we consider the coordinate σ_{i} suitable for lifting along the surface of the terrain.

By defining the scenarios above, the aim is to reduce the cost complexity for every scenario where the elevation difference is higher i.e., $h(C_C) < h(N_C)$.

Lemma 4.1.1. *Let A, B $\in \sigma$ and h(A) > h(B). If a gravitational flow least cost path exists from A to B, it isn't necessary that the gravitational flow path would be of the least cost from A to B.*

Proof. Let the gravitational flow least cost from A to B be GFC(A,B), C be a coordinate such that *C* $\in \tau$ and $h(C) < h(A)$, the lift based flow least cost from A to B be LFC(A,B), D be a coordinate on the gravitational flow path from A to B such that *D* $\in \tau$ where $h(A) > h(D)$ and $h(D) > h(B)$. Let LFC(A,C) be less than the GFC(A,D) and GFC(D,B) = LFC(C,B). The gravitational flow cost from A to B is given by:

$$GFC(A,B) = GFC(A,D) + GFC(D,B)$$

The lift based flow cost from A to B is given by

$$LFC(A,B) = LFC(A,C) + LFC(C,B)$$

Since $GFC(C,B) = LFC(D,B)$, we rewrite the above statement as

$$LFC(A,B) = LFC(A,C) + GFC(D,B)$$

Since $LFC(A,C) < GFC(A,D)$, we can rewrite the above statement as

$$LFC(A,B) < GFC(A,D) + GFC(C,B)$$

Since $GFC(A,B) = GFC(A,D) + GFC(D,B)$, the statement translates to

$$LFC(A,B) < GFC(A,B)$$

Hence, if a gravitational flow least-cost path exists from A to B, it is not necessary that the gravitational flow path would be of the least cost from A to B. □

Remark. *The higher the lift elevation, the higher the coverage is possible, depending on the terrain. When doing so, the number of lifts needs to be maintained in memory. (For Ex. Let 400 be the elevation of the center cell, 403 and 409 be the elevations of neighbors of the center cell. It may be necessary that cell having an elevation of 409 might have higher coverage when estimated the impact using viewshed analysis. [14, 12])*

After defining the lift elevation function and cost functions for the lift based flow model, the least cost path algorithm needs to be defined. The input parameters for the lift based flow model are the same as the input parameters for the gravitational flow model. Due to the same grid data structure as the input, we will be reusing the Algorithm 1 to solve the Lift Based Flow Model based on these requirements. The procedure almost remains the same except that the based on whether the current cell is recommended to lift at this cell. If it is recommended to lift for the center cell, then calculate the lift elevation value and its corresponding cost for the path from center cell to its neighbouring cell.

Algorithm 2 Lift Based Flow Path algorithm

```
1:  procedure LEAST COST PATH($terrain, costOfConstruction$)
2:      $Cost[src\_node] = 0; Cost[rest\_nodes] = \infty$;
3:      create priority_queue<customData> Q; add start node to the queue;
4:      while Q is not empty do
5:          $curr\_center\_node \leftarrow$ Get the minimum element from the Queue
6:          for each neighbour of curr_center_node do
7:              if neighbour node is in boundary then
8:                  if current node needs to be lifted then
9:                      Calculate the lift elevation value and cost value
10:                 else
11:                     Calculate the cost from curr_center_node to this neighbouring node
12:                 end if
13:                 if $Cost[curr\_center\_node] + \zeta(C_C, N_C) < Cost[this\_neighbour\_node]$ then
14:                     $Cost[this\_neighbour\_node] \leftarrow Cost[curr\_center\_node] + \zeta(C_C, N_C)$
15:                     Update the lift elevation value
16:                     Update the priority_queue Q
17:                 end if
18:             end if
19:         end for
20:     end while
21:     Return the least cost path from $S_C$ to $E_C$
22: end procedure
```

The resultant flow path in a Lift Based Flow Model can be described as $\langle P_0, P_1, P_2, \ldots, P_{s_0}, \ldots, P_{e_0}, \ldots, P_{s_1}, \ldots, P_{e_1}, \ldots, P_{s_j}, \ldots, P_{e_j}, \ldots, P_k \rangle$. Here the paths from P_0 to P_{s_0} and P_{e_j} to P_k determines the gravitational flow paths. Its corresponding height elevation difference would be described as $h(P_0) \geq h(P_1) \geq h(P_2) \geq \ldots \geq h(P_{s_0})$ and $h(P_{e_j}) \geq h(P_{e_j+1}) \geq h(P_{e_j+2}) \geq \ldots \geq h(P_k)$. There are certain intermediate paths in the resultant flow path that determines the gravitational flow paths such that $\forall x$, $h(P_{e_x}) \geq h(P_{e_x+1}) \geq h(P_{e_x+2}) \geq \ldots \geq h(P_{s_{x+1}})$ where x={0,1,2,....,j-1} where the elevation of the current cell is always greater than or equal to its neighbouring cell. The remaining paths determine the lifts that are necessary for the flow from start coordinate to end coordinate. The height elevation of these paths are of the sequence $h(P_{s_0}) < h(P_{s_0+1}) < h(P_{s_0+2}) < \ldots < h(P_{e_0})$ and $\forall y$, $h(P_{s_y}) < h(P_{s_y+1}) < h(P_{s_y+2}) < \ldots < h(P_{e_y})$ where y={0,1,2,....,j}

4.2 Constrained Flows

What is a constrained flow? A constrained flow is represented as the flow where the resultant flow path is found to be controlled due to certain restrictions provided by the user. Why is a constrained flow necessary? In an ideal world, one would consider that constraints wouldn't be part of it. Whereas, in a real-world scenario, multiple factors play significant roles in determining the problem's outcome. Among multiple constraints applicable in the real-world, we define two predominant constraints that significantly impact our problem's development. One of the parameters is the restricted polygon regions where the canal's construction is restricted in the areas defined. The canal construction is dependent on term Land use. Land use describes land usage, such as agriculture, industrial, residential, commercial, transport, etc. For example, industrial and commercial types of land use regions cannot be used for canal construction. The other parameter is the set of pass-through coordinates. As canal construction is viewed as utility infrastructure, it is necessary to determine the best outcome for longer lengths canal, keeping in view that this utility infrastructure will be accessible for vast regions. By defining these predominant constraints, the proposed algorithms can determine the optimal path on the basis of cost. We will discuss each of the parameters in detail and update the current models to achieve the least cost paths.

4.2.1 Pass through coordinates

As described earlier, pass through coordinates describes the set of coordinates that the resultant flow needs to visit these coordinates. The set of pass through coordinates is defined as ξ. The resultant flow path starting from S_C to E_C should pass through all the coordinates of ξ.

4.2.2 Restricted Polygonal Regions

As described earlier, restricted polygonal regions defines the set of polygonal regions where the flow is restricted. We define the set of polygonal regions as ψ. The resultant flow path starting from S_C to E_C should avoid polygonal regions ψ.

4.2.2.1 Gravitational Flow Model

In an unconstrained flow, the least cost route model path needs to be identified from S_C to E_C. In a constrained flow, the path needs to be found from S_C to E_C covering all the pass through coordinates ξ as well as avoiding the restricted regions ψ.

In order to determine the set of coordinates that belong to the restricted polygonal regions, we use the Ray Casting algorithm [19] to determine if a coordinate belongs to one of the restricted polygons. Ray Casting algorithm is based on how often a ray starts from the point and goes in any fixed direction, intersecting polygon's edges. The Ray Casting algorithm can be used for each cell of the terrain, τ to determine if the cell is restricted for the water flow. The time complexity of this would be $O(n*|\psi|)$.

As the bounded regions increase, the time complexity increases. To optimize the number of computations, coordinates data is cached when calculating for the first time a row. This helps in evaluating if a coordinate belongs to the bounded regions while calculating for subsequent cells of a row. The time complexity for calculating a row while visiting for the first time is *O(*$|\psi|$*)*. For all the rows of a terrain τ, is *O(R**$|\psi|$**w)* where $|\psi|$ is the total number of vertices of all polygons and *w* is the average number of points of all polygons ψ. For the subsequent iterations of a row, identifying if a cell belongs to bounded regions of ψ can be identified in a constant time.

After determining whether a coordinate $P_q \in \tau$ is in bounded regions of ψ, the elevation of P_q is set to ∞; i.e., $h(P_q) = \infty$ as we can no longer route the model through this coordinate.

Theorem 4.2.1. *Let A, B $\in \xi$ and h(A) > h(B). If a gravitational flow path from A to B doesn't exist, then there is no gravitational flow path from S_C to E_C passing through all coordinates of ξ*

Proof. Consider two arbitrary coordinates U, V $\in \xi \ni$ *h(U) > h(A)* and *h(B) > h(V)*. No gravitational flow path exists between the coordinates U and V passing through A; otherwise there would be gravitational flow path from U to V passing through A and B as $\{A, B\} \in \xi$. Hence if there is no gravitational path from U to V, then there is no gravitational path from $\mathrm{S_C}$ to $\mathrm{E_C}$ passing through all coordinates of ξ as the resultant flow path must start from $\mathrm{S_C}$ and end with $\mathrm{E_C}$ covering all the coordinates of ξ. □

We define the union of start coordinate, the set of coordinates to that the resultant flow path needs to pass through ξ and the end coordinate as σ such that $\sigma = \{\mathrm{S_C}\} \cup \{\xi\} \cup \{\mathrm{E_C}\}$. We reuse the Gravitational flow model algorithm that was explained earlier to determine the least cost model by considering these additional parameters. The Gravitational Flow Model algorithm will be remodeled to satisfy the new requirements. We first try to preprocess the restricted polygonal regions to identify the set of coordinates that will be updated in the model to avoid traversing through these coordinates. As mentioned in Theorem 4.2.1, if there is no gravitational flow path from σ_i to $\sigma_\mathrm{i+1}$, then there is no gravitational flow path from $\mathrm{S_C}$ to $\mathrm{E_C}$; in such cases we just abort the algorithm execution. We use the Gravitational Flow Model algorithm to identify the resultant flow path that passes through each coordinate defined in ξ.

Algorithm 3 Constrained Gravitational Flow Path algorithm

```
1: procedure LEAST COST PATH($terrain, costOfConstruction, \xi, \psi$)
2:     Compute the polygonal regions $\psi$ and modify the elevation if $P_q \in \xi$ such that $h(P_q) = \infty$
3:     for each coordinate i in $\sigma$ do
4:         if $h(\sigma_i) < h(\sigma_{i+1})$ then
5:             Return no gravitational path from $S_C$ to $E_C$
6:         end if
7:         Calculate Gravitational Flow Model Least cost path from $\sigma_i$ to $\sigma_{i+1}$
8:     end for
9:     Return the least cost path from $S_C$ to $E_C$
10: end procedure
```

After applying the Gravitational Flow Model algorithm using priority queue min-heap for the defined set of new parameters, if a resultant flow path exists, then the resultant flow path coordinates would be of sequence $\langle P_0,P_1,P_2,....,P_{\xi_0},....,P_{\xi_1},....,P_{\xi_{|\xi|}},....,P_k\rangle$ and the relation between the points can be described as $h(P_0) \geq h(P_1) \geq h(P_2) \geq .... \geq h(P_{\xi_0}) \geq .... \geq h(P_{\xi_1}) \geq .... \geq h(P_{\xi_{|\xi|}}) \geq .... \geq h(P_k)$ where P_0 corresponds to the start coordinate S_C, P_k corresponds to the end coordinate. $\langle P_{\xi_0}, P_{\xi_0}, ..., P_{\xi_{|\xi|}} \rangle \in \xi$. The set of other points are the coordinates of how path should flow from one point to the another.

Let n be the number of cells of the terrain τ, R be the total number of rows of the terrain τ, w is the average number of points of all polygons ψ, m is $|\xi|$+1. The time complexity for processing the restricted polygons is $O(|R| * |\psi| * w)$ whereas the time complexity for finding out the path between σ_i and σ_{i+1} is $O(n \log n)$ by using Dijkstra's algorithm with Fibonacci heaps [7]. The overall time complexity for Least Cost Route Model for Gravitational Flow is $O(|R| * |\psi| * w + nmlogn + n)$.

4.2.2.2 Lift Based Flow Model

The preprocessing for the Lift Based Flow Model would the equivalent to that of the Gravitational Flow Model. We use the earlier defined Lift Based Flow Model algorithm to identify the resultant flow path that passes through each coordinate defined in ξ and avoids the restricted polygonal regions ψ.

Algorithm 4 Constrained Lift Based Flow Path algorithm

```
1: procedure LEAST COST PATH($terrain, costOfConstruction, passThroughCoordinates$)
2:     Compute the polygonal regions $\psi$ and modify the elevation if $P_q \in \xi$ such that $h(P_q) = \infty$
3:     for each coordinate $i$ in $\sigma$ do
4:         Calculate Lift Based Flow Least cost path from $\sigma_i$ to $\sigma_{i+1}$
5:     end for
6:     Return the Lift Based Flow least cost path from $S_C$ to $E_C$
7: end procedure
```

The path can be described as $\langle P_0,P_1,P_2,....,P_{s_0},....,P_{e_0},....,P_{s_1},....,P_{e_1},....,P_{s_j},....,P_{e_j},....,P_k\rangle$. Here the paths from P_0 to P_{s_0} and P_{e_j} to P_k determines the gravitational flow paths. Its corresponding height elevation difference would be described as $h(P_0) \geq h(P_1) \geq h(P_2) \geq .... \geq h(P_{s_0})$ and $h(P_{e_j}) \geq h(P_{e_j+1}) \geq h(P_{e_j+2}) \geq .... \geq h(P_k)$. There are certain intermediate paths in the resultant flow path that determines the gravitational flow paths such that $\forall x$, $h(P_{e_x}) \geq h(P_{e_x+1}) \geq h(P_{e_x+2}) \geq .... \geq h(P_{s_{x+1}})$
where x=$\{0,1,2,....,j\text{-}1\}$ where the elevation of the current cell is always greater than or equal to its neighbouring cell. The remaining paths determine the lifts that are necessary for the flow from start coordinate to end coordinate. The height elevation of these paths are of the sequence $h(P_{s_0}) < h(P_{s_0+1}) < h(P_{s_0+2}) < .... < h(P_{e_0})$ and $\forall y$, $h(P_{s_y}) < h(P_{s_y+1}) < h(P_{s_y+2}) < .... < h(P_{e_y})$ where y=$\{0,1,2,....,j\}$. The sequence describes that the elevation at the current cell is always less than its subsequent neighbouring cell. The resultant flow path also contains the set of coordinates where $\langle P_{i_1}, P_{i_2},....,P_{i_{|\xi|}}\rangle = \xi \in \tau$. As a whole, the resultant path in a lift based flow model would be a combination where gravitational flow as well as lifting the water along the surface of the terrain occurs. The solution also conveys a pattern

describing the coordinates where the lift occurs. The set of coordinates $\langle P_{s_0}, P_{s_1}, P_{s_2}, ..., P_{s_j} \rangle$ describes where the lift occurs since its preceding cell's elevation is greater than or equal to that of the current cell's elevation which is less than its succeeding neighbouring cell's elevation. It also conveys from where the gravitational flow occurs as well. The set of coordinates $\langle P_{e_0}, P_{e_1}, P_{e_2}, ..., P_{e_j} \rangle$ describes where the gravitational flow occurs since its preceding cell's elevation is less than that of the current cell's elevation which is greater than its succeeding neighbouring cell's elevation.

The time complexity for processing the restricted polygons is $O(|R| * |\psi| * w)$ whereas the time complexity for finding out the path between σ_{i} and $\sigma_{\text{i+1}}$ is *O(n log n)* by using Dijkstra's algorithm with Fibonacci heaps [7]. The overall time complexity for Least Cost Route Model for Lift Based Flow is $O(|R| * |\psi| * w + nmlogn + n)$.

The Dijkstra's algorithm will work to certain extent but may not scale for very high resolution data sets as the number of computations for a single grid will take a lot of time where the number of nodes will be very large. In order for our models to perform on high resolution data sets, we will be using TerraCost algorithm [10], that is highly scalable for massive grid-based terrains. Main steps of TerraCost algorithm is as follows:

Step1: Intra-Tile Dijkstra, partition the grid into tiles and run Dijkstra from the source and the boundaries.

Step2: Sort the boundary stream created in Step1.

Step3: Inter-Tile Dijkstra, compute the least-cost paths to all the boundary vertices.

Step4: Final Dijkstra, for each tile, compute the least-cost paths to all internal points by running Dijkstra starting at the boundary points along with any internal source points.

Using the above process, the algorithm will be mostly as follows:

Algorithm 5 Lift Based Flow Path algorithm

```
1: procedure LEAST COST PATH($Terrain, Costofconstruction$)
2:     With the help of TerraCost algorithm, where the grid divided into tiles, at each point, calculate
       the cost using the function $\zeta(\rho, \omega)$ defined along with the canal boundaries defined earlier.
3:     Return the least cost path from start coordinate to end coordinate
4: end procedure
```

4.3 Parameterized Paths

In this section, we will discuss about the parameterized paths problem. We will discuss what is the worst case solution, the base case solution and how to interpret the solution from input parameters. Finally, we will discuss the trade-off between cost and distance.

4.3.1 Worst case and best case solution

Given two coordinates on a plane, multiple paths are possible. In the LCRM problem, we try to find the best possible path from S_C to E_C based on the cost function. Shortest path between two points, is a straight line is a known fact. We use this concept to describe the worst case solution. In a geodetic system, this distance is termed as euclidean distance. Considering the multiple paths that are possible, gravitational flow path is the longest distance and the euclidean distance is the shortest distance. Considering the general scenario, if we analyze the cost of the path for gravitational and euclidean paths, the cost to build the gravitational flow path will be less than the euclidean flow path. The number of lifts required for gravitational flow path will always be zero as the elevation from the center cell to its neighbouring cell will always be greater than or equal to zero, whereas, for the euclidean flow path the number of lifts will always be greater than or equal to zero. If the number of lifts in the resultant flow path is considered as a basis for worst case scenario, the euclidean flow path would be the worst case solution as it contains the maximum number of lifts. One may argue that euclidean need not be the worst case as one of the worst case scenarios is visiting all the nodes of the terrain. However, to build a canal certain guidelines needs to be followed. One of the main guidelines is to avoid meandering wherever applicable. In this worst case scenario, flow path length is the shortest of all possible resultant flow paths. Similarly, the gravitational flow path would be the best case scenario as the number of lifts is zero in the gravitational flow path. However, the length covered by the gravitational flow path would be the largest.

4.3.2 Interpretation from Worst case and Best case scenarios

From the worst case and best case scenarios explained earlier, in terms of number of lifts, euclidean flow path is the worst case and the gravitational flow path is the best case scenarios. We will now discuss the problem of identifying the different paths given either the number of lifts or the distance covered by the resultant paths. As explained earlier, the resultant flow path in any flow would be of sequence $\langle P_0, P_1, P_2, \ldots, P_k \rangle$. Let the resultant path be an euclidean path from S_C to E_C. Let P_{eu} be one of the points of the euclidean resultant path. Let P_{eu-1} be the predecessor and P_{eu+1} be its succeeding paths of P_{eu}. If there exists a point P_{eu} such $h(P_{eu-1}) < h(P_{eu})$, then at least one lift is required to lift the water along the surface of the terrain. If no such point exists, then the euclidean flow path is the gravitational flow path as the elevation of predecessor cell's is greater than or equal to the center cell's elevation which is grater than or equal to its succeeding neighbour's cell elevation. The possibility of reducing a lift can be achieved if the flow from $P_{eu\text{-}1}$ can be diverted to one of its neighbours whose elevation is less than or equal to the current cell i.e., $h(P_{eu\text{-}1}) > h({P_{eu\text{-}1}}^n)$; where ${P_{eu\text{-}1}}^n$ is one of the neighbours of $P_{eu\text{-}1}$ and $n \neq eu - 1$. If $h(P_{eu\text{-}1}) \geq h({P_{eu\text{-}1}}^n)$, then a possibility of gravitational path exists. Let *k* be the number of cells along the resultant path from S_C to E_C.

4.3.3 Trade-off between cost and distance

We have discussed two models that tries to determine the least-cost path. If we refine the problem a bit more, given a cost parameter, are there a set of paths that are possible for the mentioned cost and vice-versa.

Consider the use cases shown in Figure 4.5. Let

- G be the starting point
- H an intermediate point along the Euclidean path
- D be the destination point
- k-n be the kth n point with distance k-n from the intermediate point H
- k-max be the threshold point from the intermediate point H where the flow path cannot exceed the triangular route of G, D and k-max.

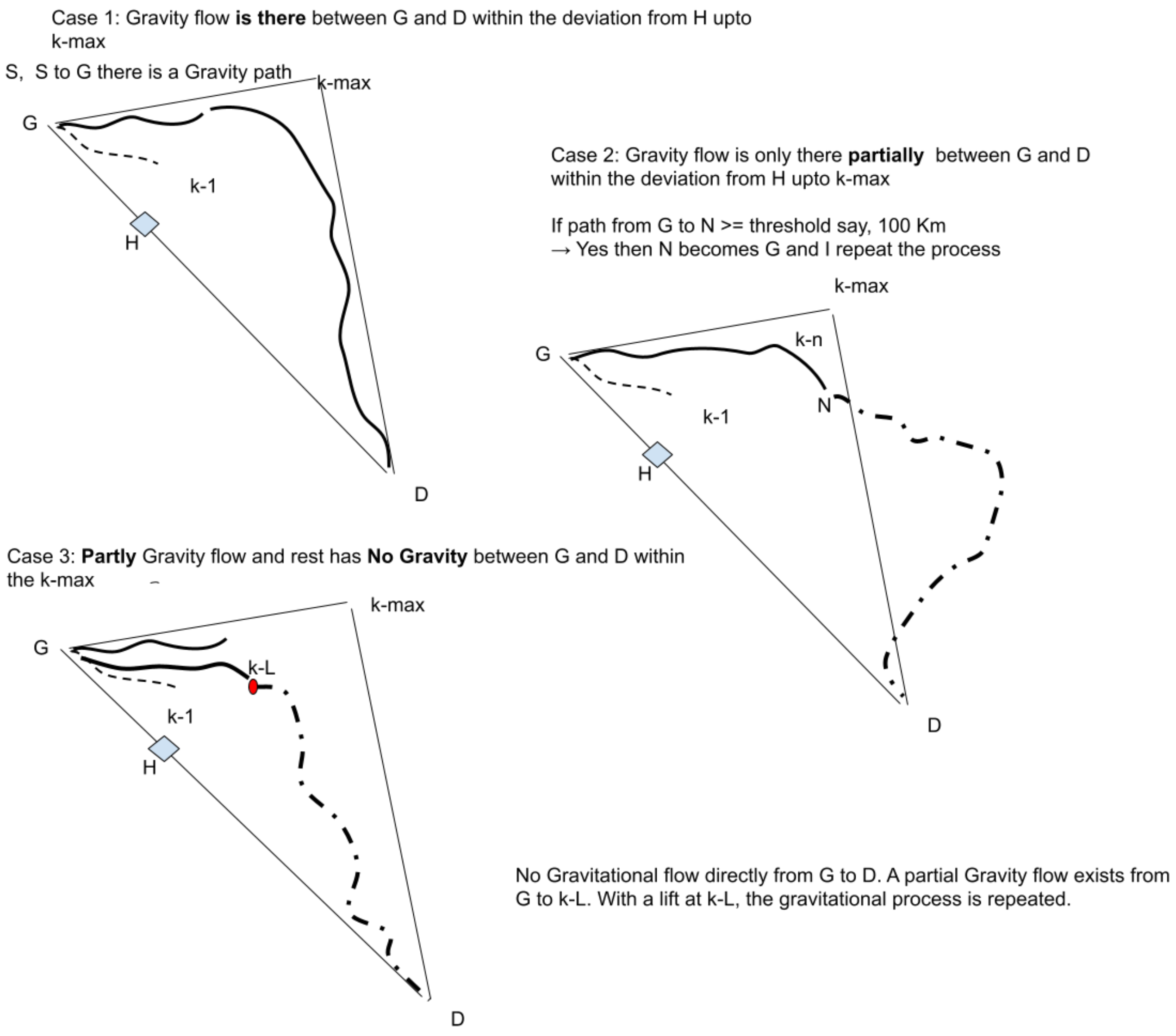
There are three different use cases identified. They are explained as follows:

- **Case-1:** Gravitational flow is present between G and D. A gravitational path is within the bounds of G, D and k-max. So, one of the bounds will be the gravitational flow path. We will discuss the bound as we move on.
- **Case-2:** Gravitational flow is partially present between G and D, meaning the gravitational path exceeds the k-max(threshold) point. Let N be the point along the gravitational path within the bounds of k-max. We use the gravitational path from G to N to identify the remaining paths for this use case. We try to find another gravitational path from N to D. This process is repeated until a path is found.
- **Case-3:** Partly gravitational flow exists between G and D within the k-max point. In this case, we use the partly gravitational flow. Let k-L be the point where the first point of the lift is required. We will use the water lifting along the surface approach to get to the destination point. At k-L, lift the water along the surface and then the same gravitational flow path algorithm is repeated. This procedure is applied throughout until a solution is found.

We define a function $f(g_c + l_c)$ where g_c is the gravitational path cost and l_c is the lift-based path cost. The problem we solved earlier is the case where $f(g_c + l_c)$ is minimum. For any given two points, Euclidean path is the shortest path. So, we will be using that principle to determine the set of paths. But before that, we need to set the lower and the upper boundaries.

- *f*($g_c + l_c$) is maximum when the path is a Euclidean path considering the canal boundaries.
- *f*($g_c + l_c$) is minimum when the path either has a gravitational path or has the least number of lifts considering the canal boundaries.

In other terms, we defined the above statements as, if the path cost is maximum, then the length of the canal is the shortest and has a high number of lifts. Similarly, if the path cost is minimum, then either the path is a gravitational path or has the least amount of lifts. So, within these bounds, cost ranging from minimum to maximum, there could be numerous paths that are possible. We will identify these sets of paths and plot it on a graph.



Case 1: Gravity flow **is there** between G and D within the deviation from H upto k-max

S, S to G there is a Gravity path

Case 2: Gravity flow is only there **partially** between G and D within the deviation from H upto k-max

If path from G to N >= threshold say, 100 Km
→ Yes then N becomes G and I repeat the process

Case 3: **Partly** Gravity flow and rest has **No Gravity** between G and D within the k-max

No Gravitational flow directly from G to D. A partial Gravity flow exists from G to k-L. With a lift at k-L, the gravitational process is repeated.

Figure 4.5 Use-cases to determine possible paths

Chapter 5

Data and Results

In this section, we first describe the data which is used for terrain analysis. Then, we will demonstrate the usage of algorithm by using an example which helps in understanding of the flow of algorithm. We then proceed on to the environment settings that was used to run the data that was specified earlier. Followed by applying the proposed algorithms on the real-world data sets that was described in the data section. And finally we will discuss the results after applying the proposed algorithms on the data sets.

5.1 Data

There are multiple different types of terrain data models. They are Digital Elevation Model (DEM), Digital Terrain Model (DTM), Digital Surface Model (DSM) and Triangulated Irregular Network (TIN). In a DEM data, each cell represents an elevation value of the surface above the sea level. DTM describes not only the elevation but also the geographical elements. DSM contains the elevation which includes the surface objects such as trees, buildings etc. TIN is a 3D surface model derived from irregularly spaced points and break line features. Since multiple types of data models are present, we will pick up DEM data model as it has data that represents only the elevation data with respect to the sea level. DEM data model contains different types of data sets and data resolution. A data resolution is a measure used to determine the size of each cell. DEM data models contains multiple resolutions ranging from 1 meter to several Kilometers. Multiple sources are available related to DEM data models; they are SRTM, ASTER, LiDAR and Bhuvan. For the ease of understanding, we have used the publicly available data sets. We used two resolutions of data in our experiments. The first data is of resolution 1KM belonging to Indian terrain covering roughly 6000KM by 4800KM (data courtesy of USGS). The terrain model consists of about 28.8 million grid points having a wide variety of topologies such as mountains, riverbeds, flatlands, highlands, etc. This data contains the elevation ranges from 0 to 7804. The second data is a higher-resolution 90-meter grid covering a 540KM by 540KM region in Indian terrain (data courtesy of SRTM). The data has 36 million grid points. Multiple data sets of the 90-meter resolution were considered having unique topology.

5.2 Demonstration using an example

Considering Figure 5.1 as an example where each cell value represents the elevation, we need to identify the least cost-path from (0,0) to (4,4). From (0,0), there are only2 possibilities for gravitational path to (0,1) and (1,0). And from there on, the current cell elevation is lower than its neighbours. In this case, we must use a lift to solve this example. So, lifts can be used at multiple points, (0,2), (1,1) and (2,0). And from then on, the gravitational path exists from (0,2), (1,1) and (2,0). To reach (2,2), 1 more lift is required in any case, as the elevation of that cell is higher than its neighbours. This process will be repeated until the (4,4) cell is reached. So, the least-path would be from (0,0) –> (1,0) –> (2,1) –> (3,2) –> (3,3) –> (4,4)

5.3 Environment Settings

To run our experiments, we have considered the following environment settings. The implementation of both of our algorithms in C++, using the GNU g++ compiler, version 5.4.0. The experiments that we conducted were run on a workstation with an Intel Core i5-2450 CPU. This is a four-core processor with 2.50GHz per core, and the main memory of this computer is 7.6 Gigabytes. Our implementations run on a Linux Ubuntu operating system, release 16.04.

5.4 Applying algorithms on real-world data sets

We have implemented the algorithms in two phases. The first phase is Unconstrained flow and the other being constrained flow. We will discuss the procedure that was followed while applying the proposed algorithms on real world data sets.

5.4.1 Unconstrained flow

In the unconstrained flow, initially Gravitational Flow Model algorithm is executed on data sets mentioned earlier for inputs $\mathrm{S_C}$, $\mathrm{E_C}$ and γ. For the same set of inputs, algorithms are run on different resolutions. The data which was considered has various start and end coordinates containing different topologies, as shown in Table 5.2. Distinct $\mathrm{S_C}$ and $\mathrm{E_C}$ values are considered ranging from a Euclidean distance of 25KMs to 300KMs. The input for the gravitational flow algorithm is applied where the elevation difference is always positive, i.e., $\Delta H(S_C, E_C) > 0$. If a gravitational flow path can be found from $\mathrm{S_C}$ to $\mathrm{E_C}$, the result would be a sequence of points on how the flow needs to be built. Following this, Lift Based Flow Model algorithm is executed for the same data sets that was mentioned earlier with the addition of cost of lift parameter that can raise the water up to a lift of 10 meters is C_l. Lift Based Flow model is applied when Gravitational Flow Path doesn't exist or when the cost of construction is too high to built for the gravitational flow. Based on these conditions, Lift Based Flow model is applied.

100	99	101	97	100
100	102	102	98	99
101	104	105	96	101
98	95	101	95	94
97	96	96	93	92

Figure 5.1 Elevation profile data example

0	1	$2 + k$	$3 + k$	$4 + 2k$
1	$\sqrt{2} + k$	$1 + \sqrt{2} + k$	$2 + \sqrt{2} + k$	$3 + \sqrt{2} + 2k$
$2 + k$	$1 + \sqrt{2} + k$	$2\sqrt{2} + 2k$	$1 + 2\sqrt{2} + k$	$2 + 2\sqrt{2} + 2k$
$3 + k$	$2 + \sqrt{2} + k$	$1 + 2\sqrt{2} + k$	$2 + 2\sqrt{2} + k$	$1 + 3\sqrt{2} + k$
$4 + k$	$3 + \sqrt{2} + k$	$2 + 2\sqrt{2} + k$	$3 + 2\sqrt{2} + k$	$2 + 3\sqrt{2} + k$

Figure 5.2 Cost analysis for the elevation profile

The cost to move from C_C to N_C, $\zeta(C_C, N_C)$ is calculated at each step. The lift elevation cost, which is applicable only for lift based flow, is calculated when applicable, i.e., when $h(C_C) < h(N_C)$. The cumulative cost is calculated at each subsequent point. Based on the cumulative path cost, an optimal solution is considered. Table 5.2 shows the results of applying both the gravitational flow model and lift based flow model algorithms.

5.4.2 Constrained flow

In the constrained flow, additional parameters, set of pass through coordinates ξ and restricted polygonal regions ψ are taken into account. Instead of random values for these additional parameters, based on an analysis, distinct values of ξ are based on some coordinates along the hillshade path of the terrain from start start coordinate to end coordinate. The polygonal areas, ψ are considered based on the land use data. We have considered the commercial and residential regions, ψ for this data. First, Gravitational Flow Model algorithm is executed for inputs (S_C, E_C, γ, C_l, ξ, ψ). The Gravitational Flow Model algorithm is applied for each of the coordinates σ_i and σ_{i+1}. The execution of the algorithm will be skipped when elevation difference from σ_i and σ_{i+1} is higher; i.e., $h(\sigma_i) < h(\sigma_{i+1})$. For the same set of inputs, Lift Based Flow Model algorithm is applied. As mentioned, this algorithm is applied for each consecutive pair of coordinates σ_i and σ_{i+1}. The resultant path would be a sequence of points on how the flow needs to be built containing the pass through coordinates for both these flows. The results on applying both the algorithms can be viewed in Table 5.2.

5.5 Results

Table 5.1 describes the different regions that are considered for evaluating the algorithm where OC represents the number of overlapped cells visited with respect to DCW data along the resultant path; RC represents the number of cells of the DCW data along the path. Figure 5.3 describes the hill shade area of the NSP Right Canal, the bounded region represents the NSP Right Canal area that we will discuss in detail. Figure 1.1 describes the origination of the NSP. Figure 5.4 describes the height vs distance profile of the NSP Right Canal for the Euclidean path. Whereas Figure 5.5 describes the height vs distance profile of the NSP Right Canal on applying the proposed algorithm.

By considering the Euclidean path as the canal path, the distance travelled will be minimum but the cost may vary depending on the elevations that the flow passes along this Euclidean path. So, by considering the Euclidean path, the distance travelled will be minimum but the cost will be high as the height variation is irregular as can be seen in Figure 5.4. Considering the elevations in Figure 5.4, 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. In order to avoid high costs for canal construction, we have proposed the gravitational flow model algorithm. So, on applying the gravitational flow model algorithm, the distance may be a bit higher but the cost of construction would be minimal as the flow of water will always be gravitational force and the elevation profile will be such that it supports gravitational flow. For the use case defined earlier, the elevation values ranges from 164 to 125 as can be seen in Figure 5.4. In the case of the Euclidean path, the elevation along the path is irregular which requires elevation clearance for gravitational flow resulting in expensive canal construction. The total distance covered along the path is around 20 KM. In case of applying the algorithm, the elevation along the path is smooth where the center cell's elevation is greater than or equal to its neighboring cell as can be seen in Figure 5.5. The total distance covered along the path is around 23 KM.

Table 5.2 reports the results obtained by running the algorithms described above. The first six columns present our main parameters: the start coordinate $\mathrm{S_C}$, the end coordinate $\mathrm{E_C}$, the cost of construction per unit length γ is 100 units, the cost to raise the water along the surface of the terrain say 10 meters is C_l is 2500 units, the collection of coordinates to pass through ξ and the polygonal areas to evade ψ. The next two columns present the number of cells in the optimal resultant flow path overlapped with respect to reference data, i.e., DCW in the resultant path. Followed by the two columns representing the number of lifts from $\mathrm{S_C}$ to $\mathrm{E_C}$ and followed by the next two columns presenting the number of cells of the reference data, i.e., DCW along the path from $\mathrm{S_C}$ to $\mathrm{E_C}$. Finally, the two columns representing the accuracy of applying algorithms for the set of specified parameters. For each of the input, the results for both 1KM resolution as well as 90 meters resolution are published. The results are combinations of both Gravitational Flow Model as well as Lift Based Flow Model.

Figure 5.6 and 5.10 demonstrates the results on applying gravitational flow path algorithm on some of the parameters defined for 1KM resolution data set, whereas Figure 5.8 and 5.12 represents the result for the same set of parameters but on 90 meters resolution data set. They denote the resultant flow path on applying the gravitational flow algorithm from $\mathrm{S_C}$ to $\mathrm{E_C}$. Figure 5.14 demonstrates the results on applying both gravitational flow path algorithm and lift based flow path algorithm for one of the parameters defined on the 1KM resolution data set. It denotes the resultant flow path on applying the algorithm from $\mathrm{S_C}$ to $\mathrm{E_C}$. It also describes the point where the shade of green denoting that lift is necessary to achieve the least-cost path. Figure 5.16 represents the result for the same set of parameters but on 90 meters resolution data set.

Figure 5.18 to 5.24 demonstrates the results of the trade-off problem between two parameters i.e., lifts and distance. The blue line indicates the minimum values and the orange line indicates the maximum values. So, for each of the result, the lift and distance graph varies. As the distance increases, the number of lifts required is decreasing.

5.6 Experimental Evaluation

This section is mostly about the analysis based on the results from the previous section. First, we discuss the main aspect of the algorithm, which is running time and the algorithm comparison with

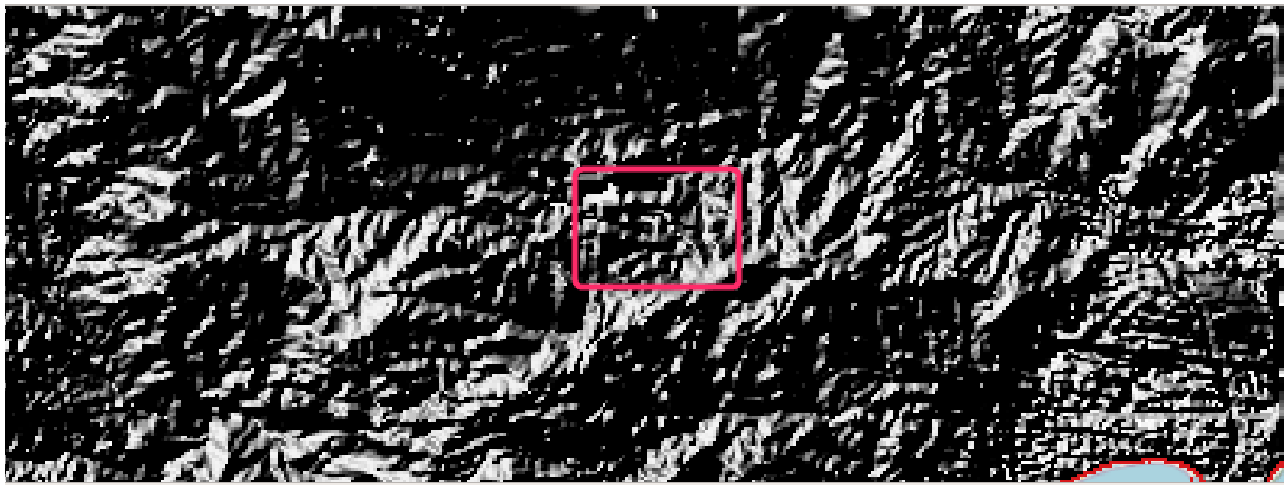


Figure 5.3 Hill shade of the study area

the existing least cost for canal model algorithms. Later we discuss the correctness of the algorithm, describing what is the data and why is the data needed for comparison and how the resultant data will be compared with the proposed algorithm results.

5.6.1 Evaluating algorithms

As described earlier, the time complexity of gravitational flow path is $O(|R| * |\psi| * w + nmlogn + n)$. Where n is the number of cells of the terrain τ, $|R|$ is the total number of rows in the terrain τ, w is the average number of points of all polygons ψ, m is $|\xi| + 1$ and $m \ll n$. On different data sets taken, the total execution time of the gravitational flow algorithm is from 6 minutes to 8 minutes. Around 5 minutes 30 seconds of the time taken is consumed by I/O stream because of the massive grid size. The remaining time is for the model algorithm to run. The time complexity of lift based flow path is $O(|R| * |\psi| * w + nmlogn + n)$. On different data sets taken, the running time of the lift based flow algorithm is from 6 minutes to 10 minutes. The execution time values are based on the Environment Settings specified earlier. As compared to the existing algorithm proposed by Walter et al. [1], which is of $O(kn^2)$, where k is a number of repeated iterations to compute the result and is proportional to the path complexity. Our proposed solution for gravitational flow is computationally better, which is linearithmetic compared to polynomial order. So, from the time complexity point of view, both our algorithms perform well when compared with Walter et al. [1]. Walter's solution avoid higher slopes i.e, if a neighbouring cell's elevation is higher than the center cell's elevation, the algorithm doesn't consider the path. In short terms, if $h(C_C) < h(N_C)$, Walter's algorithm stops finding path in that particular direction. So, Walter's algorithm is just restricted to gravitational flow model only. Based on this, Walter's algorithm can only be compared to our Gravitational Flow Model algorithm. Since Walter's study is evaluated on the synthetic data sets instead of real-world data sets, it is hard to provide

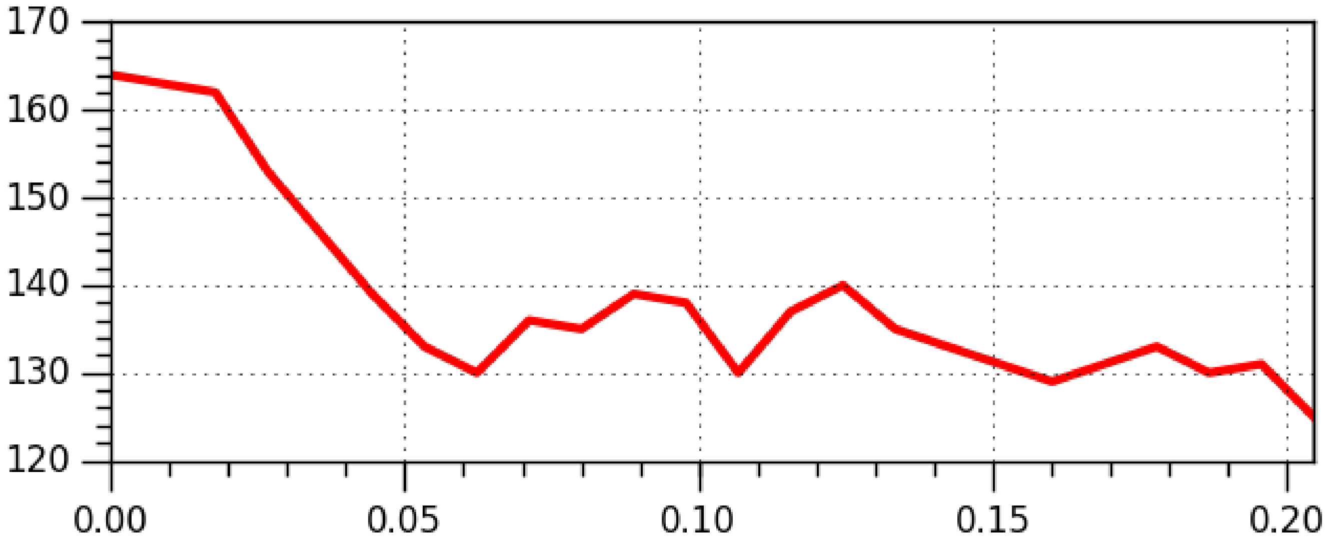


Figure 5.4 Euclidean path elevation profile for the study area

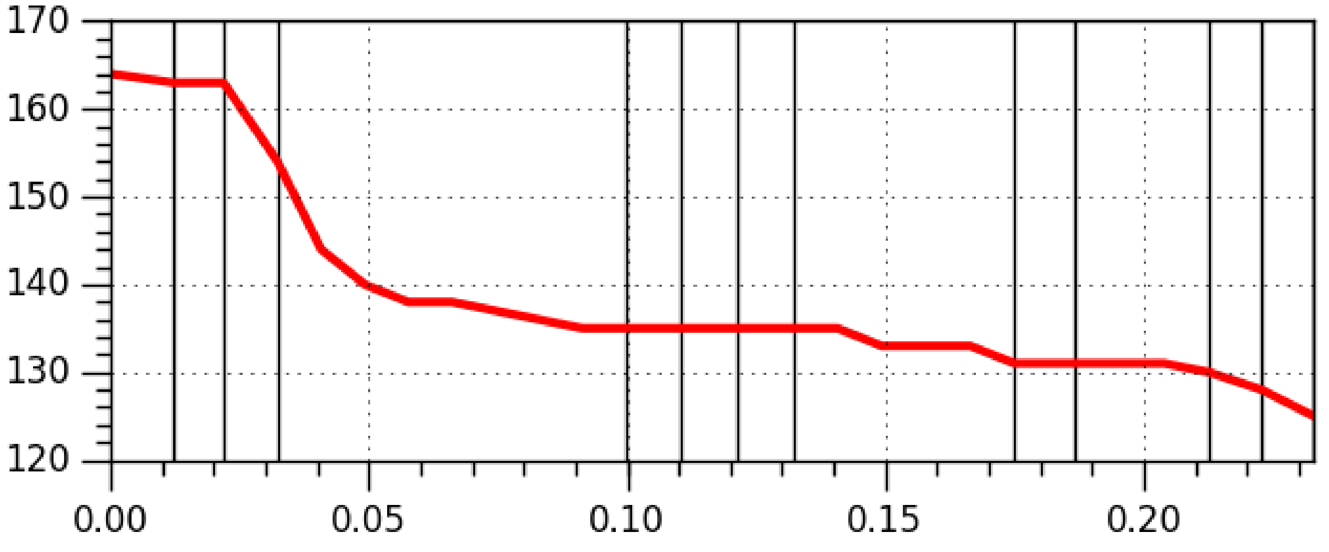


Figure 5.5 Gravitational path elevation profile for the study area

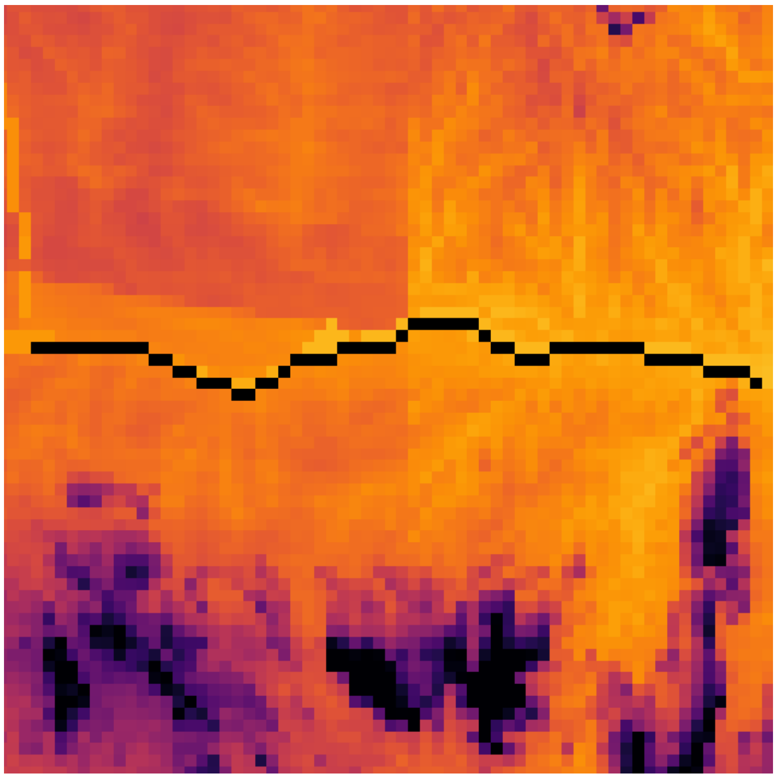


Figure 5.6 Case1: Resultant path on 1KM resolution

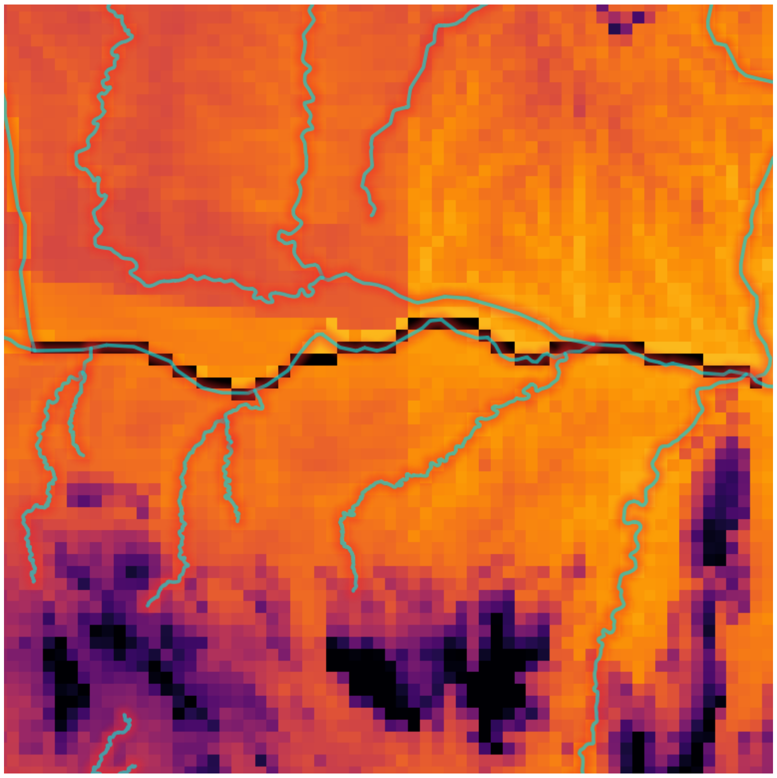


Figure 5.7 Case1: Comparison with DCW data on 1KM resolution

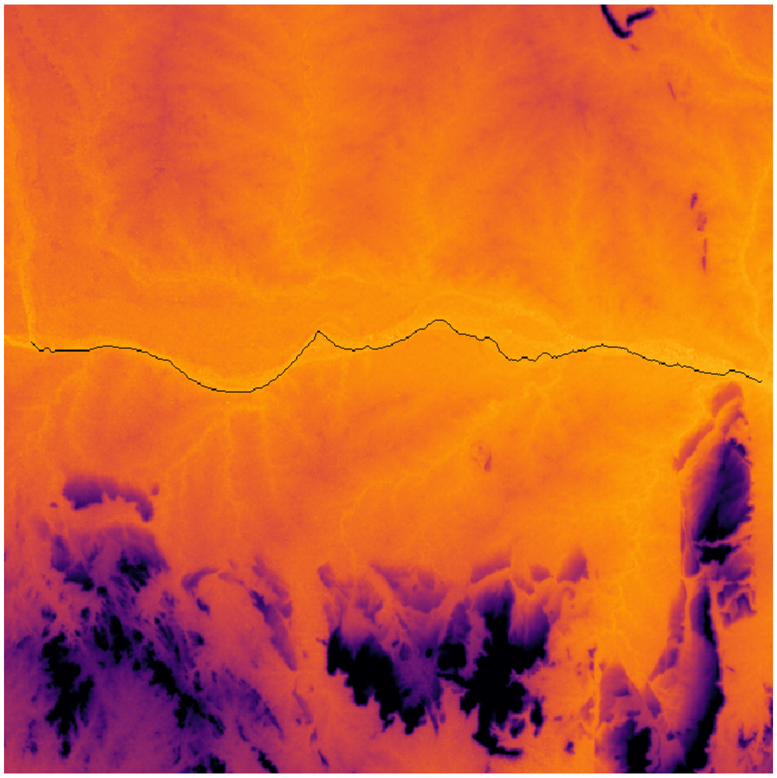


Figure 5.8 Case1: Resultant path on 90 meters resolution

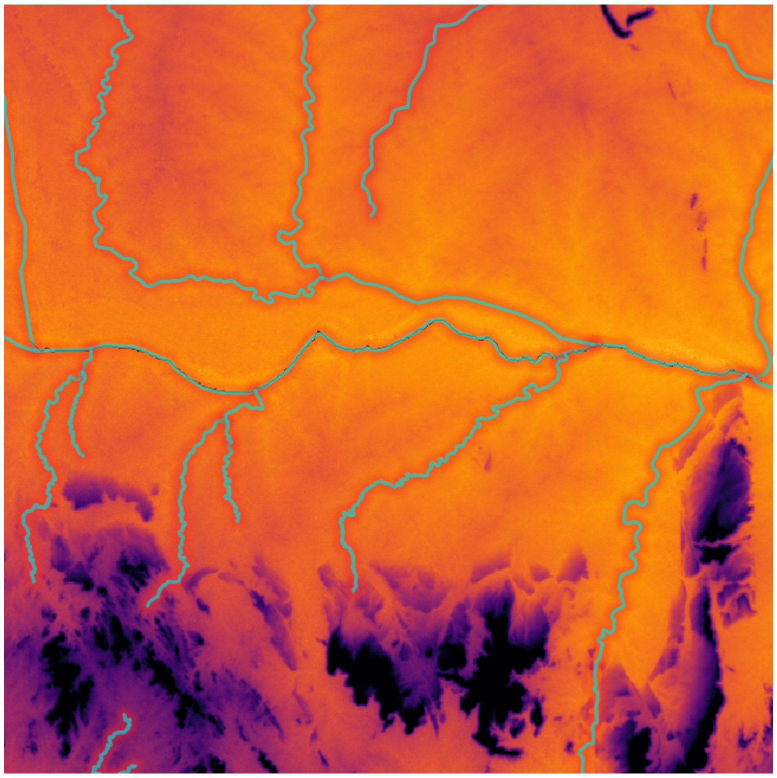


Figure 5.9 Case1: Comparison with DCW data on 90 meters resolution

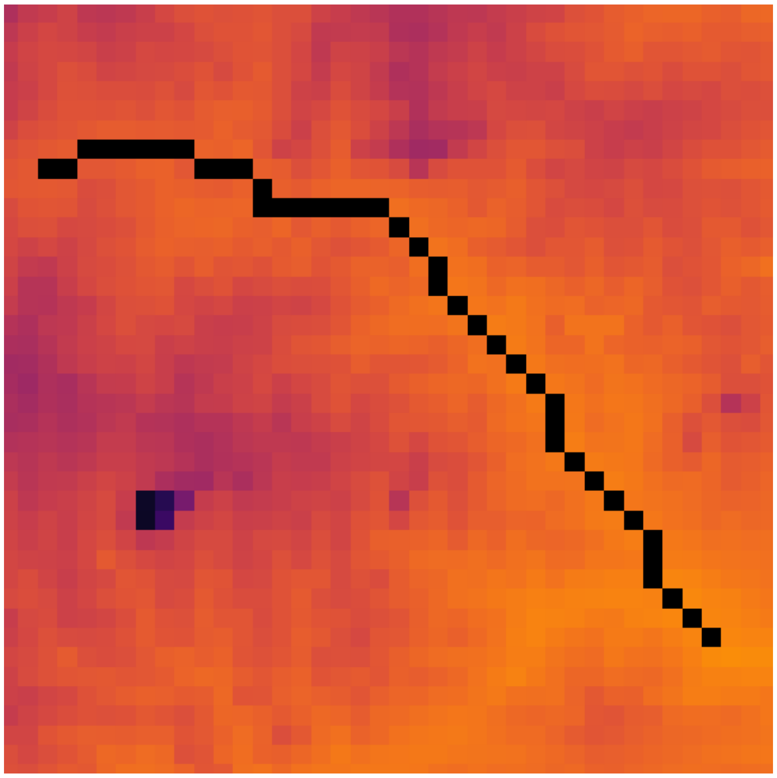


Figure 5.10 Case2: Resultant path on 1KM resolution

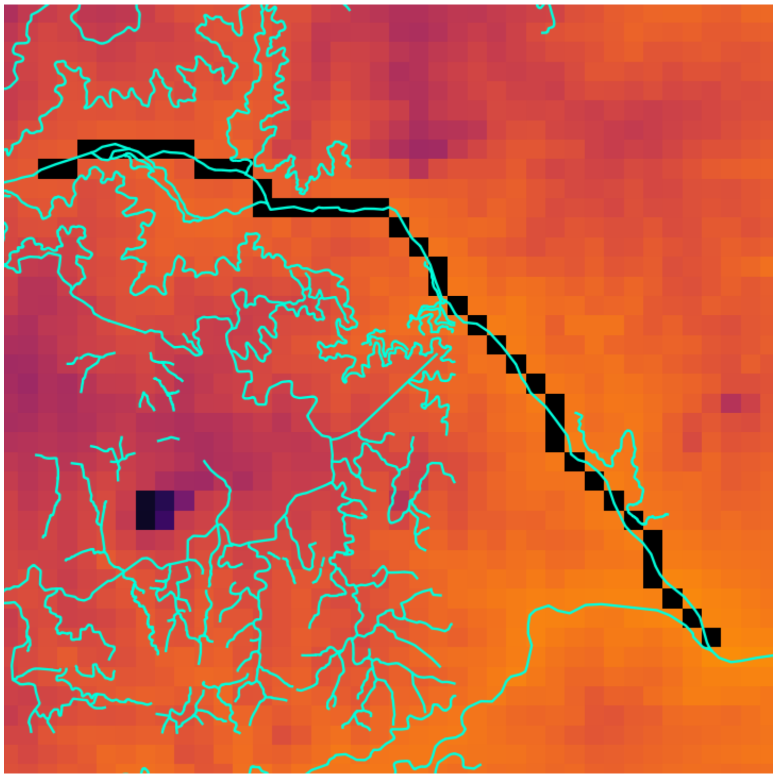


Figure 5.11 Case2: Comparison with DCW data on 1KM resolution

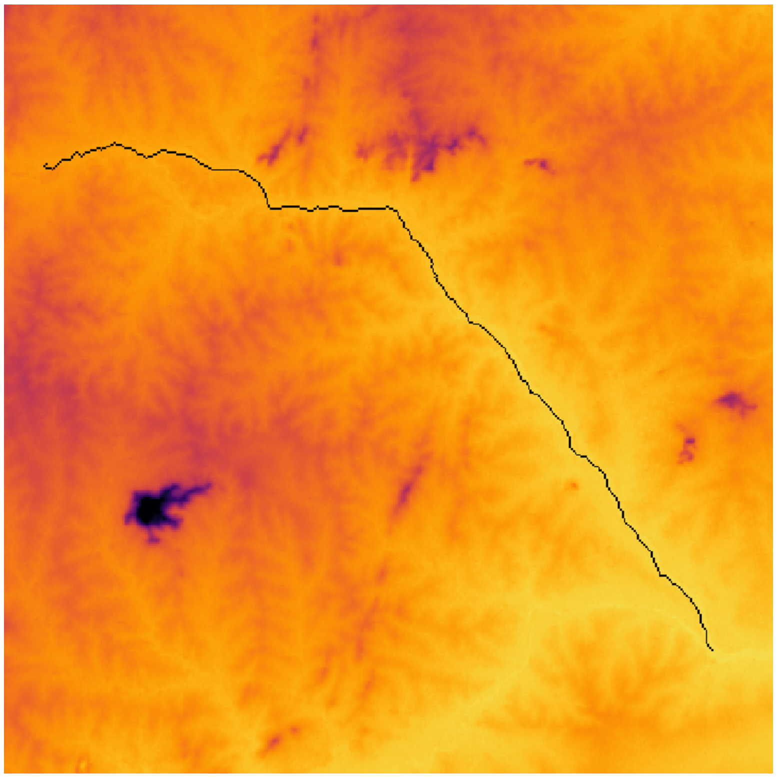


Figure 5.12 Case2: Resultant path on 90 meters resolution

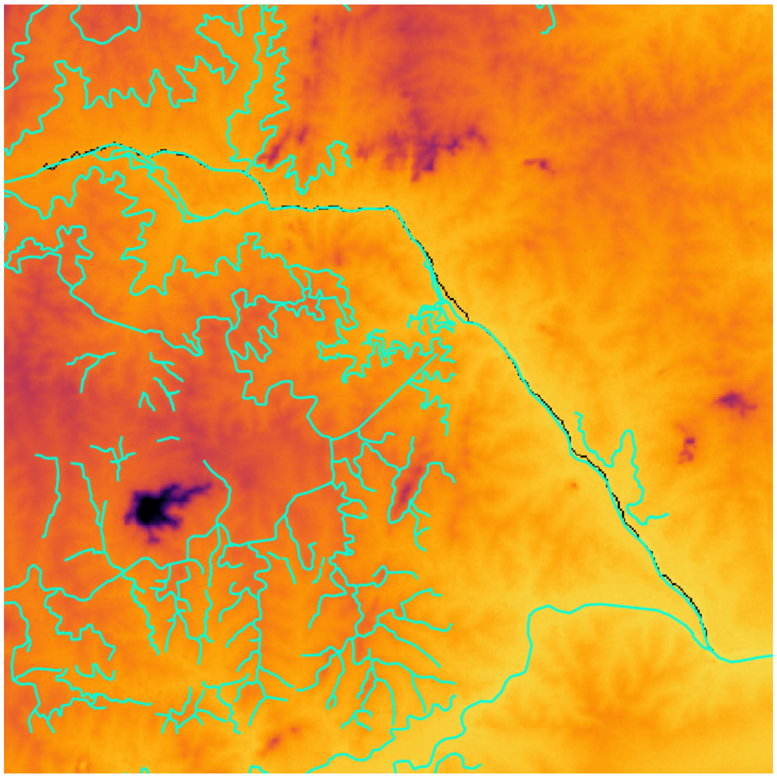


Figure 5.13 Case2: Comparison with DCW data on 90 meters resolution

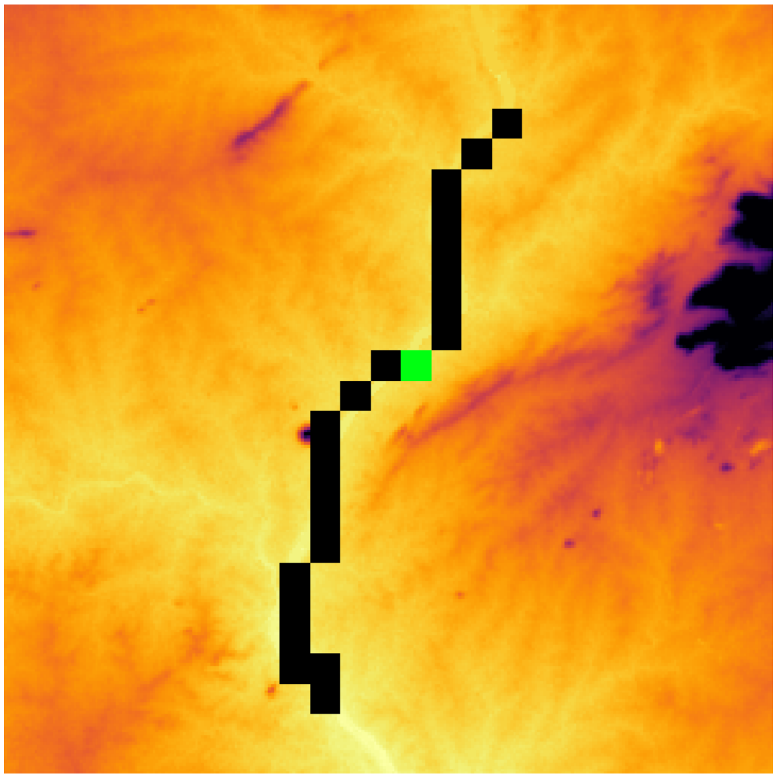


Figure 5.14 Case3: Resultant path on 1KM resolution

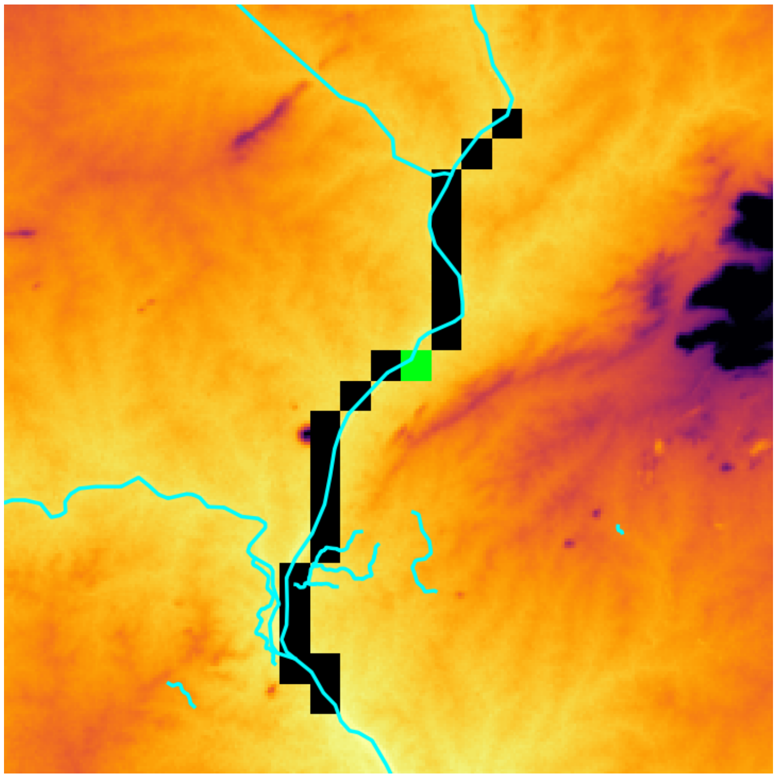


Figure 5.15 Case3: Comparison with DCW data on 1KM resolution

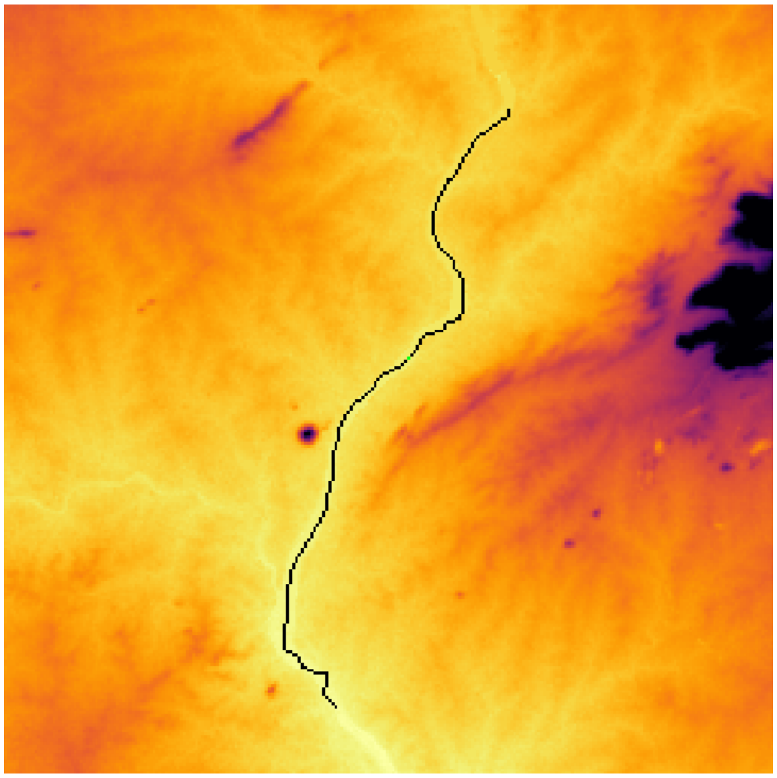


Figure 5.16 Case3: Resultant path on 90 meters resolution

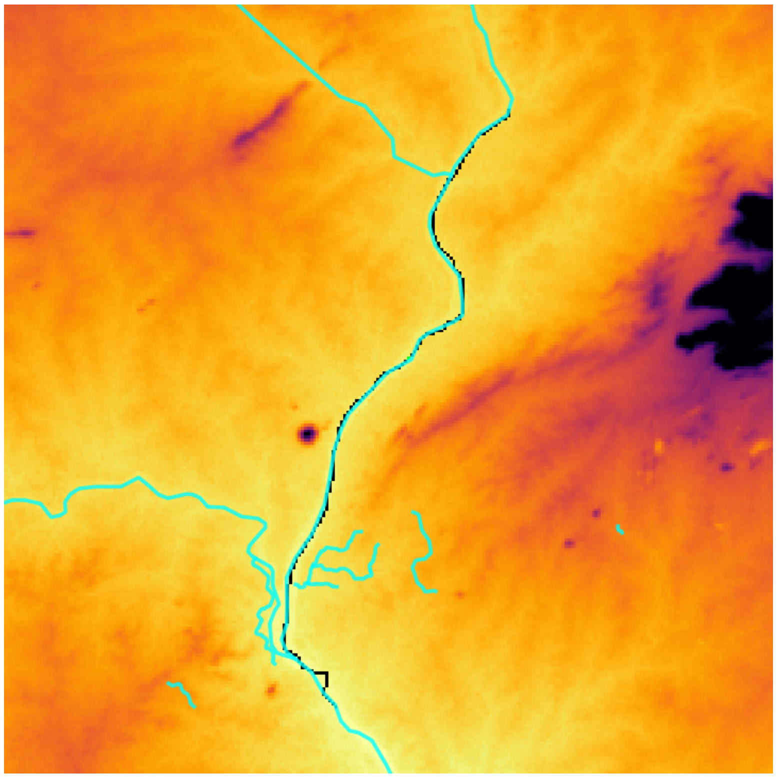


Figure 5.17 Case3: Comparison with DCW data on 90 meters resolution

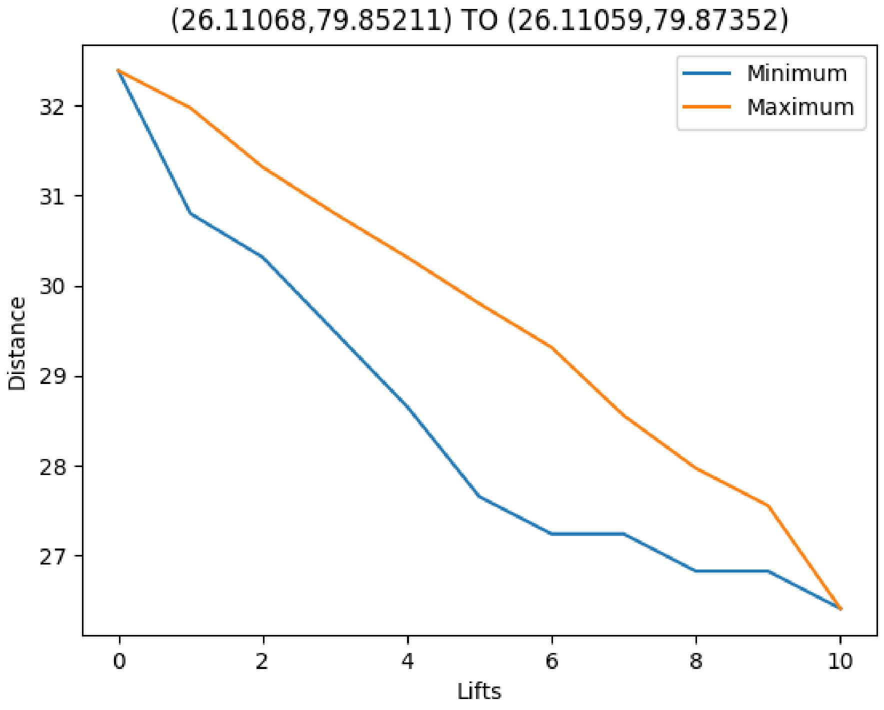


Figure 5.18 Case1: Trade-off graph

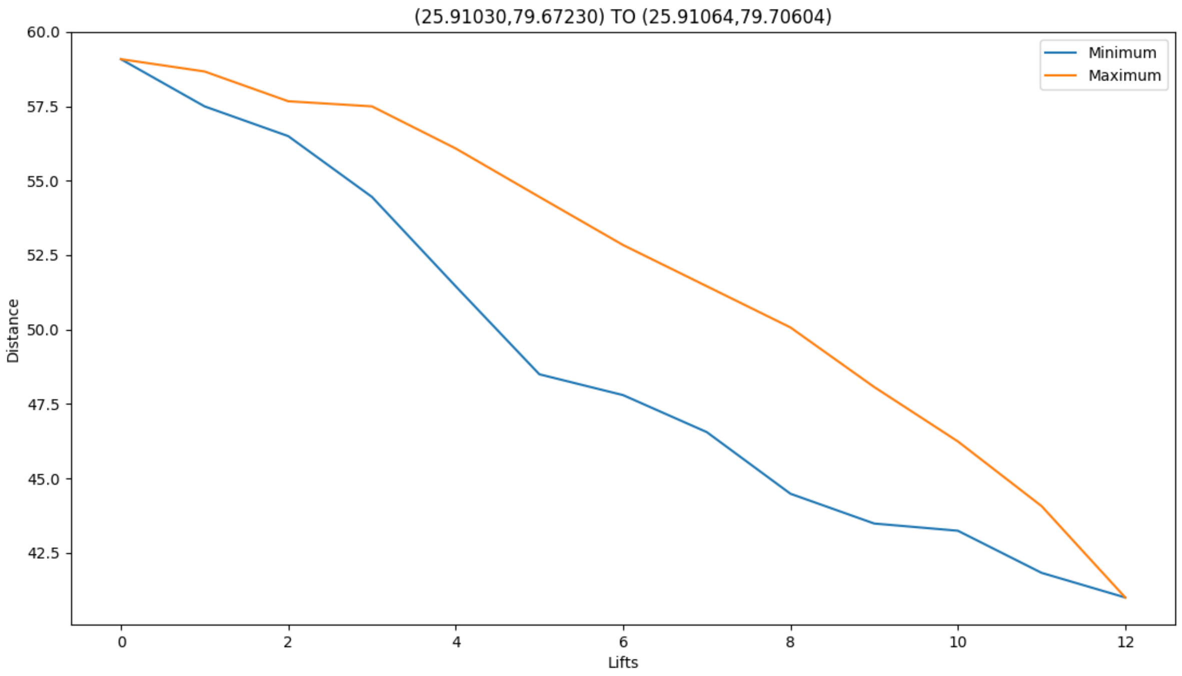


Figure 5.19 Case2: Trade-off graph

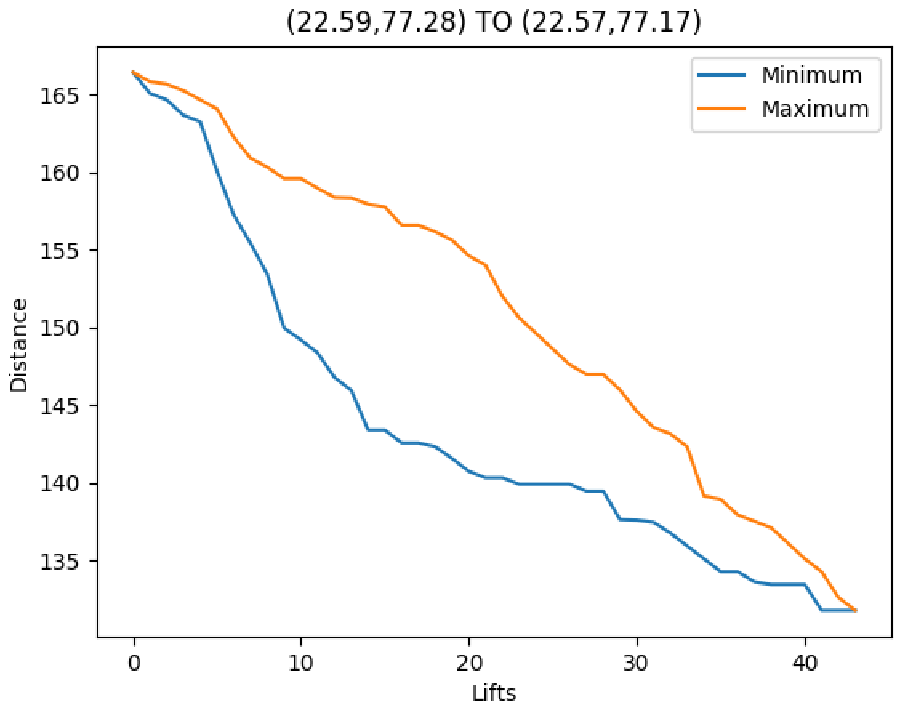


Figure 5.20 Case3: Trade-off graph

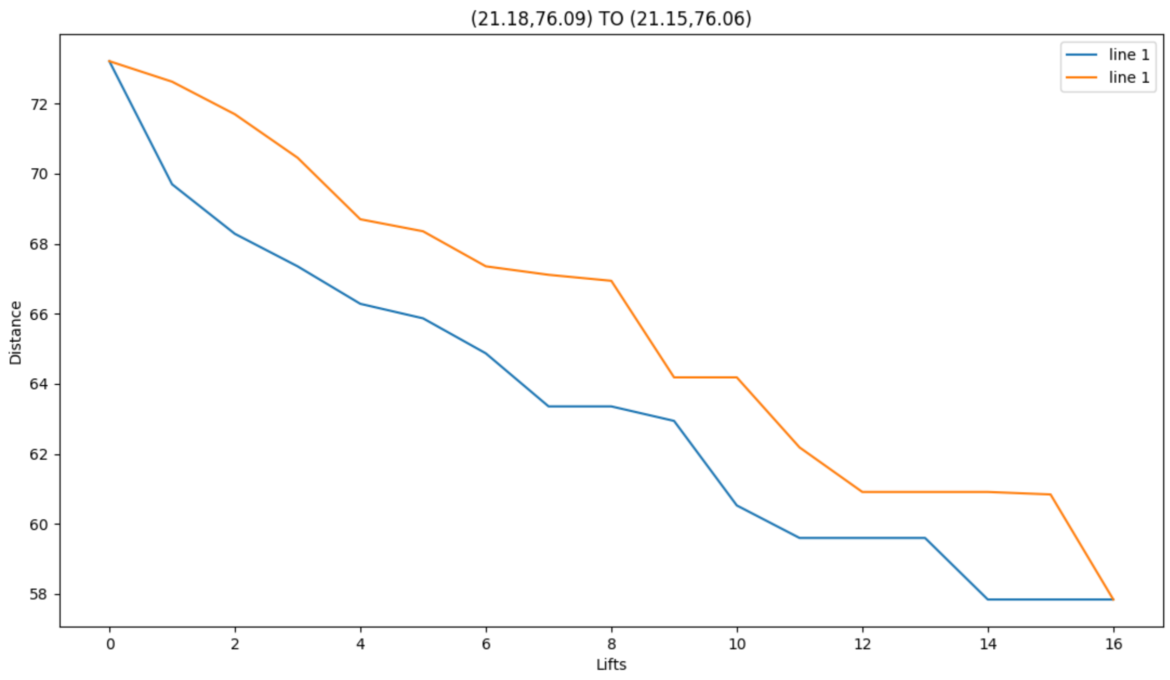


Figure 5.21 Case4: Trade-off graph

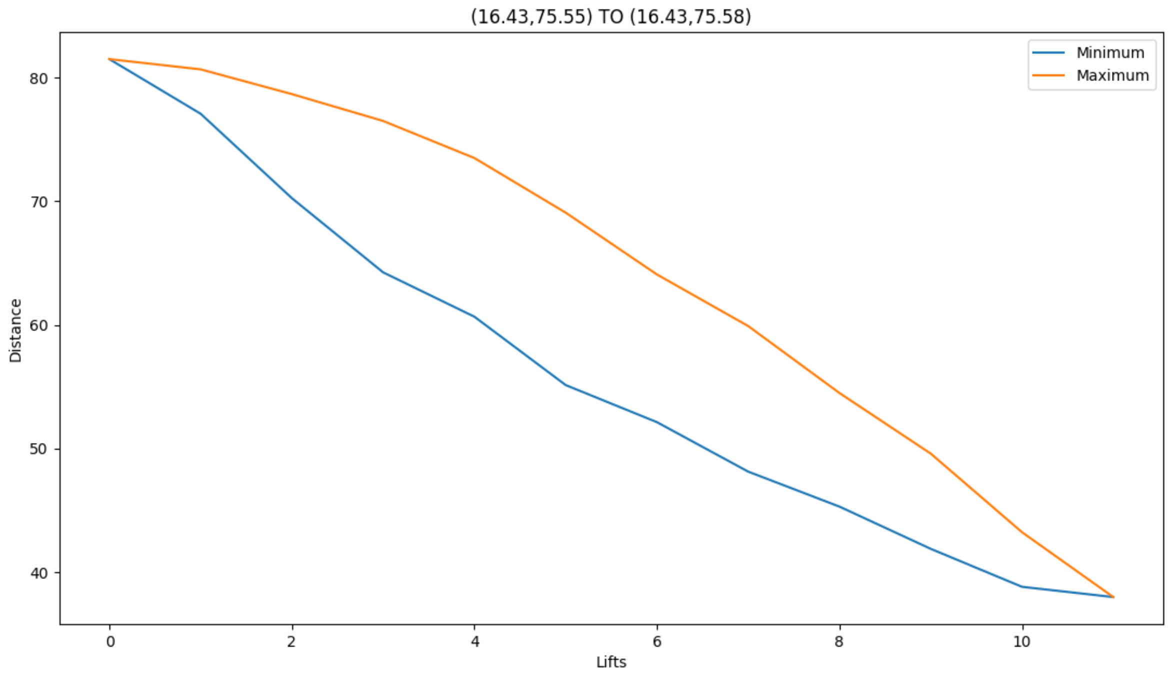


Figure 5.22 Case5: Trade-off graph

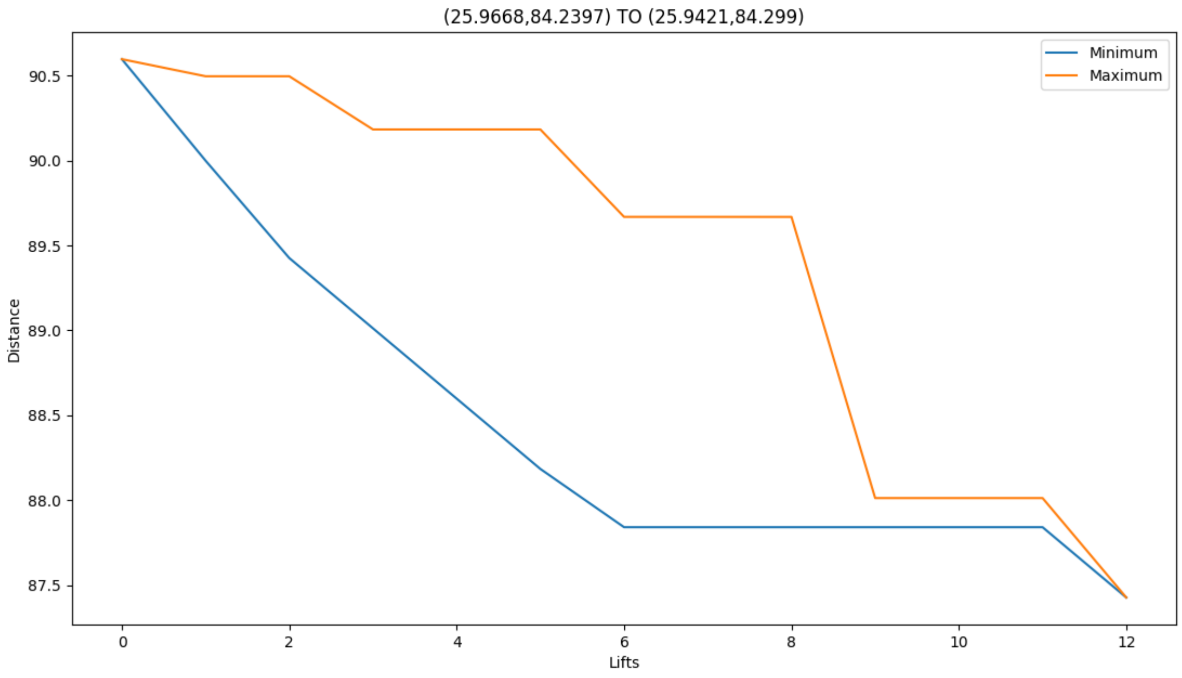


Figure 5.23 Case6: Trade-off graph

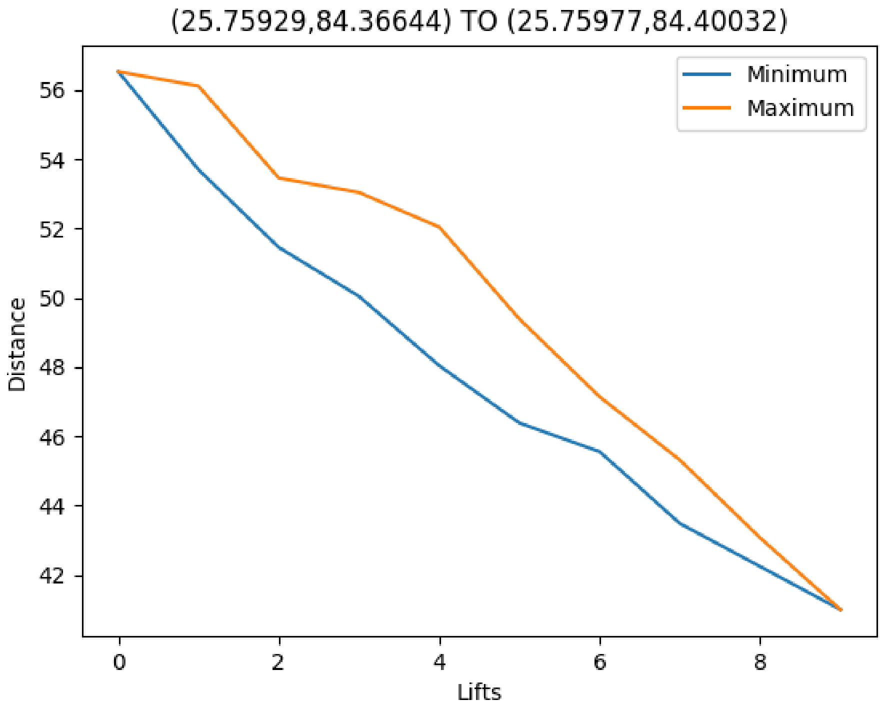


Figure 5.24 Case7: Trade-off graph

an honest comparison against our technique. In addition to that, the factors such as datasets, overhead, and preprocessing weren't described while developing the algorithm. So, it is even hard to compare the algorithms computationally. Due to multiple reasons mentioned above, we do not compare our algorithm against Walter et al. [1].

5.6.2 Algorithm correctness

In order to verify the correctness, the results need to be verified on a valid reference data set. This section discusses the reference data that was used for comparison and proceed onto the comparison of our results against the reference data set.

5.6.2.1 DCW Data

For the comparison of the results, we use Digital Chart of the World, DCW data. DCW is a comprehensive digital map of Earth which is freely available and is a standard reference for the real-world data set generated at 1KM using a vector contour-to-grid approach. 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 was produced by USGS. The data contains different thematic layers, of which we are most interested in the hydrography and drainage system.

5.6.2.2 Comparing resultant data with DCW data

As the data is a vector line data, which is not quite a replica of the real-world data. To replicate the real-world data, we have used the concept of a buffer zone [5] replicating the real-world data. A buffer is useful for proximity analysis. The buffer distance is based on the resolution of the input data that we have considered. For a 1KM resolution data, we have used a buffer of 100 meters, whereas, for a 90 meters resolution data, we have used a buffer of 9 meters. In short, 10% of the resolution of the data is considered as a buffer zone. By including the 10% buffer zone, the vector contour data is converted to a raster data of different resolutions. For our problem, as resolutions of 90 meters and 1 Kilometer are considered, we will be comparing our resultant data with its DCW resolution data sets.

Figure 5.7 and 5.11 demonstrates the results on applying gravitational flow path algorithm on some of the parameters defined for 1KM resolution data set in addition to the overlap of DCW data, whereas Figure 5.9 and 5.13 represents the result for the same set of parameters but on 90 meters resolution data set in addition to the overlap of DCW data. They denote the resultant flow path on applying the gravitational flow algorithm from S_C to E_C. Figure 5.15 demonstrates the results on applying both gravitational flow path algorithm and lift based flow path algorithm for one of the parameters defined on the 1KM resolution data set. It denotes the resultant flow path on applying the algorithm from S_C to E_C. It also describes the point where the shade of green denoting that lift is necessary to achieve

Table 5.1 Different regions of study areas

Region	**Map Extent (UL, LR)**	**Length of the Canal**	**OC**	**RC**	**Accuracy**
Part of the river along Krishna River	(16.4371, 75.5885), (16.1892, 76.3044)	87 Kms	62	90	68.88%
Canal along Ganga basin	(20.5737, 78.2699), (20.29971, 78.7992)	78 Kms	50	72	69.44%
Canal along Bembla Dam	(15.2583, 76.3283), (15.5943, 76.8328)	72 Kms	51	67	76.11%
Canal along Krishna Sagar Dam	(12.4226, 76.629), (12.2152, 76.9129)	42 Kms	42	44	95.45%
Canal along Nagarjuna Dam (NSP Right Canal)	(16.5696, 79.3107), (16.4761, 79.5242)	23 Kms	26	26	100%

the least-cost path. Figure 5.17 represents the result for the same set of parameters but on 90 meters resolution data set in addition to the overlap of DCW data.

We define correctness as the percentage of the number of resultant path cells overlapped with respect to DCW data to the total number of cells that are present from S_C to E_C in DCW data. By using this definition, we calculate the correctness of the algorithm. As seen in Table 5.2, the columns, number of overlapped cells visited with respect to the reference data, i.e., DCW in the resultant path and the total number of cells of the reference data are necessary to calculate the correctness. We calculate the correctness for each set of input parameters provided. On an average, the algorithm correctness values for the 1KM resolution stand at 82.10%, whereas for 90 meters resolution stands at 82.08%. The average value stands out to be 82.09%. The results are a combination of both algorithms, i.e., gravitational flow and lift based flow algorithms. If we look into the range of values for 1KM resolution, they stand from 68.83 to 100, whereas, the range for 90 meters resolution on different data sets stands from 70.92 to 96.61. The difference in range values might be because not all Earth's terrestrial parameters are taken into account.

The research shows that an engineering planning problem, where trade-offs between two or more key parameters for determining the choice of the final path is challenging. By theory, the results are supposed to lie between an upper bound and lower bound, and this design presents an elliptical graph across these parameters. The set of all plausible options will lie within these bounds. In our work, this was clearly demonstrated in 2 cases, while in other cases, there were deviations. We anticipate that this can be due to the complexity of the terrain, which is currently not explicitly considered in our approach.

Table 5.2 Results on applying algorithms on multiple data sets of different resolutions

Parameters				OCV		No. of lifts		RC		Accuracy(%)	
				Resolution		Resolution		Resolution		Resolution	
S_C	E_C	PT	PI	1KM	90 M	1KM	90 M	1KM	90 M	1KM	90 M
(15.2583, 76.3283)	(15.7593, 76.9412)	0	0	53	678	0	0	77	903	68.83	75.08
(20.5737, 78.2699)	(20.2997, 78.7992)	0	0	50	656	0	0	72	872	69.44	75.22
(22.3511, 76.7838)	(22.7969, 77.7787)	0	0	94	1330	0	0	129	1845	72.86	72.08
(16.1892, 76.3044)	(16.3463, 77.6861)	0	0	137	1828	0	2	182	2467	75.27	74.09
(15.2583, 76.3283)	(15.5943, 76.8328)	0	0	51	693	0	0	67	884	76.11	78.39
(16.4371, 75.5885)	(16.1892, 76.3044)	0	0	62	844	0	0	90	1190	68.88	70.92
(12.4226, 76.6290)	(12.2152, 76.9129)	0	0	42	447	0	0	44	545	95.45	82.01
(27.1000, 81.4825)	(26.7940, 82.1018)	0	0	61	953	0	0	74	1045	82.43	91.19
(27.1000, 81.4825)	(26.8371, 81.8339)	3	2	36	479	0	0	46	573	78.26	83.59
(21.7242, 82.7340)	(21.6977, 83.2507)	0	0	61	700	0	0	62	711	98.38	98.45
(18.3517, 80.4596)	(17.5638, 81.2596)	2	2	144	1856	0	2	145	1921	99.31	96.61
(11.5762, 77.7446)	(11.4020, 77.6986)	0	0	22	275	1	1	22	315	100	87.30

OCV: Number of overlapped cells visited with respect to DCW data along the resultant path from S_C to E_C; **RC**: Number of cells of the DCW data along the path from S_C to E_C; **PT**: Number of coordinates to pass through ξ; **PI**: Number of restricted polygons ψ;

Chapter 6

Conclusions

This study proposed models for the least-cost route for canals. The models are built using the DEM data considering multiple parameters such as cost of construction, cost to lift the water along the surface of the terrain, the pass through points and the restricted regions.

The first algorithm is the gravitational flow model algorithm. This flow algorithm discusses flow where the movement of water is under the force of gravity and the cost is proportional to the length of the resultant path. TerraCost algorithm is used where it internally uses the Dijkstra's algorithm along with the canal properties defined. A path is identifiable with this algorithm only when a gravitational path exists. This algorithm runs in $O(|R| * |\psi| * w + nmlogn + n)$ complexity.

The second algorithm is the lift based flow algorithm. This flow algorithm discusses the flow where the water is lifted by using an external force such as motors, but along the surface of the terrain and the cost function is related to lift elevation function and the length of the resultant path. Similar, to the gravitational flow algorithm, this also uses the TerraCost algorithm. However, one change is how the cost function is defined. This algorithm runs in $O(|m| * 2^m + nm^2logn)$ complexity.

The results of the trade-off between the cost and the distance show that for various cases of the terrain and the spatial scale of the data, the patterns tend towards the elliptical bounds, though they can have different patterns at some other parametric combinations.

Using DCW data, we compared our results by applying the algorithms on different resolutions of multiple real-world data sets. The results proved to have an accuracy of 82.09%. It also proves that the algorithm scales linearithmically according to the resolution data and also efficient in terms of time complexity with respect to the existing algorithm.

For future work, we plan to improve the accuracy by involving several new parameters such as cost function related to land covers and further optimizing the lift based flow model. Although we have mentioned these models are for Canals, it isn't limited to just canals. It can be extended to drainage networks and many other networks as well. One more interesting problem that can be considered is applying these algorithms on different utility infrastructures such as roads.

Related Publications

- Sai Chaitanya Reddy Ponnathota and K. S. Rajan, ”Optimal Routing of Irrigation Canal Paths using a Digital Elevation Model,” page 1587–1596, 41st Asian Conference on Remote Sensing, 2020
- Sai Chaitanya Reddy Ponnathota and K. S. Rajan, ”Topology Aligned Least Cost Routing Model for Canals,” IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium, Pasadena, CA, USA, 2023, pp. 3024-3027, doi: 10.1109/IGARSS52108.2023.10282779

Bibliography

[1] W. Collischonn and J. V. Pilar. A direction dependent least-cost-path algorithm for roads and canals. *International Journal of Geographical Information Science*, 14(4):397–406, June 2000.

[2] M. Costa-Cabral and S. Burges. Digital elevation model networks (demon): A model of flow over hillslopes for computation of contributing and dispersal areas. *Water Resources Research*, 30:1681, 06 1994.

[3] D. W. D. Tarboton, K. Schreuders and M. Baker. Generalized terrain-based flow analysis of digital elevation models. In *Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation*, pages 2000–2006, 2011.

[4] D. H. Douglas. Least cost path in gis using an accumulated cost surface and slope lines. *Cartographica*, 31(3):37–51, Autumn 1994.

[5] A. Ebregt and P. de Greve. Buffer zones and their management: Policy and best practices for terrestrial ecosystems in developing countries. 2000.

[6] J. Fairfield and P. Leymarie. Drainage networks from grid digital elevation models. *Water Resources Research*, 27(5):709–717, May 1991.

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

[8] T. G. Freeman. Calculating catchment area with divergent flow based on a regular grid. *Comput. Geosci.*, 17(3):413–422, mar 1991.

[9] S. H. Modelling overland-flow hydrology for dynamic hydraulics. In *Overland flow, Hydraulics and Erosion Mechanics*, pages 89–103. edited by A. J. Parsons & A. D. Abrahams, 1992.

[10] T. Hazel, L. Toma, J. Vahrenhold, and R. Wickremesinghe. Terracost: A versatile and scalable approach to computing least-cost-path surfaces for massive grid-based terrains. *ACM Symposium on Applied Computing*, 1:52–57, April 2006.

[11] J. W. Inman. *Navigation and Nautical Astronomy for the Use of British Seamen*. W. Woodward, C. & J. Rivington, London, UK, 1835.

[12] R. S. Kim, Y. and S. Wise. Exploring multiple viewshed analysis using terrain features and optimisation techniques. *Computers & Geosciences*, 30(9):1019–1032, November 2004.

[13] P. A. J. Lea, N. L. and A. D. Abrahams. An aspect driven kinematic routing algorithm. In *Overland flow: Hydraulics and Erosion Mechanics*, pages 147–175. edited by A. J. Parsons & A. D. Abrahams, 1992.

[14] J. Lee and D. Stucky. On applying viewshed analysis for determining least-cost paths on digital elevation models. *International Journal of Geographical Information Science*, 12(8):891–905, 1998.

[15] W. Z. D. R. Montgomery. Digital elevation model grid size, landscape representation, and hydrologic simulations. *Water Resources Research*, 30(4):1019–1028, April 1994.

[16] B. of Indian Standards. Criteria for design of lined canals and guidance for selection of type of lining. 2000.

[17] J. F. O'Callaghan and D. M. Mark. The extraction of drainage networks from digital elevation data. *Computer vision, graphics, and image processing*, 28(3):323–344, December 1984.

[18] P. F. Quinn, K. J. Beven, P. Chevallier, and O. Planchon. The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models. *Hydrological Processes*, 5:59–79, 1991.

[19] S. D. Roth. Ray casting for modeling solids. *Computer Graphics and Image Processing*, 18(2):109–144, February 1982.

[20] R. L. D. T. W. Gardner, K. C. Sasowsky. Automated extraction of geomorphometric properties from digital elevation data. *Zeitschrift fur Geomorphologie, Supplementband*, 80:57–68, January 1990.

[21] D. G. Tarboton. A new method for the determination of flow directions and upslope areas in grid digital elevation models. *Water Resources Research*, 33(2):309–319, February 1998.

[22] D. Unwin. Gis, spatial analysis and spatial statistics. *Progress in Human Geography - PROG HUM GEOGR*, 20:540–551, 12 1996.

[23] P. E. Weinmann and E. M. Laurenson. Approximate flood routing methods: A review. 1979.

[24] J. P. Wilson and J. C. Gallant. Terrain analysis: Principles and applications. 2000.