Serdica Journal of Computing 18(1), 2024, pp. 27-60, 10.55630/sjc.2024.18.27-60
Published by Institute of Mathematics and Informatics, Bulgarian Academy of Sciences

Linear Programming Model for Optimizing Freshwater Distribution in Bangalore

Hemal Nayyar

Canadian International School, Bangalore, India
hemal.nayyarhn@gmail.com

Abstract

This research paper presents a linear programming model for network flow optimization, addressing the challenge of freshwater management in Bangalore. The model highlights the practicality of linear programming in real-world scenarios. Specifically, the efficient allocation and distribution of freshwater resources from various sources, including reservoirs, rivers, and groundwater, to meet the growing demands of domestic, industrial, and agricultural sectors, while adhering to sustainable practices.

Keywords: Linear Programming, Water Resource Management, Freshwater Distribution, Optimization, Network Flow Models

ACM Computing Classification System 2012: Theory of computation → Design and analysis of algorithms → Mathematical optimization, Applied computing → Physical sciences and engineering → Earth and atmospheric sciences → Environmental sciences

Mathematics Subject Classification 2020: 90C05, 90B50, 76S05

1 Introduction

Bangalore, a city currently ranked No. 1 in water scarcity among urban areas and India's, faces a water crisis due to rapid urbanisation, population growth, and erratic monsoon patterns over the last couple years. Efficient water management in Bangalore should therefore be enhanced to help reduce the current situation and provide comfort to those who are suffering from the shortage.

This paper proposes a linear programming model, to model the water distribution network in Bangalore, using publicly available data to project and plan

Received: October 25, 2024, *Accepted:* November 8, *Published:* November 27, 2024

Citation: Hemal Nayyar, Linear Programming Model for Optimizing Freshwater Distribution in Bangalore, Serdica Journal of Computing 18(1), 2024, pp. 27-60, https://doi.org/10.55630/sjc.2024.18.27-60

various scenarios until complete water depletion: No rainfall, incorporating rainfall, bore water + reservoir water reliance.

By utilising real-world data, open to the general public, the model simulates month-by-month variations in supply and distribution constraints across key areas of Bangalore: Yelahanka, White Field, Electronic City, Peenya, Banashankari, JP Nagar, KR Puram, Marathahalli, Sarjapur, Nandi Hills, Chandapura, and other regions. The model aims to showcase the feasibility of Linear Programming Models for large urban areas.

The findings of this research, hope to illustrate the feasibility of large scale linear-programming models in real world scenarios. Potentially aiding in mitigating future and current water crises, such as Bangalore, if further explored and developed to incorporate more real world data and external variables.

1.1 Literature Review

1.1.1 Introduction to Network Flow

In understanding linear programming for network flow, [1] offers a comprehensive guide on the application of linear programming to network flow problems. The paper highlights how linear programming can be used to optimise network flows, alongside examples for managing water distribution networks in urban environments.

For more complex integration of linear programming and advanced concepts, Henry Adams' [2] collection of video series offers a comprehensive understanding of the topic, including understanding Simplex Algorithms and how they function.

1.1.2 Network Flow Models and Linear Programming Approaches

Network flow models are tools used for optimising distribution of resources in systems. These models have been used in optimization research and computer science to solve various real-world problems, including water distribution networks.

Iancheva and Kelevedzhiev (2001) [3] provide a foundational approach to managing water resource systems using linear programming. Their paper is based on the upper basin of the Iskar River and serves as an introduction to mapping network flows over larger geographic areas. This approach demonstrates the potential for similar methodologies to be applied in more complex urban environments, such as Bangalore, as explored in this paper.

1.1.3 Bangalore's Water Supply Challenges

Bangalore's water crisis is due to the disparity between water supply and demand, affected by the changes in seasonal rainfall patterns (over the last few years) and the city's growing population. The Bangalore Water Supply and Sewerage Board (BWSSB) [4] and the Central Ground Water Board (CGWB) [5] have published reports detailing the city's water supply infrastructure and groundwater status. These reports have provided data for modelling and understanding Bangalore's water distribution system, which has served as the backbone to develop the Bangalore Model used in this study.

1.1.4 Reservoirs

The Central Ground Water Board's *Ground Water Year Book 2021-2022: Karnataka and Goa [5]* and BWSSB's [4] annual reports offer data on the current state of Bangalore's reservoirs. These reports examine the importance of maintaining optimal reservoir levels to prevent shortages during dry periods and overflows during the monsoon season.

1.1.5 Historical Rainfall Patterns

The Indian Meteorological Department (IMD) [6] and the Central Water Commission (CWC) [7] provide data on historical rainfall patterns for predicting water availability. If integrated correctly into a linear programming model, it would allow for more accurate water resource management.

1.1.6 Gaps in Literature

There is extensive literature on the understanding of linear programming. However, there are very few resources that apply linear programming in a real-world scenario, specifically network flow. Additionally, there are gaps in accessing real-time data for such studies, hence maximising assumptions. This is furthered by the lack of resources available for accurately mapping out a network flow model for a larger urban area such as Bangalore.

2 Objectives

1. **Model the Water Supply Network:** Represent the city's water distribution system as a flow network with nodes – representing water sources /

reservoirs, treatment plants, and consumers; and edges – representing pipes with specific capacities.
2. **Ensure Necessary Water Distribution:** Ensure that all necessary constraints for agriculture, industry, and residential water usage are met.
3. **Scenario Analysis:** Model possible scenarios for the next 5 years for the supply of freshwater in Bangalore, considering changes in receiver demands and rainfall values.
4. **Maintain Longevity of Water Supply:** Ensure freshwater is distributed to all necessary locations while minimising usage, providing only the bare minimum to meet requirements.

3 Preliminaries

3.1 Terminology and Definitions

Graph: A graph consists of a set of vertices and a set of edges. Each edge connects two vertices and can have an associated weight/min-max-capacity.

Vertex (Node): A vertex of the graph represents an entity: water source, treatment plant, or consumer endpoint in a water supply network.

Edge (Link): An edge is a connection between two vertices in the graph, representing a pipeline in the water supply network. It can have an associated capacity indicating the maximum and minimum flow it can handle.

Capacity: The capacity of an edge is the maximum amount of flow that can pass through the edge. In a water supply network, it represents the maximum volume of water that can flow through a pipeline.

Flow Network: A flow network is a directed graph where each edge has a capacity and each edge receives a flow. The amount of flow on an edge cannot exceed the capacity of the edge.

Source: The source in a flow network is the starting node from which flow originates. In a water supply network, it represents the water sources or treatment plants.

Sink: The sink in a flow network is the terminal node where flow is absorbed. In a water supply network, it represents the consumer endpoints.

Linear Program (LP): A mathematical model representing a problem that seeks to maximize or minimize a linear objective function, subject to a set of linear constraints, excluding multiplication and division.

Decision Variables: Variables that represent the decisions to be made in the problem, which will optimize the objective function.

Objective Function: A linear function that needs to be maximized or minimized. It is expressed in terms of decision variables.

Constraints: A set of linear inequalities or equalities that represent the restrictions or limitations on the decision variables.

Feasible Region: The set of all possible points (combinations of the values of the decision variables) that satisfy the constraints of the linear program. It is often visualized as a convex polytope.

Vertex (Corner Point): A point in the feasible region where two or more constraints intersect. In the context of the Simplex algorithm, the optimal solution is often found at one of these vertices.

Simplex Algorithm: An iterative method used to find the optimal solution of a linear program by moving from one vertex of the feasible region to an adjacent vertex with a better objective value, until the optimal solution is reached.

Optimal Solution: A feasible solution that maximizes or minimizes the objective function value. In the Simplex algorithm, it is the final solution where no further improvements can be made.

3.2 Assumptions

Static Network: The capacities of the pipes and the water demands at each node do not change over time. This simplifies the model but does not capture real-world fluctuations, such as the age of pipes and pipe deterioration.

Single Commodity Flow: The model assumes a single type of flow rate through the network. In reality, water quality and pressure might vary, but these factors are not considered in the current model. However, such features may be added.

Steady-State Conditions: The analysis assumes steady-state conditions, meaning that the flow rate is constant over time. Transient states, such as those during peak hours or maintenance periods, are not considered.

Perfect Infrastructure: The network infrastructure is assumed to be perfect, with no leaks, breaks, or maintenance issues. This may slightly limit accuracy, although it should not be considerable.

Linear Relationships: All relationships between the decision variables and the objective function, as well as between the decision variables and the constraints, are linear. This means that each term in the objective function and the constraint is either a constant or a constant multiplied by a decision variable.

Feasibility: The feasible region formed by the simultaneous fulfillment of all constraints is a convex polytope, meaning that an optimal solution exists at one of the vertices of the polytope. This ensures there are no contradictory constraints that would make the problem infeasible.

Non-Negativity: All decision variables are non-negative. This is an assumption in linear programming to reflect realistic conditions, such as non-negative quantities of goods or resources.

No Backflow: The model assumes that water is unable to flow in reverse through pipes, implying that the system operates on either a downhill slope or flat plane.

Boundedness: The feasible region is bounded, meaning there are limits on the values that decision variables can take. This prevents the objective function from being unbounded, ensuring a finite optimal solution and improving feasibility.

Single Objective Function: The problem has a single objective function to be optimized (either maximized or minimized). This simplifies the analysis and ensures clarity in the optimization direction.

4 Algorithms

The network flow model is based on the concept of Linear Programming, a mathematical programming to achieve the most optimal outcome in a mathematical model whose requirements are represented by linear relationships and constraints. The concept of linear programming is further explored in the collection of videos by Henry Adams [2].

4.1 Implementation of the Simplex Algorithm

LP_Solve: LP_Solve [8] is an open-source solver for linear programming problems that uses the Simplex algorithm and its variations.

- **Usage in Study:**
  - *Initialization:* LP_Solve was initially used to develop and test the water distribution model.
- **Advantages:**
  - Its ability to handle large datasets and include complex constraints made it ideal for the testing stages and practical problems.
- **Disadvantages:**
  - Limits users to add variables as constraints, resulting in it being time-consuming for larger iterations.

- Since LP_Solve is not an actual programming language, iterations must be made manually when running through the terminal.

PuLP: PuLP is a Python library for linear programming, similar to LP_Solve.

- **Usage in Study:**
  - *Modelling and Implementation:* After the initial model validation with LP_-Solve, PuLP was used for further development, simulations, and experimentation before shifting back to LP_Solve due to issues in logic.
- **Advantages:**
  - The integration with Python allowed for easier manipulation of data and model parameters, including the use of Pandas and MathPlotLib.

5 Secondary Data

5.1 Population Data (External/Secondary)

- **Current Population (2023):**
  - Total population: 13,300,000
  - Growth rate: 5.13% per year
- **Breakdown by area (with individual growth rates):**
  - Yelahanka: 320,000 (4.17%)
  - Whitefield: 350,000 (7.376%)
  - Electronic City: 250,000 (7.51%)
  - Peenya: 270,000 (6.326%)
  - Banashankari: 420,000 (4.076%)
  - JP Nagar: 260,000 (6.87%)
  - KR Puram: 340,000 (5.956%)
  - Marathahalli: 240,000 (8.31%)
  - Sarjapur: 230,000 (9.35%)
  - Nandi Hills: 80,000 (8.29%)
  - Chandapura: 120,000 (8.28%)
  - *Other areas:* (Calculated based on remaining population and overall growth rate)

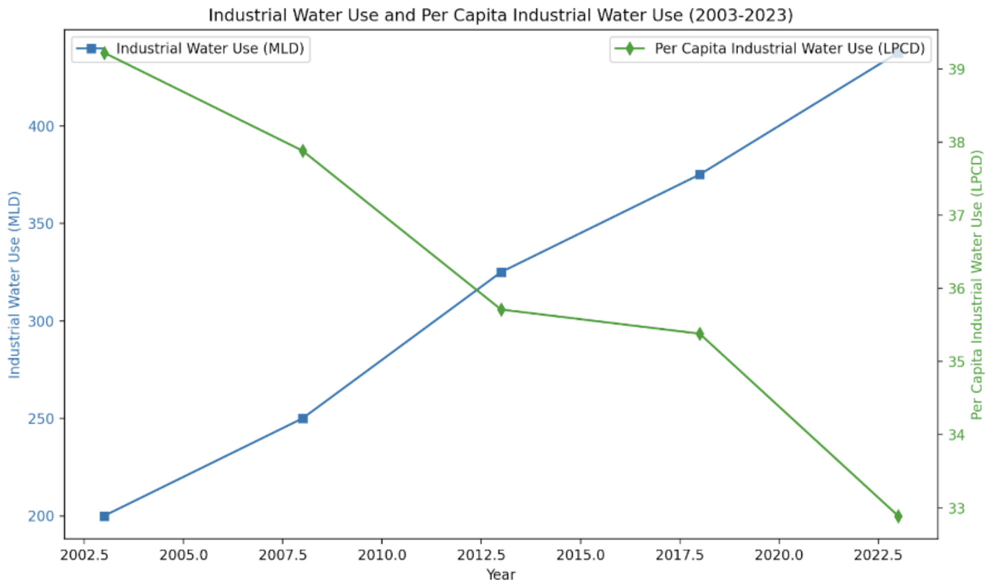
Area	**2003**	**2008**	**2013**	**2018**	**2023**
Yelahanka	150	180	220	270	320
Whitefield	100	140	200	270	350
Electronic City	70	100	140	190	250
Peenya	90	120	160	210	270
Banashankari	200	240	290	350	420
J.P. Nagar	80	110	150	200	260
K.R. Puram	120	160	210	270	340
Marathahalli	60	90	130	180	240
Sarjapur	50	80	120	170	230
Chandapura	30	43	60	86	120
Nandi Hills	20	29	40	57	80
Total	970	1292	1720	2253	2880
Bangalore	5100	6600	9100	10600	13300
Percentage of Total Population	19.02%	19.58%	18.90%	21.25%	21.65%

Table 1: Population Data from 2003 to 2023.

Year	**Population (million)**	**Industrial Water Use (MLD)**	**Per Capita Industrial Water Use (LPCD)**
2003	5.1	200	39.22
2008	6.6	250	37.88
2013	9.1	325	35.71
2018	10.6	375	35.38
2023	13.3	437.5	32.89

Table 2: Industrial Water Use and Per Capita Industrial Water Use (2003-2023).

5.2 Per Capita Industrial Water Use



5.3 Rainfall Analysis

Month	Average Rainfall (mm)
Jan	3.1
Feb	1.78
Mar	6.3
Apr	21.975
May	14.325
Jun	18.9
Jul	16.925
Aug	49.975
Sep	6.7
Oct	46.5
Nov	6.475
Dec	14.675

Table 3: Average Monthly Rainfall (mm). Note: There is a 2% annual decrease in rainfall, which was taken into account in the model.

Reservoir Name	Area (km^2)	Area (m^2)
Thippagondanahalli Reservoir	1453	1,453,000,000
Hesaraghatta Lake	73.83	73,830,000
Meenkara Dam	90.65	90,650,000
Garajanar Dam	419.58	419,580,000
Vani Vilasa Dam	5.374	5,374,000
Mangalam Dam	48.85	48,850,000
Markonahalli Dam	4103	4,103,000,000
Kabbini Dam	2141.9	2,141,900,000
Basava Sagar Dam	47850	47,850,000,000
Tungabhadra Dam	28180	28,180,000,000
Harangi Dam	419.58	419,580,000

Table 4: Reservoir Areas. This data was used in conjunction with the total surface area of each dam/lake, assuming that each dam in its entirety is outdoors and that each km^2 receives an equal amount of rainfall.

5.4 Water Sources(External + Hypothetical Reasoning)

5.4.1 River Sources

River	Average Inflow (MLD)
Cauvery River	1450
Arkavathi River	123

Table 5: Average Inflow from River Sources.

5.4.2 Reservoirs and Dams

See Table 6.

5.5 Treatment Plants (External + Hypothetical Reasoning)

- **T.K. Halli Treatment Plant:** 1450 MLD (However, this number is unrealistic and was changed later in the study).
- **Vrishabhavathi Valley Treatment Plant:** 180 MLD (However, this number is unrealistic and was changed later in the study).

Reservoir / Dam	Maximum Capacity (MCM)	Maximum Outflow
Thippagondanahalli Reservoir	74	7.4
Hesaraghatta Lake	125	12.5
Markonahalli Dam	68	6.8
Gajanur Dam	56	5.6
Harangi Dam	130	13.0
Mangalam Dam	70	7.0
Kabbini Dam	150	15.0
Tungabhadra Dam	200	20.0
Basava Sagar Dam	190	19.0
Meenkara Dam	80	8.0
Vani Vilasa Sagara	90	9.0

Table 6: Reservoir and Dam Capacities with Assumed Maximum Outflow (10% of Maximum Capacity) [9].

5.6 Water Data (External + Hypothetical Reasoning)

5.6.1 Residential Water Demand

- Per capita consumption: 135 liters per day (India's recommended).
- Demand for specific areas calculated based on population and growth rates.

5.6.2 Agricultural Water Demand

- Current agricultural land: 59,049 acres.
- Daily water requirement: 20,000 liters per acre (Hypothetical: based on averages of water usage).
- Annual decrease in agricultural land: 2.41%.

5.6.3 Industrial Water Demand

- Current demand: 437.5 MLD
- Annual increase in demand: (to be specified).

6 Formula

6.1 Total Population Water Usage

$$= \frac{(\text{Initial Population} \times (\text{Rate of Increase})^{(\text{Days}/365)}) \times \text{Per Capita Water Usage}}{1000}$$

Assumptions

- Initial Population (P) = Sum of populations in all areas
- Per Capita Water Usage (W) = Average water usage per person per day (liters)
- Rate of Increase (R) = Constant for all years

Values

- Rate of Increase (R) = 5.13%
- Per Capita Water Usage (W) = 0.000135 MLD
- Initial Population (P) = 13,300,000

6.2 Agriculture Water Usage

$$= \frac{\left(\text{Initial Number of Acres} \times \text{Rate of Decrease}^{\frac{365}{\text{Days}}}\right) \times \left(\frac{\text{Per Acre per day usage}}{1{,}000{,}000}\right)}{1{,}000}$$

Assumptions/Values:

- Rate of Decrease of Number of Acres due to Urbanization (R) = -2.41% per year
- DAYS (D) = The period in days over which the calculation is made (i.e., one year = 365 days)
- Per Acre per day usage (P) = 20,000 liters
- Initial Number of Acres (N) = 59,049 acres

6.3 Industrial Water Usage

$$= \frac{\text{Initial Population} \times \text{Rate of Increase}^{\frac{Days}{365}} \times \text{Per Capita Industrial Water Usage}}{1000}$$

Assumptions/Values:

- Initial Population (P) = Sum of populations in all areas
- Rate of Increase (R) = Constant for all years (5.13%)
- DAYS: The period in days over which the calculation is made (e.g., one year = 365 days)
- Per Capita Industrial Water Usage (W) = Average water usage per person per day (liters), W = 0.000036216 MLD
- Initial Population (P) = 13,300,000

6.4 Bore-Water Usage

Bore Water Usage (MCM) = Total Water Usage (MCM) × 0.3
Bore Water Usage is assumed to be 30% of any system's water usage.

6.5 Calculating Yearly Rainfall Change

$$\text{Rainfall (N)} = \text{Rainfall (L)} \times \left(1 - \frac{y}{100}\right)^{(N-2023)}$$

Assumptions:

- Rainfall (L) = Rainfall for the base year (2023)
- Rate of Decrease (Y) = Percentage Decrease Per Year
- Rainfall (N) = Predicted rainfall for N years after 2023

6.6 Rainfall Volume Formula

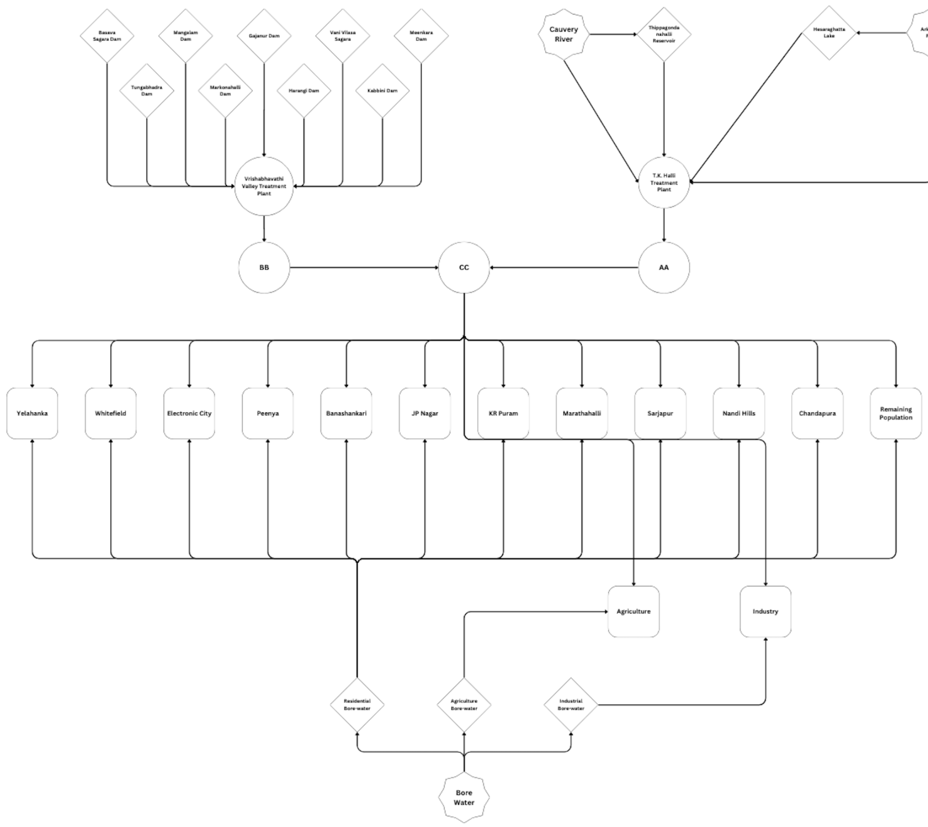
$$\text{Volume (MCM)} = \frac{\text{Rainfall} \times \left(1 - \frac{x}{100}\right)^{(n-2023)} \times \left(\frac{A \times 1{,}000{,}000}{1{,}000}\right)}{1,000,000}$$

Assumptions:

- A = Area of Reservoir/Water Body
- Rainfall = Initial Rainfall (2023)
- n = Years after 2023

7 Model

The model presented in the study was run for a total time span of 5 years. The reason for running the model for only 5 years was to avoid a greater amount of uncertainty that may be caused by irregular changes due to climate change or the intervention of laws that prohibited childbirth. Additionally, the rapid movement of society into an age of not having children could have affected the study. These various factors are not covered in this study. The main objective of this study is to illustrate that linear programming models can be used for large data sets, and thus the model was only run for 5 consecutive years.



The model shown here is based on purely assumptions lacking any physical evidence,provided to the public, to justify the accuracy of the model. The model was constructed by analysing various resources to ultimately design a model that should be nearly-accurate to the Bangalore fresh-water network. The model has 2 main sources of freshwater:Cauvery River and the Arkavathi River. However, there is an additional source of freshwater that, unlike the rivers, is replenished through rainfall and surface run-off(the inflow of surface-run-off is not implemented in this study due to the complexity and missing variables), bore water.

The model in total has 14 distinct sinks, 11 reservoirs excluding the additional 3 bore water sources that are split from one main bore water source(unrealistic due to there only being 3 main sources, however this was made due to complexity of the model and lack of conclusive data, but should work similar to a real-life example) and 2 treatment plants.

The model only includes big-landmarks/sinks of water due to the millions of individual sources. However to increase the accuracy of the model an additional sink was added, labelled 'Remaining Population', including the rest of Bangalore total population but not including the water consumption of the 11 main areas: Yelahanka, White Field, Electronic City, Peenya, Banashankari, JP Nagar, KR Puram, Marathahalli, Sarjapur, Nandi Hills and Chandapura.

It is also important to note that originally the model included 3 additional lakes (reservoirs): Bellandur Lake, Doddanekundi Lake and Madiwala Lake. However, due to the lack of data regarding maximum capacity and maximum output, the 3 lakes were removed from the study, limiting the accuracy.

Additionally, it is important to note that the study only covers inflow to the reservoirs by rainfall and not through external lakes and rivers, due to the lack of data, further limiting the accuracy.

The Treatment Plants indicated in the model by Circles do not affect the results presented in this study, due to the lack of data to conclude the maximum MLD however it is assumed for the T.K. Halli Treatment Plant to be 1.95 MCM per day and 3.67 MCM per day.

7.1 Mathematical Model Description

The linear programming model is formulated as a system of equalities and inequalities that ensure the water flow through the network obeys constraints due to capacities, demands, and source availability. The following is its formulation:

Decision Variables

- x_{ij}: Volume of water flowing from node i to node j (in MCM).
- s_i: Supply available at node i (in MCM).
- d_j: Demand at node j (in MCM).
- c_{ij}: Maximum capacity of the pipeline between i and j (in MCM).

Objective Function

The goal is to minimize the total water flow while achieving the evolving demands:

$$\text{Maximize} \sum_{(i,j)\in E} x_{ij} - \text{Penalty Terms (e.g., unmet demand, overflow).}$$

Constraints

1. **Flow Conservation:** At every node i, the sum of inflows equals the sum of outflows plus the net supply:

$$\sum_{j\in N} x_{ji} - \sum_{j\in N} x_{ij} = s_i - d_i, \quad \forall i \in N.$$

2. **Capacity Constraints:** The flow between any two nodes cannot exceed the pipeline capacity:

$$0 \leq x_{ij} \leq c_{ij}, \quad \forall (i,j) \in E.$$

3. **Reservoir Constraints:** Each reservoir r must maintain a non-negative water level, with the net inflow limited by its capacity:

$$0 \leq \text{Water Level}_r \leq \text{Max Capacity}_r, \quad \forall r \in R.$$

4. **Non-Negativity:** The flow variables must be non-negative:

$$x_{ij} \geq 0, \quad \forall (i,j) \in E.$$

5. **External Factors:** Rainfall and other external inflows e_r into reservoirs are added as external variables:

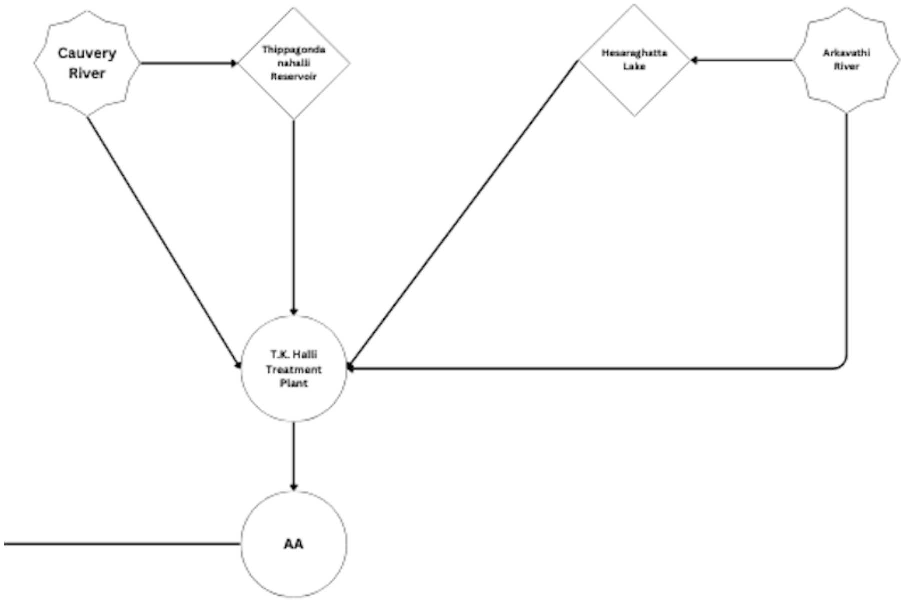
$$\text{Water Level}_r = \text{Initial Level}_r + \sum_{i\in N} x_{ir} - \sum_{j\in N} x_{rj} + e_r.$$

6. **Demand Satisfaction:** The total water delivered to a demand node must meet or exceed its demand:

$$\sum_{i\in N} x_{ij} \geq d_j, \quad \forall j \in D.$$

7.2 Close ups of Model

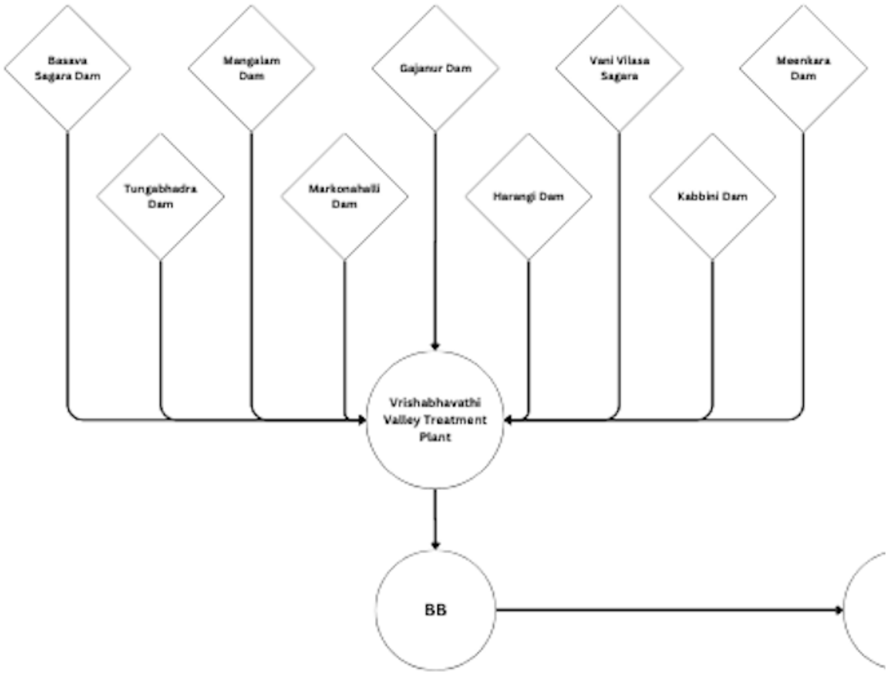
7.2.1 Part A+B



Water Flow

- **Cauvery River (SA) → Thippagondanahalli Reservoir (RA):**
  - Max Flow (MCM) = 1.45
- **Thippagondanahalli Reservoir → T.K. Halli Treatment Plant:**
  - Max Flow (MCM) = 0.5
- **Cauvery River → T.K. Halli Treatment Plant:**
  - Max Flow (MCM) = 1.45
- **Arkavathi River → Hesaraghatta Lake:**
  - Max Flow (MCM) = 0.125
- **Hesaraghatta Lake → T.K. Halli Treatment Plant:**
  - Max Flow (MCM) = (to be specified)
- **Arkavathi River → T.K. Halli Treatment Plant:**
  - Max Flow (MCM) = 0.125

7.2.2 Part C

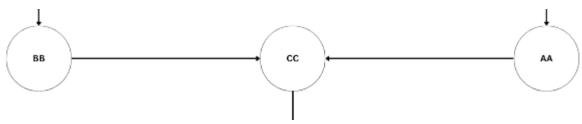


Water Flow

- **Basava Sagar Dam → Vrishabhavathi Valley Treatment Plant:**
  - Max Flow (MCM) = 0.19
- **Tungabhadra Dam → Vrishabhavathi Valley Treatment Plant:**
  - Max Flow (MCM) = 0.2
- **Mangalam Dam → Vrishabhavathi Valley Treatment Plant:**
  - Max Flow (MCM) = 0.07
- **Markonahalli Dam → Vrishabhavathi Valley Treatment Plant:**
  - Max Flow (MCM) = 0.68
- **Gajanur Dam → Vrishabhavathi Valley Treatment Plant:**
  - Max Flow (MCM) = 0.56
- **Harangi Dam → Vrishabhavathi Valley Treatment Plant:**
  - Max Flow (MCM) = 0.13

- **Vani Vilasa Sagara → Vrishabhavathi Valley Treatment Plant:**
  - Max Flow (MCM) = 0.9
- **Kabbini Dam → Vrishabhavathi Valley Treatment Plant:**
  - Max Flow (MCM) = 0.15
- **Meenkara Dam → Vrishabhavathi Valley Treatment Plant:**
  - Max Flow (MCM) = 0.8

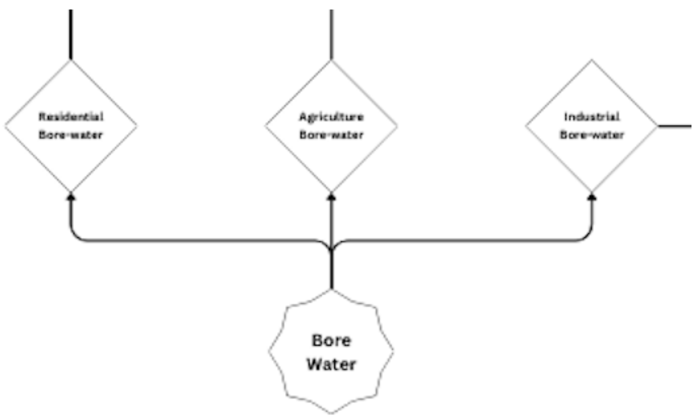
7.2.3 Part D



Water Flow

- **Vrishabhavathi Valley Treatment Plant → BB:**
  - Max Flow (MCM) = 0.5
- **BB → CC:**
  - (Max Flow not specified)
- **T.K. Halli Treatment Plant → AA:**
  - Max Flow (MCM) = 1.95

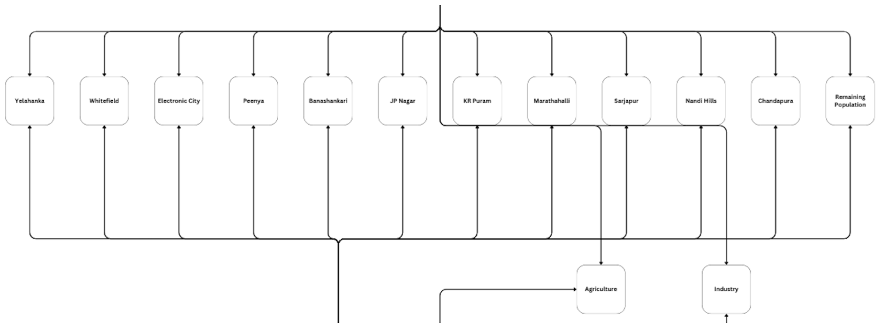
7.2.4 Part E



Water Flow

- **Bore Water → Residential Bore Water:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **Bore Water → Agricultural Bore Water:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **Bore Water → Industrial Bore Water:**
  - Max Flow (MCM) = Dependent on Continuous Variables

7.2.5 Part F



Water Flow

- **CC → Yelahanka:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **CC → Whitefield:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **CC → Electronic City:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **CC → Peenya:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **CC → Banashankari:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **CC → JP Nagar:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **CC → KR Puram:**
  - Max Flow (MCM) = Dependent on Continuous Variables

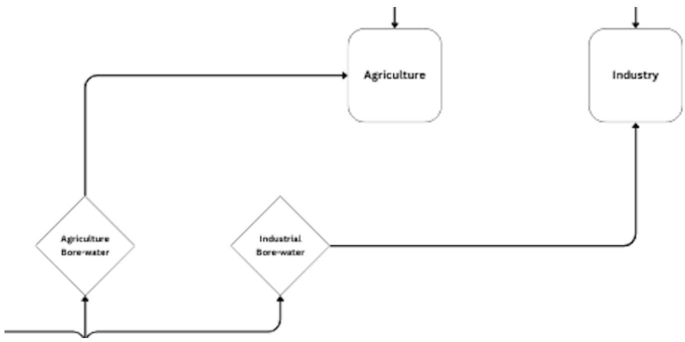
- **CC → Marathahalli:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **CC → Sarjapur:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **CC → Nandi Hills:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **CC → Chandapura:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **CC → Other Areas:**
  - Max Flow (MCM) = Dependent on Continuous Variables

- **Agricultural Bore Water → Yelahanka:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **Agricultural Bore Water → Whitefield:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **Agricultural Bore Water → Electronic City:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **Agricultural Bore Water → Peenya:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **Agricultural Bore Water → Banashankari:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **Agricultural Bore Water → JP Nagar:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **Agricultural Bore Water → KR Puram:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **Agricultural Bore Water → Marathahalli:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **Agricultural Bore Water → Sarjapur:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **Agricultural Bore Water → Nandi Hills:**

- Max Flow (MCM) = Dependent on Continuous Variables

- **Agricultural Bore Water → Chandapura:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **Agricultural Bore Water → Other Areas:**
  - Max Flow (MCM) = Dependent on Continuous Variables

7.2.6 Part G



Water Flow

- **Agricultural Bore Water → Agriculture:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **Industrial Bore Water → Industry:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **CC → Agriculture:**
  - Max Flow (MCM) = Dependent on Continuous Variables
- **CC → Industry:**
  - Max Flow (MCM) = Dependent on Continuous Variables

7.3 Linear Programming Model Structure and Notation

7.3.1 Key Notation

- If the first letter is capital ‘R’, the constraint is a reservoir.
- If the second letter following is ‘P’, the constraint is a treatment plant.
- If the first letter ‘U’ is followed by ‘R’, it represents water flowing into a reservoir.

- If the first letter 'V' is followed by 'R', it represents water flowing out of a reservoir.
- 'BR' represents bore water for residential use.
- 'BI' represents bore water for industrial use.
- 'BA' represents bore water for agricultural use.
- 'AA' or 'BB' represents the collection point for more than one reservoir, dam, or lake.
- 'CC' represents the collection point of both 'AA' and 'BB'.
- The number after the letters represents the month: 1 = 1st month, etc.
- The '_' between two constraints/nodes represents that this constraint is for the pipe/weight, not a node.

7.3.2 Additional Notation (Requirement for External Variables (i.e. Rainfall))

- If the first letter is capital 'E', the constraint is an external variable.

Note: All the constraints below are not specific to this model and serve as an explanation of how the LP model is structured. The explanations were added to guide users who may want to recreate such a model.

Additionally, the names of the variables presented in the explanation do not need to be identical or in capital letters and will work regardless. Any constraint or variable declared must not include any arithmetic operators.

7.3.3 Reservoirs

Initial Values of Reservoir The initial reservoir values are modeled as follows:

$$RA = 30;$$

This needs to be written only once at the beginning of the LP_Solve program.

In all the scenarios presented in this study, the reservoirs are at 100% capacity initially. However, the values of the reservoirs can be changed accordingly by modifying the initial values.

How to Declare a Reservoir in LP_Solve In order to declare a constraint to be a reservoir, the format in this model is as follows (this format was created specifically for this study and does not take inspiration from external sources due to limited resources):

$$RA1 = RA + URA1 - VRA1 + ERA1$$

Where:

- $RA1$ = Current volume of water in the reservoir.
- RA = Last month/week/day/year's water in the reservoir.
- URA = Water entering the reservoir (internal, e.g., pipes).
- VRA = Water leaving the reservoir.
- ERA = Water entering the reservoir (external, e.g., rainfall) (optional).

7.3.4 Constraints

How to Declare a Constraint for a Pipe Basic constraints for pipes are modeled as follows:

ParentNode_ChildNode $<$ 5;

The number '5' can be replaced with any variable value as well.

The relational operator between the variables is limited to:

- $<$, $>$, $=$, $\leq$, $\geq$

How to Declare a Constraint for a Node Basic constraints for nodes are modeled as follows:

NameOfTheNode $>$ 5;

How to Declare a Variable Variables are modeled as follows:

NameOfTheVariable $>$ 5;

7.3.5 How to Simulate the Passage of Time (Months/Days/Years)

In order to simulate the passage of time, it is necessary to copy and paste all the constraints of the model (excluding the initial reservoir values) beneath the existing model.

$$RA1 = RA + URA1 - VRA1 + ERA1$$

$$RA1 = RA + URA1 - VRA1 + ERA1$$

Accordingly, you must increment the end number values of each variable, constraint, and reservoir:

$$RA1 = RA + URA1 - VRA1 + ERA1$$
$$RA2 = RA1 + URA2 - VRA2 + ERA2$$

Additionally, you must increment the variables inside the Max or Min objective function as well.

8 Methodology

A linear programming model of the Bangalore Fresh-Water Network Flow was designed, incorporating:

- Reservoirs/Dams
- Sources
- Treatment Plants

Note: Due to limited public availability of certain data, the model was constructed using the available data, supplemented by reasonable assumptions.

8.1 Objective

Construct the variable your model will be solving: Maximum Flow, Minimum Flow.

8.2 Factors Considered

This data was paired in conjunction with factors such as:

- Reservoir Capacity (min-capacity not included in this study due to the lack of available data)
- Reservoir Out-flow Min and Max Capacities (not included in this study due to the lack of available data)
- Rainfall Data
- Maximum Flow
- Minimum Flow (not included in this study due to the lack of available data)
- Treatment Plants Capacity (not included in this study due to the lack of available data)

8.3 Constraints

Constraints in linear programming were designed for each section and combined to complete the structure of the model.

8.4 Reservoirs

Reservoirs are added based on the logic described in the previous section.

8.5 Data Input

Data is added to the model based on a 30-day cycle: each time the model runs, it simulates 30 days.

8.6 Time Simulation

The initial model is repeated for X amount of time (30-day cycle * X), with each step/repetition including new constraints (depending on the requirements) and new data values for each constraint (depending on the requirements).

8.7 Execution

The model is run in the terminal (explored further on the website `LP_solve`).

9 Results

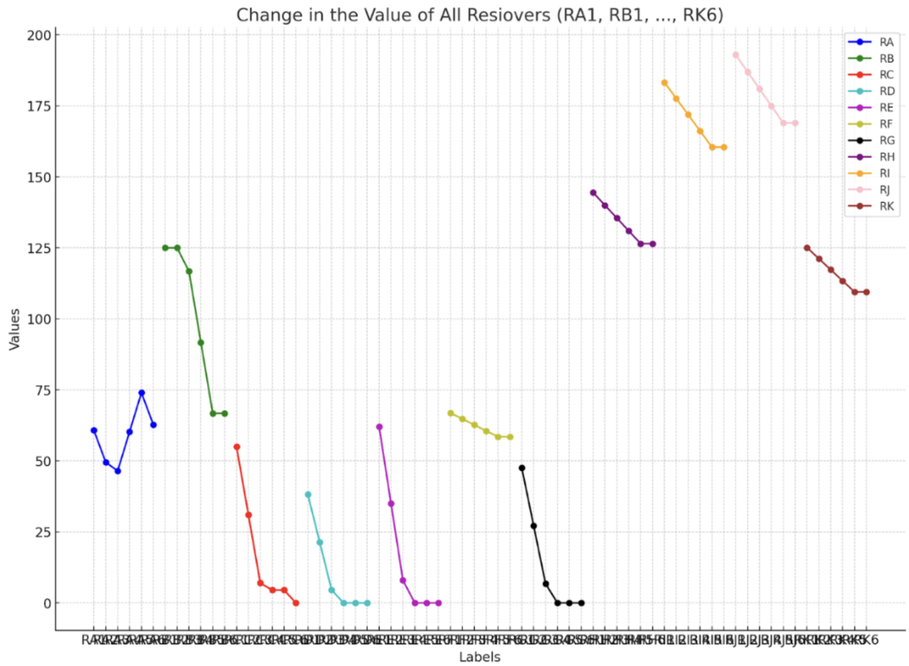
9.1 Scenario 1 [10]: Evaluating Bore Water Sustainability During Water-Scarcity

9.1.1 Bore Water Contribution

According to available data, 30% of Bangalore's total water supply is sourced from bore water. A test was conducted to evaluate the longevity of this water supply distribution.

Initial Conditions: Reservoirs and bore water sources started at 100% capacity. The system did not receive any additional water through rainfall or other sources.

Distribution: Each sector – Residential, Industrial, and Agricultural – received 30% of their water from bore water sources.



9.1.2 Findings

Duration: The system sustained water supply for 6 months before the bore water sources were depleted.

The graph below illustrates the depletion of bore water over the 6-month period. Although the graph appears linear, it represents a gradual change in water usage, indicating non-linear consumption patterns due to increasing usage for Residential and Industrial purposes, and decreasing usage for Agricultural purposes.

This graph indicates the change in supply of water from the reservoirs for the first 6 months, ultimately until the bore-water was depleted:

- RA : Thippagondanahalli Reservoir
- RB : Hesaraghatta Lake
- RC : Meenkara Dam
- RD : Gajanur Dam
- RE : Vani Vilasa Sagara
- RF : Mangalam Dam
- RG : Markonahalli Dam

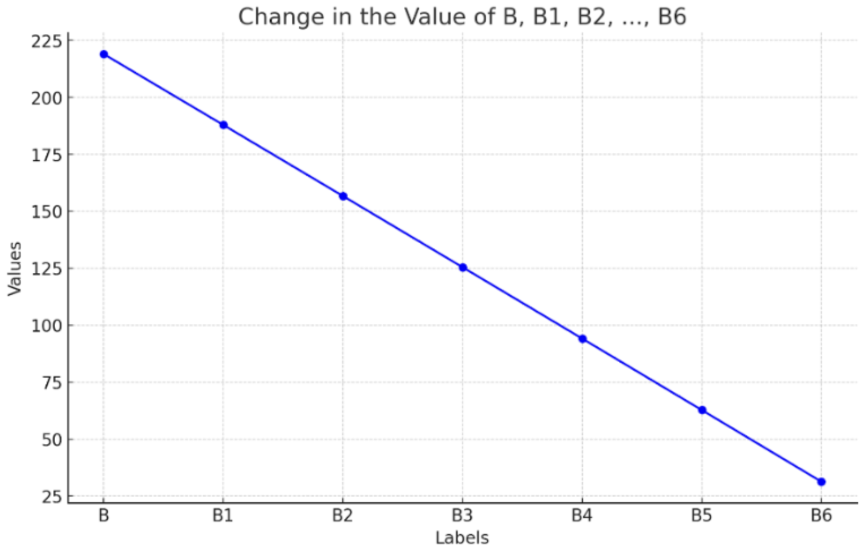


Figure 1: Water Supply from Reservoirs Over the First 6 Months.

- RH : Kabbini Dam
- RI : Basava Sagar Dam
- RJ : Tungabhadra Dam
- RK : Harangi Dam

9.2 Model Analysis

Upon running this model, it is important to note that the model is not fully accurate, as there is no minimum and maximum outflow rate for each of the reservoirs. *Note:* In this study, it is assumed that the maximum outflow is 10% of the maximum capacity of the reservoir. This results in only one reservoir being depleted before moving to the next reservoir.

A possible solution to prevent this would be to enforce a minimum amount of water that must remain in the reservoir at any given time. Additionally, with further data, the model can be optimized by adding constraints for both minimum and maximum outflow rates of water.

Note: This would only be accurate if real-world data, which is not currently available to the public, is used. Such data was not incorporated into this model.

9.3 Test Scenario 2 [10]: Reservoir-Only Utilisation After Bore Water Depletion - Worst Case Scenario

Scenario: Once the bore water was depleted, the system relied solely on reservoirs until exhaustion.

Initial Conditions: Reservoirs started at 100% capacity, with no bore water available after the initial 6 months, based on Test Scenario 1.

9.3.1 Findings

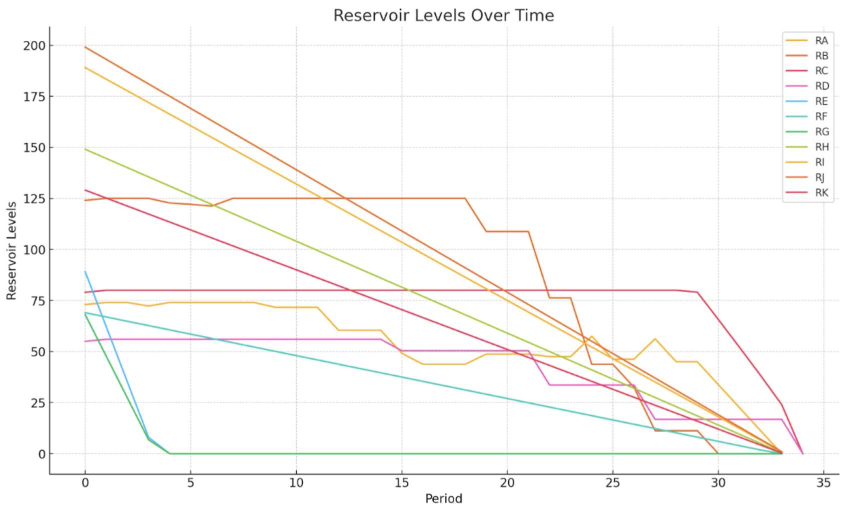


Figure 2: Reservoir Water Depletion Over the 34-Month Period.

Duration: The system sustained water supply for 34 months before the reservoir water was depleted.

Graph Analysis: The graph above illustrates the depletion of reservoir water over the 34-month period. Although the graph appears linear, this is due to the lack of available information mentioned in the previous scenario (min/max outflow rates, min/max inflow rates, daily capacity of treatment plants, minimum amount of water needed in a reservoir).

9.4 Test Scenario 3 [10]: Incorporation of External Factors (Rainfall)

Initial Conditions: Reservoirs and bore water sources started at 100% capacity. The system – including bores, reservoirs, and dams – received additional water through rainfall but no other external factors.

Distribution: Each sector – Residential, Industrial, and Agricultural – received 30% of their water from bore water sources. The model was run for a total of 6 months to mimic the initial test. However, in this scenario, the bore water did not receive any external water due to the limited data on how to calculate the water received by each bore.

Assumptions: In this scenario, it is assumed that each reservoir, dam, or lake receives the same amount of rainfall (in mm) at a given time.

9.4.1 Findings

The model failed after the first month due to the limited capacity of the reservoirs. In response, the initial value of each reservoir was set to zero. However, around the third month and again after the fifth month, the model failed as the LP software was unable to prioritize certain reservoirs over others.

Additionally, the failure was likely due to rainfall data not being entirely accurate, as the limited data available to the general public and the likelihood that the entirety of the reservoir, dam, or lake does not receive 100% of the rainfall (if any) contributed to inaccuracies.

However, the results demonstrated the model's ability to incorporate more complex and external factors, such as rainfall, highlighting the feasibility of this model design in real-world scenarios. By simulating the variability in rainfall patterns and the impact of seasonal monsoons, the model could provide accurate insights into how different sectors – residential, agricultural, and industrial – can be managed to maintain a sustainable water supply.

10 Discussion

10.1 Analysis

10.1.1 Uncertainty and Limitations

Upon conducting the first and second scenarios mentioned above, a flaw was quickly identified in the model. The LP model prioritized using reservoir one over the rest of the reservoirs, irrespective of the amount of water in each reservoir at the time of depletion.

This issue could be mitigated by using an alternative programming library such as `PuLP`, which is not limited to linear variables. Utilizing such a library would also allow for automatic iteration through scenarios, eliminating the need for manual adjustments during each iteration of the linear program.

Furthermore, the absence of key data – such as minimum and maximum flow rates for each reservoir and pipes – hindered the model's accuracy. It is important to note that in this model, the maximum outflow for each reservoir was set at 10% of its maximum capacity. The general public's limited access to such specific data contributes to the lack of precision in real-world simulations, meaning the model could only use assumptions to approximate reality.

10.1.2 Validation and Calibration

The model was run one year prior to the current day, 2024, due to the incomplete availability of 2024 data at the time of analysis. Running the model for 2023 allows for validation of the results with historical data, providing a means to compare the model's output with actual data.

10.2 Future Iterations

In future iterations of this model, the implementation of more datasets and real-world data would enhance accuracy. The current model operates with various assumptions that highly limit its accuracy, such as exact reservoir capacities, pipe flow rates, age of pipes, up-to-date rainfall projections, exact capacities of treatment plants, real-world data for industrial, residential, and agricultural water usage, and the amount of water collected through rainfall.

Additionally, the system failed to account for factors such as the age of pipes, alternate water sources, alternate inflow sources (to lakes, dams, and reservoirs), surface runoff, varying water pressure, and water traffic.

Moreover, the inclusion of more data on seasonal water usage trends, groundwater replenishment rates, and maintenance requirements for the city's infrastructure would allow for a more dynamic and adaptive model. These additional data sources would enable the simulation of more complex scenarios, such as drought periods, peak usage times, and infrastructure failure, making the model a more robust tool for long-term water resource management in Bangalore.

An important insight was realized after the third and final scenario failed due to the limiting capacity of the reservoirs. This highlighted the importance of training the model with a large data set before incorporating external factors.

By training the model, it will have an understanding of when to prioritize certain reservoirs over others during high-demand hours, monsoon seasons, etc.

10.2.1 Incorporating Climate Change

The feasibility and real-world significance of this model would be greatly enhanced by the incorporation of climate change factors. Climate change, with its potential to alter precipitation patterns, increase temperature extremes, and shift monsoon reliability, directly influences water availability and demand in urban areas like Bangalore. Future iterations of the model must account for these evolving environmental conditions to remain relevant and predictive.

Incorporating climate change into the model would require integrating data on expected changes in rainfall intensity, frequency, and distribution, as well as rising temperatures and their effects on evaporation rates and water supply. By simulating different climate scenarios, such as reduced monsoon rainfall or extended dry spells, the model can project how water resource allocation will need to adapt to ensure sustainability.

The real-world implications of such a model would be profound. Policymakers could use it to plan for potential future water shortages, implement preventive measures like water rationing, and develop infrastructure to capture and store water more efficiently. Additionally, by anticipating the impacts of climate change, city planners could prioritize investments in alternative water sources, such as desalination, wastewater treatment, and rainwater harvesting, to create a more resilient water supply network. This would ultimately enhance Bangalore's ability to mitigate and adapt to the looming challenges posed by climate change.

Moreover, the model designed in this paper allows for the incorporation of climate change (although due to the lack of relevant data, it was not further explored in this study). Similar to the third scenario that incorporated rainfall data, the model's design allows for customization down to individual months, weeks, or even days, enabling complex scenarios like climate change to be added to the existing design.

10.2.2 Alternative Water Sources

Another constraint not taken into account in the scenarios presented in this study was the incorporation of lakes and surface runoff. The lack of data to justify the amount of water used, stored, or sourced from these areas was the primary reason for this omission.

Including such sources could improve the accuracy of the model and provide additional supply to meet the city's water demand. The absence of such data likely affected the model's overall accuracy.

10.2.3 Case Study Comparison with Other Cities

A model such as the one presented in this study is not limited to a city like Bangalore and would likely perform better in a study where the government is more transparent about water supply data. This model could be better suited for small suburban areas due to the smaller number of constraints, which would improve its accuracy.

11 Conclusion

The primary objective of this research was to develop and evaluate the feasibility of a linear programming model to optimize fresh-water flow in Bangalore. By simulating various test scenarios – bore water depletion, reservoir-only utilization, and incorporation of rainfall – the model highlighted both the capabilities and limitations of using a linear programming model for the optimization of large data sets.

Throughout the study, the limitations of the linear programming model were addressed, and solutions were proposed accordingly. The three scenarios allowed us to conclude that such a model could be implemented in real-life scenarios. However, it would require large amounts of accurate data on a multitude of external factors to provide a reliable solution.

It was thus concluded that this type of model would be significantly more feasible and reliable for smaller urban or suburban areas, where data constraints are less and the number of variables is more manageable.

Acknowledgments

I would like to express my sincere gratitude to Mr. Emil Kelevedjiev for his invaluable guidance throughout this project. I am also grateful to the organizers and the participants of the 24th Annual Summer Research School in Mathematics and Informatics, where this project was conducted, for providing an inspiring and collaborative environment complete with many hardworking individuals and mentors. Their support made this research paper possible.

References

[1] S. C. Dimri, M. Ram, Flow Maximization Problem as Linear Programming Problem with Capacity Constraints, *Applications and Applied Mathematics: An International Journal (AAM)*, 13:33, 2018.

[2] Henry Adams, Linear Programming Video Series, https://www.youtube.com/watch?v=e313WNw1olk&list=PLDndWhwv4Ujo10_a2T4R4Uqng1nduvfu1, [25/11/2024].

[3] S. Iancheva, E. Kelevedjiev, Linear Programming Approach to Management of Water Resource Systems, *Comptes Rendus de l'Academie Bulgare des Sciences*, 54:25–28, 2001.

[4] Bangalore Water Supply and Sewerage Board (BWSSB), Bangalore Water Supply and Sewerage Board, https://bwssb.gov.in, [25/11/2024].

[5] Ministry of Water Resources, Government of India, Central Ground Water Board (CGWB), https://cgwb.gov.in/en, [25/11/2024].

[6] Ministry of Earth Sciences, Government of India, Indian Meteorological Department (IMD), https://mausam.imd.gov.in/, [25/11/2024].

[7] Ministry of Jal Shakti, Department of Water Resources, Central Water Commission (CWC), http://www.cwc.gov.in/en, [25/11/2024].

[8] LP Solve, A Linear Programming Solver, [25/11/2024], https://lpsolve.sourceforge.net/5.5/

[9] Wikipedia Contributors, Reservoirs and Dams in Karnataka, [25/11/2024], https://en.wikipedia.org/wiki/List_of_dams_and_reservoirs_in_Karnataka

[10] Git Repository, Linear Programming Model for Bangalore, https://github.com/HemalWebsite/Linear-Programming-Model-For-Bangalore, [25/11/2024].