The Sydney Journal of Interdisciplinary Engineering

SJIE 1(3) X-X (2025) https://doi.org/XXXX/XXXX

Water Transfer Pipeline from the Congo River to Northern Africa: Addressing Agricultural Needs and Climate Challenges

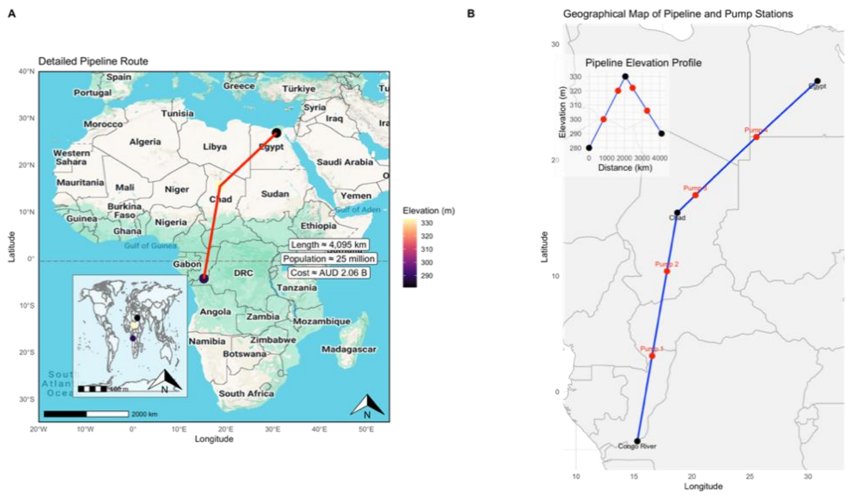
Jingyi Yao [1], Xin Chen[1] and Ruoheng Wang[1]

[1] Faculty of Engineering, University of Sydney, Sydney, Australia

E-mail: jyao0006@sydney.uni.au, xche0627@uni.sydney.edu.au, rwan0669@uni.edu.au

Received xxxxxx
Accepted for publication xxxxxx
Published xxxxxx

Graphical Abstract



The proposed transnational pipeline spans approximately 4,095 km, is expected to benefit 25 million people, and is estimated to cost AUD 2.06 billion.

Abstract

The uneven distribution of global water resources necessitates trans-regional water transfer to alleviate shortages. This study proposes a pipeline from the Congo River through Chad to Egypt to address water scarcity in the Nile Delta. The pipeline spans approximately 4,094 km, featuring four pumping stations equipped with Grundfos NBG 300-250-350/370 centrifugal pumps, with each station housing 22 pumps to maintain a flow rate of 432,000 m³/day. The project could help 25 million people. This design chooses to use solar energy to reduce carbon emissions by approximately 95%. The total investment is estimated at 2.06 billion Australian dollars, covering both capital and operational costs. Geospatial analysis and hydraulic modelling optimize the route, minimizing pressure loss and energy consumption. The study demonstrates feasibility with engineering adjustments and risk management.

Keywords: Transnational Water Transfer, Sustainable Water Management, Hydraulic Modelling, Geospatial Analysis, Climate Impact Assessment, Risk Management

1. Introduction

1.1 Research background

The distribution of global water resources is extremely uneven, and trans-regional water transfer has become an important means to alleviate water shortage[1,2]. Droogers et al. analyzed in North Africa, especially Egypt, water scarcity has severely limited agricultural production and economic development[3]. Egypt's main water source, the Nile, is under pressure from climate change, population growth and competition for water from upstream countries[4,5].

The study proposes to alleviate water scarcity in the Nile Delta by channelling water from the Congo River to Egypt via Chad and promoting regional cooperation and sustainable management, but the project faces significant geographical, technical, environmental and economic challenges that require a systematic feasibility study.

1.2 Challenges

The transnational water diversion project faces multiple challenges, including the need for the pipeline to adapt to the extreme environment of the tropical rainforest, the arid Sahel region and the Sahara Desert. The height difference along the route is large, so the pump station layout and pipeline slope need to be optimized to reduce energy loss[6]. Water sources need to be pre-treated, pipeline materials need to be considered for corrosion resistance, high pressure bearing capacity and wind erosion resistance, and complex geographical conditions must be addressed for construction, maintenance and laying methods[7,8].

It is necessary to analyse temperature changes along the pipeline i.e. equatorial, Sahel and desert climate conditions.

The Congo River basin is the equatorial region, which has a humid climate and abundant rainfall, but the stability of pipeline materials may be affected by high temperature and high humidity[9]. Therefore, corrosion and hydrolysis resistant materials, such as high-density polyethylene (HDPE) or stainless-steel liners, must be used to ensure long-term operation of the pipeline[10].

Chad (the Sahel region) has a dry climate and a large temperature difference between day and night, and the pipeline needs to have a good thermal expansion and cold contraction adaptability and take insulation or heat dissipation measures to reduce material fatigue caused by temperature changes. Egypt's Sahara Desert is extremely hot and UV intense, which can lead to the aging of pipeline materials, and wind erosion will aggravate the surface wear of the pipeline. Therefore, the outer layer should be coated with anti-ultraviolet coating, and design anti-sand measures, such as partially buried pipes or the use of guardrail protection[11].

The route involves a wide range of terrain from the low-lying rainforest of the Congo Basin to the semi-arid plains near Lake Chad to the highlands of the Sahara Desert. Due to terrain changes, water needs to be transported through a multi-stage booster pump station, so the layout of the pump station and energy consumption control need to be considered. How to rationally plan the spacing and head of pumping stations, reduce energy consumption and optimize transportation efficiency is a difficult point. Because the route is long and the terrain varies greatly, it may involve steep sections, which can lead to a sharp increase in pressure inside the pipeline, so pressure buffer systems such as pressure regulators and buffer reservoirs need to be designed to prevent pipeline rupture or leakage.

Because long-distance transportation involves multiple areas, some areas may have geological disasters such as mudslides, earthquakes or dune movements, and it is necessary to carry out detailed geological surveys and take adaptive measures, such as avoiding landslide-prone areas or taking protective reinforcement.

1.3 Research objectives

The main objectives of this study include:

The mathematical model of large-scale water resources transportation is constructed.

Optimize water pipeline route and hydraulic system to improve transportation efficiency.

Combined with technology, structure and economy, the comprehensive feasibility analysis is carried out.

1.4 Current Situation

At present, there are several large-scale water transfer projects for reference. First, China's South-to-North Water Transfer project has proved the technical feasibility of long-distance and large-flow water transfer[12,13]. The California State Water Project in the United States has provided experience in water transfer and reservoir management across climatic zones[14]. The Great Man-made River project in Libya demonstrates the challenges of pipeline construction and maintenance in a desert environment[15]. The Congo Enga hydropower project exemplifies the experience of large-scale infrastructure construction in the Congo region[16].

This study will draw on the technology, optimization methods and policy framework of the above engineering cases to develop a reasonable implementation plan for the Congo-Egypt water pipeline project.

2. Methods

2.1 Geospatial Analysis and Pipeline Route Selection

Geospatial analysis was employed to identify the optimal pipeline route for transnational water transport from Congo to Egypt via Chad. Elevation data were systematically extracted using the elevatr R package, and Geographic Information

Systems (GIS) analyses were conducted using the sf and ggmap R packages. Critical nodes along the pipeline route (Congo, Chad, Egypt) were defined and transformed into spatial (sf) objects to accurately determine elevation values. These spatial data points were then integrated into detailed geospatial visualizations to assess topographical constraints and optimize the pipeline route.

2.2 Hydraulic Modelling and Flow Analysis

First, the design needs to know the total length of the pipe. The length of the pipeline is initially estimated by defining the latitude and longitude of the three key points and connecting these points into a line with R.

This project is mainly aimed at solving agricultural needs and climate challenges, so a medium-sized water supply system is adopted, and the flow rate is designed to be 10 m^3/s[17,18]. Use the continuity equation to solve the pipe diameter:

$$Q = A \times v = \frac{\pi D^2}{4} \times v \quad (1)$$

Water speed choose 1.5m/s to avoid too low to cause deposition, too high to cause friction loss and noise. From this, the pipe diameter D is calculated. In this design, the pressure drop and flow rate are calculated by Darcy-Weisbach equation and Hazan-Williams equation, and the friction factor is solved by Colebroke-White iteration method.

Darcy-Weisbach equation:

$$\Delta P = f \times \frac{L}{D} \times \frac{\rho v^2}{2} \quad (2)$$

Hazen-Williams equation:

$$h_f = 10.67 \times L \times \frac{Q^{1.852}}{C^{1.852} \times D^{4.87}} \quad (3)$$

Colebrook-White iterative method:

$$\frac{1}{\sqrt{f}} = -2\log\left(\frac{\frac{k_S}{D}}{3.7} + \frac{2.51}{Re\sqrt{f}}\right. \quad (4)$$

The Reynolds number is calculated using the diameter (D) and assumed velocity to determine if the flow is turbulent. In turbulent flow, the Colebrook-White iterative formula is applied, and when the difference between the left and right sides is below the set accuracy, the obtained f value is the required one. After finding the friction loss, the pipe pressure drop is calculated using the Darcy-Weisbach equation. The pressure drop is then compared with the Hazen-Williams equation, and the results should align, confirming the calculation's reliability. Finally, the pressure drop is used to estimate the required pumping power.

Sensitivity analysis was then performed. It is assumed that the diameter of the pipe ranges from 2.5 m to 4.0 m, and the flow rate ranges from 5 m^3/s to 15 m^3/s[19], with other parameters remaining unchanged. Use R's ‘expand.grid’ to create a grid for pipe diameter (2.5 to 4.0 m) and flow rate (5 to 15 m^3/s). Then, use ‘mapply’ to calculate pressure drop and pump power for each combination. The result includes two contour plots: one for pressure drop vs. diameter and flow, and the other for pump power changes. The code automatically finds the parameter combination with the lowest pump power and displays the corresponding diameter, flow, pressure drop (bar), and pump power (MW).

The number of pumping stations is calculated according to the obtained results, assuming that each pumping station can provide 3 bars[20], the number of pumping stations n is calculated. To ensure that the pressure loss of each section is relatively uniform, the main pipe length is divided into n+1 parts, and the pumping station is set up respectively. The number of pumps in each pumping station is calculated by the total pressure drop and total flow. The parallel of the pump is used to hit the required flow rate, and the series is used to stack to a higher head. Finally, backup pumps need to be set up at each pumping station.

2.3 Cost and Energy Optimization

Linear programming (LP) is used to solve the optimal economic scheme[21]. Specifically, we use the number of pumps required to be installed at each pump station as a decision variable, and the combined cost of a single pump (including CAPEX and OPEX) as a coefficient of the objective function to minimize the total system cost. They need to meet technical constraints such as flow and head at the same time. Using R's 'lpSolve' package, we build such a mathematical model and solve the most economical configuration scheme.

In this project, the choice of financing mode is crucial to ensure the long-term sustainability and economic efficiency of the project. Therefore, a variety of financing strategies are adopted to adapt to the complexity and needs of the project.

2.4 Solar energy

To improve system sustainability and resilience, solar PV is proposed as a supplementary energy source for the pumping stations. Congo, Chad, and Egypt each receive over 1800 kWh/m^2/year of solar irradiance, ensuring reliable generation. PV lifecycle emissions are as low as 24–50 g CO_2-eq/kWh[22], far below coal-based power (> 900 g CO_2-eq/kWh). Simulated energy payback times are 10.7 years (Congo), 9.3 years (Chad), and 8.5 years (Egypt). Combined with battery storage, PV systems can form microgrids to support off-grid pump operation with greater stability and lower environmental impact.

2.5 Climate Impact Analysis

Firstly, in the data preparation stage, the climate data used in this study came from the WorldClim database, which provided 12-month average temperature (°C) and precipitation (mm). The terra package of R language processed the monthly raster data for annual mean value and obtained the raster data of annual average temperature and annual total precipitation with

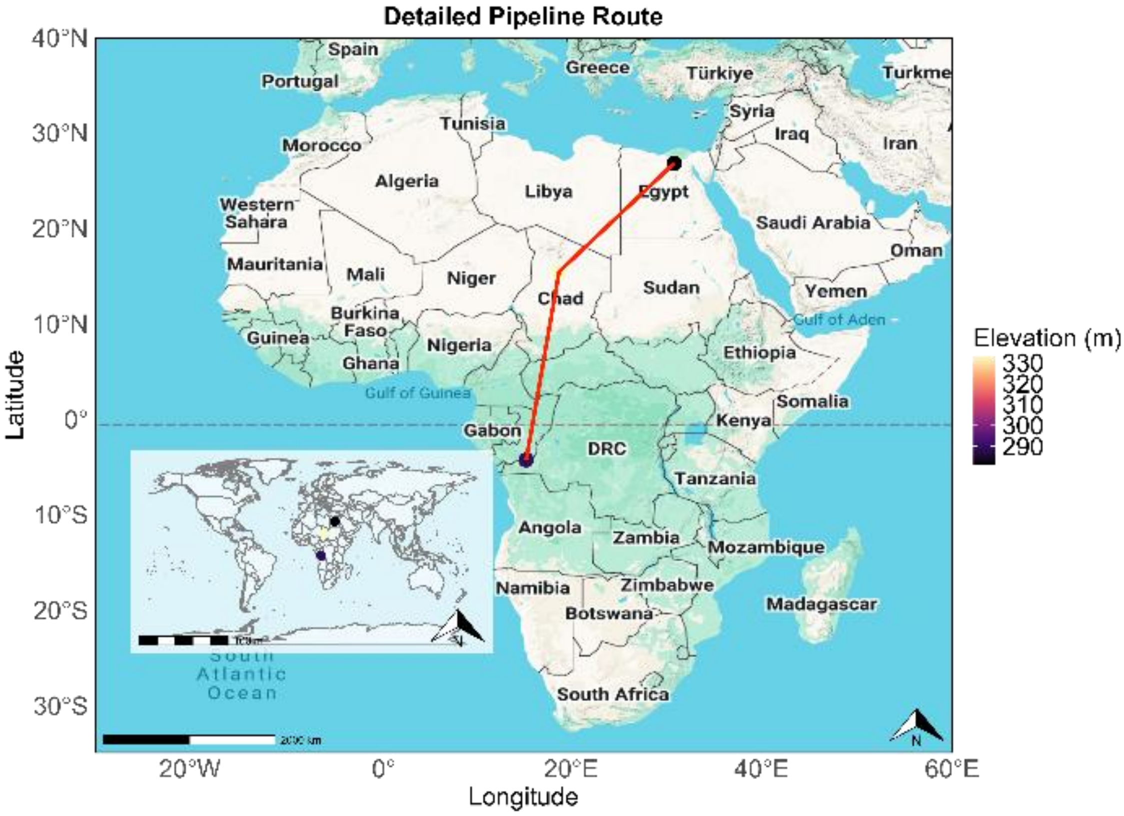


Figure 1. Optimized pipeline route from Congo via Chad to Egypt

high spatial resolution to ensure the accuracy and precision of the analysis results.

To assess possible evaporation losses along the pipeline route, an empirical estimation model based on temperature and precipitation was used:

$$E = k \times (T_{avg} - T_{min}) \times \left[1 - \frac{P_{avg}}{P_{max}}\right] \quad (5)$$

Where E represents annual evaporation (mm/year), T_{avg} and T_{min} are the average annual temperature and total annual precipitation, respectively, T_{min} and P_{max} are the empirical threshold, set at 10°C and 2000 mm, k is the empirical coefficient, with the value 0.7. The model is suitable for estimation of evaporation in arid and semi-arid regions in the absence of detailed meteorological data such as wind speed, humidity, and radiation[23,24] .

Finally, ggplot2 and viridis packages are used to superposition the pipeline path and the evaporation loss grid data, and the spatial distribution of evaporation loss is displayed in a color gradient. The results clearly reflect the variation characteristics of the intensity of evaporation loss along the pipeline, and provide an important reference for the specific design, operation and maintenance of pipeline engineering.

2.6 Risk Analysis

This design identifies the risks in the water conveyance system, including potential faults in the pipeline and pumping station. Prolonged contact with water and corrosive substances can lead to corrosion of metal parts, shortening equipment lifespan[25]. Minerals in the water may form scale, increasing friction and reducing pump efficiency[26]. Microorganisms can create biofilms, affecting flow rate and equipment operation[27]. External risks, such as device aging, external damage, and environmental factors, may also impact the system.

Risk matrixes are then used to demonstrate the severity of different risks and provide a basis for subsequent risk management. In this case, the horizontal axis represents the impact degree of the risk, from "Negligible" to "Severe"; The vertical axis shows the probability of the risk occurring, from "Very Unlikely" to "Very Likely." The risk score is obtained by multiplying the corresponding values of occurrence probability and influence degree, and the value ranges from 1 to 25. Based on the score, the risk level is divided into four levels: A score of 1-3 is Low, a score of 4-6 is Moderate, a score of 7-14 is High, and a score of 15-25 is Extreme.

3. Results and Discussion

3.1 Geospatial and Route Optimization Results

As shown in the results of Figure 1, the optimized pipeline runs from the Congo through Chad to Egypt in a variety of climatic and topographic regions, including tropical rainforests, arid grasslands, and extreme deserts. This cross-

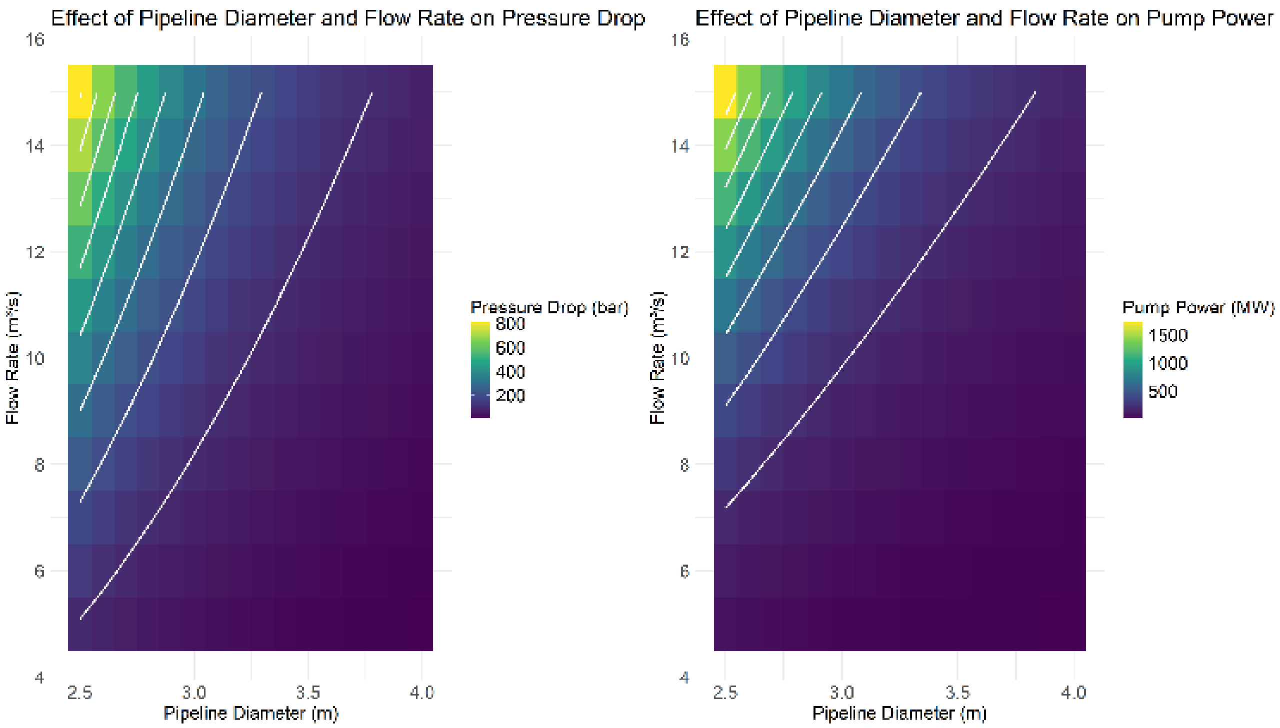


Figure 2. Pressure drop and pump power change with pipe diameter and flow rate respectively

regional design allows water to be transported from the Congo Basin, where rainfall is abundant and water resources are abundant, to the Sahel region (Chad) and the marginal Sahara region (Egypt), where water is scarce and agricultural water is stressed. The pipeline starts in the Congo at a relatively low altitude (about 290 meters), runs north through Chad (the highest altitude is about 330 meters), and ends in Egypt at an altitude of about 300 meters. It is worth noting that the transition from northern Congo to central Chad, accompanied by a gradual uplift of the terrain and a change in climate from wet to arid, requires special attention to hydraulic design (e.g. layout of pumping stations) and material selection.

3.2 Hydraulic Modelling and Pumping Station Design

A line is created from the latitude and longitude data of the key stations (Congo River, Chad, Egypt), and the total length of this line is calculated by R to be 40,946,15 meters. With a designed flow rate of 10 m^3/s and an assumed water velocity of 1.5 m/s, the diameter (D) is approximately 3 meters, based on the continuity equation. The elevation changes are as follows: the Congo River is about 280 meters above sea level (lowest point), Chad is about 330 meters above sea level (highest point), and Egypt is about 290 meters above sea level. Using the Darcy-Weisbach equation, the Hazen-Williams equation, and R code, Figure 2 is obtained.

Through these figures, it can be intuitively seen which design area has the lowest pumping energy consumption, that is, the design point of optimal energy consumption is reached.

Finally, the optimal design points are as follows: Pipeline Diameter = 4 m, Flow Rate = 5 m^3/s, Pressure Drop = 9.304492 bar, Pump Power = 6.65 MW.

The total pressure drop in the design is approximately 9.30 bar, while the system's pumping power requirement is 6.65 MW. Assuming that each pumping station is designed to provide a lift of approximately 3 bars[20], then $n \approx 4$.

Four pumping stations were initially designed on the pipeline to ensure the normal operation of the system. For the location of the pumping stations, it is assumed that the pumping stations are evenly distributed throughout the pipeline so that each station can bear a similar pressure loss. With the help of R, the distribution map of the pumping station and the corresponding elevation Figure 3 are drawn.

There are many types of pumps used in the market, such as positive displacement pumps, dynamic pumps and centrifugal pumps. The water pumped in this design is mainly used for agricultural, municipal and industrial purposes. According to the above conclusions, the flow rate is 5 m^3/s (18000 m^3/h), and the total pressure drop is about 9bar, which belongs to the condition of large flow and medium head. Centrifugal pumps are very good at handling large flows and can cover heads from a few meters to hundreds of meters[28]. Therefore, a centrifugal pump is selected in this design.

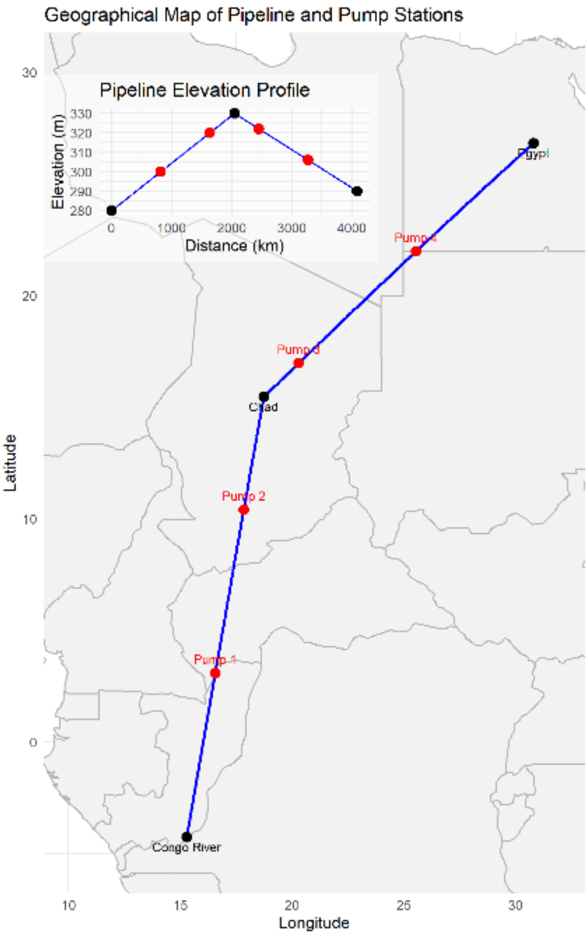


Figure 3. Distribution map of pumping station and corresponding elevation map

This design selects Grundfos NBG 300-250-350/370, whose parameters are: Rated flow: 916.5 m³/h, Rated lift: 28.86m (about 2.886 bar), Motor power: 90 kW, Speed: 1488 rpm, Price: About 63,046 AUD[29].

In this design, we required a total flow rate of 5 m³/s (i.e. 18,000 m³/h) with a total pressure drop of approximately 9.3 bar (approximately 93 m head). There are a total of 4 pumping stations, each of which distributes the total pressure drop equally, and each station bears a pressure drop of 2.325bar.

In order to achieve the total flow, it is necessary to parallel the pump, the number of parallel units is:

$$n_1 = \frac{18000}{916.5} \approx 20 \quad (7)$$

In order to achieve the total pressure drop, the pumps need to be connected in series, and only one pump in series is required in a pumping station.

Therefore, the total number of pumps required for each station is 20. Two backup pumps are placed at each pump station, so a total of 88 pumps are required.

3.3 Cost and Energy Optimization Results

To achieve the optimization of the system cost and energy consumption, a linear programming approach is used to solve the pumping station configuration scheme. The total system flow requirement is 5 m³/s (18,000 m³/h) as well as the rated flow of a single pump of 916.5 m³/h. Use this to establish flow constraints for each pump station. Number of pumps installed per pumping station: $x_i \times 916.5 \geq 18000$.

The minimum number of pumps per station is 20. This study considers both capital expenditure (CAPEX) and operating expenditure (OPEX) to evaluate long-term investment and maintenance costs. CAPEX includes the cost of purchasing and installing pumps, while OPEX covers operation, maintenance, repair, and energy use. Since CAPEX is a one-time cost, it is generally higher than OPEX. High-performance, energy-efficient centrifugal pumps are chosen despite their higher initial cost, as they offer better efficiency and a longer lifespan, ultimately reducing operating costs[30]. Table 1 shows the estimated data for each major cost item in a single pumping station. The total investment of this project is USD 1.34 billion, which is approximately 59%, or 0.59 times, the cost of Saudi Arabia's independent water transmission pipeline project (USD 2.26 billion)[31].

Table 1. Cost estimation (CAPEX of Pumps, piping material and install, OPEX of maintenance and energy consumption)

Cost items		Detailed description	Unit price (AUD)	Quantity	Total price (AUD)
CAPEX	Pumps	Purchase of centrifugal pump	63,046[29]	88	5,550,000
	Piping material	HDPE and stainless-steel pipes. The cost of other materials	HDPE:75/m Steel:200/m	4094615m	563,000,000
	Install	Install pipes and related equipment	350/m	4094615m	1,434,000,000
OPEX	Maintenance	Maintenance costs for pipework and pumping units	3% of the total cost of pipeline construction	/	60,050,000
	Energy consumption	Energy requirements for pumps	6.65 MW/h 0.25/kWh	8760h	14,500
Total (AUD)					2,062,000,000

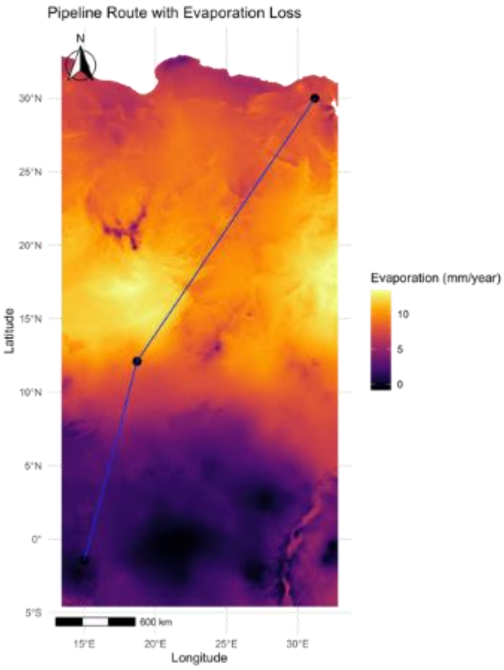


Figure 4. Spatial Distribution of Estimated Annual

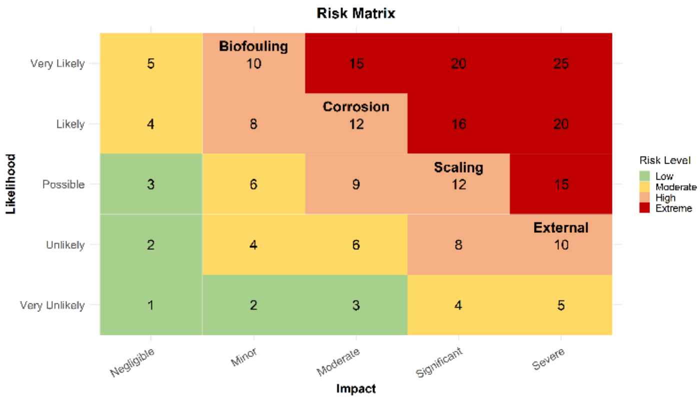


Figure 5. Risk Matrix

The project needs financial support from the government. The participation of the government brings a stable source of funding for the project and improves the protection of public interest[32]. Second, the public-private partnership (PPP) helps to spread risk, and through private investment increased financial resources, make the project more advanced and reliable in technology and operation[33]. To ensure economic independence during the operational phase, the project adopted a usage-based billing system. This promotes the rational use of water resources and provides a stable source of income for the ongoing operation of the project.

3.4 Solar PV System Cost Estimation

To reduce operational energy costs, each pump station is designed to incorporate a solar PV system sized according to local solar availability. Using 450 W panels and an 80% system efficiency, the estimated requirements are approximately 24,630 panels in Congo, 21,390 in Chad, and 18,472 in Egypt. At a unit cost of 208.94 dollars[34], the corresponding panel procurement costs are AUD 5.15 million, 4.47 million, and 3.86 million, respectively. These figures support the feasibility of integrating solar farms to improve energy autonomy and sustainability across the pipeline system.

Calculation results show that the pump station power 6.65 MW daily operation for 24 hours, the use of coal annual emissions of 52,428.6 tons of CO_2-eq, while the annual emission of solar energy 1398.1-2912.7 tons of CO_2-eq, saving carbon emissions 94.4%-97.3%.

3.5 Climate Impact

The spatial distribution of evaporation losses along the proposed transnational pipeline was assessed using climate data and previously described empirical evaporation models. The resulting map (Figure 4) shows the annual evaporation intensity of the pipeline from the Congo via Chad to Egypt (in mm/year). It can be seen from the figure that there is a significant spatial gradient in evaporation losses along the proposed pipeline route. In the southern part of the pipeline, particularly in Congo, the annual evaporation remains relatively modest, with values generally below 5 mm/year. As the pipeline extends north into Chad and Sudan, evaporation intensity increases gradually. However, even in the northernmost parts of Egypt, the simulated annual evaporation remains below 12 mm/year.

This result contradicts earlier assumptions that evaporation in the extremely arid Sahara region could exceed 2000 mm/year. The lower simulated values suggest that actual climate-driven water losses, constrained by mean temperature and annual precipitation, may be milder than initially expected.

Although the simulated evaporation rate is relatively low, the cumulative impact on more than 3,000 km of pipeline remains an important engineering consideration. Measures such as buried underground pipes, anti-evaporation materials or night-time pumping plans may further reduce losses and improve long-term operational efficiency. In addition, the model highlights the importance of using spatially explicit data rather than broad assumptions when conducting climate impact assessments on large water infrastructure. 3.6 Failure Risk

For the risk matrix (Figure 5) mentioned above:

Corrosion events, with a risk score of 16, fall in the Extreme zone, indicating high likelihood and significant impact. Prevention and maintenance should be prioritized[25].

Scaling events fall in the High category with a risk score of 9, indicating a moderate probability and impact. They can be managed through regular cleaning and water quality pretreatment[35].

Bioblockages have a moderate risk score of 6, indicating a low risk. While the risk is minor, it can be reduced through regular monitoring and cleaning[36,37].

External risk, with a risk score of 10, falls in the High zone. Key equipment should be regularly tested and updated to prevent aging[38]. Security measures and an emergency plan should also be strengthened, with real-time monitoring of weather and environmental changes to adjust operations and minimize the impact of natural disasters or extreme weather.

4. Conclusion

The water transfer pipeline project from the Congo River to North Africa is technically feasible and can effectively alleviate the water shortage problem in the target area. However, project implementation requires consideration of multiple engineering trade-offs, including environmental and water quality concerns, energy and power requirements, and engineering safety and financial risks.

Specifically, environmental and water quality issues need to be addressed through the selection of suitable materials and filtration technologies to protect water quality and reduce environmental impact. Energy efficiency can be improved by integrating renewable energy systems, thereby reducing dependence on traditional energy sources. Engineering safety needs to be ensured through precise material selection and construction techniques to prevent structural risks and maintenance failures. Finally, for long-term investment and operating costs, continuous financial planning and cost management strategies are recommended. Future work should focus on improving the durability and corrosion resistance of pipeline materials, as well as developing more efficient pumps and power systems to improve the economy and reliability of the entire water delivery system.

5. Acknowledgements

The authors of this study acknowledge that AI is used to design and modify R code and plays a key role in subsequent data visualisation.

6. Author contributions

Jingyi Yao was responsible for writing 2.2, 2.3 and 2.6 sections in the Methods. In the Results and Discussion section, Jingyi Yao wrote 3.2, 3.3, and 3.6 sections. Jingyi Yao also crafted the Conclusion section and undertook the formatting adjustments for the manuscript. At the same time, the R code corresponding to figure 2, 3, and 5 was also completed by her.

Xin Chen was responsible for writing the 2.1 and 2.5 sections in the Methods. In the "Results and Discussion" section, Xin Chen contributed to 3.1 and 3.5. Xin Chen was also responsible for developing the corresponding R code for both the geospatial and climate impact analyses.

Ruoheng Wang completed the Introduction of the section 1.1, 1.2, 1.3, 1.4, 2.4 and 3.4. She also completed the modification of the Abstract and provided suggestions for the subsequent pipeline establishment

Dr. Gobinath Rajarathnam provided conceptual guidance and project supervision, ensuring the research direction and quality adhered to academic standard.

References

[1] Han, M., Chen, G. & Li, Y. Global water transfers embodied in international trade: Tracking imbalanced and inefficient flows. *Journal of cleaner production* **184**, 50-64 (2018).

[2] Kazumba, S., Gillerman, L. & Oron, G. Managing Trans-Regional Water Transfer in Dry African Countries to Mitigate Shortages. *Journal of Innovative Research in Engineering and Sciences* **2**, 249-273 (2011).

[3] Droogers, P. *et al.* Water resources trends in Middle East and North Africa towards 2050. *Hydrology and Earth System Sciences* **16**, 3101-3114 (2012).

[4] Ashour, M., El Attar, S., Rafaat, Y. & Mohamed, M. Water resources management in Egypt. *JES. Journal of Engineering Sciences* **37**, 269-279 (2009).

[5] Abd El Moniem, A. A. Overview of water resources and requirements in Egypt; the factors controlling its management and development. *Journal of Environmental Studies* **2**, 82-97 (2009).

[6] Gong, Y. & Cheng, J. Optimization of cascade pumping stations' operations based on head decomposition–dynamic programming aggregation method considering water level requirements. *Journal of Water Resources Planning and Management* **144**, 04018034 (2018).

[7] Skretas, A., Gyftakis, S. & Marcoulaki, E. A demonstration of sustainable pipeline routing optimization using detailed financial and environmental assessment. *Journal of Cleaner Production* **362**, 132305 (2022).

[8] Almheiri, Z., Meguid, M. & Zayed, T. Review of Critical Factors Affecting the Failure of Water Pipeline Infrastructure. *Journal of Water Resources Planning and Management* **149**, 03123001 (2023).

[9] Toulmin, C. *Climate change in Africa*. (Bloomsbury Publishing, 2009).

[10] Nguyen, K. Q. *et al.* Long-term testing methods for HDPE pipe-advantages and disadvantages: A review. *Engineering Fracture Mechanics* **246**, 107629 (2021).

[11] Condini, A. UV-CURABLE PROTECTIVE COATING FOR THE INNER SURFACE OF STEEL PIPES. (2024).

[12] Zhao, Z.-Y., Zuo, J. & Zillante, G. Transformation of water resource management: a case study of the South-to-North Water Diversion project. *Journal of cleaner production* **163**, 136-145 (2017).

[13]Tang, C., Yi, Y., Yang, Z. & Cheng, X. Water pollution risk simulation and prediction in the main canal of the South-to-North Water Transfer Project. *Journal of Hydrology* **519**, 2111-2120 (2014).
[14]Chung, I. & Helweg, O. Modeling the California state water project. *Journal of Water Resources Planning and Management* **111**, 82-97 (1985).
[15]Kuwairi, A. in *Proceedings of the Institution of Civil Engineers-Civil Engineering.* 39-43 (Thomas Telford Ltd).
[16]Oyewo, A. S., Farfan, J., Peltoniemi, P. & Breyer, C. Repercussion of large scale hydro dam deployment: the case of Congo Grand Inga hydro project. *Energies* **11**, 972 (2018).
[17]Mays, L. W. *Water resources engineering*. (John Wiley & Sons, 2010).
[18]Nitivattananon, V., Sadowski, E. C. & Quimpo, R. G. Optimization of water supply system operation. *Journal of water resources planning and management* **122**, 374-384 (1996).
[19]Stephenson, D. *Pipeline design for water engineers*. Vol. 40 (Elsevier, 1989).
[20]Menon, E. S. *Working Guide to Pump and Pumping Stations: Calculations and Simulations*. (Gulf Professional Publishing, 2009).
[21]Vaccari, M., Mancuso, G., Riccardi, J., Cantù, M. & Pannocchia, G. A sequential linear programming algorithm for economic optimization of hybrid renewable energy systems. *Journal of process control* **74**, 189-201 (2019).
[22]Fthenakis, V. M., Kim, H. C. & Alsema, E. Emissions from photovoltaic life cycles. *Environmental science & technology* **42**, 2168-2174 (2008).
[23]Xiong, Y. J., Zhao, W. L., Wang, P., Paw U, K. T. & Qiu, G. Y. Simple and applicable method for estimating evapotranspiration and its components in arid regions. *Journal of Geophysical Research: Atmospheres* **124**, 9963-9982 (2019).
[24]Awal, R., Habibi, H., Fares, A. & Deb, S. Estimating reference crop evapotranspiration under limited climate data in West Texas. *Journal of Hydrology: Regional Studies* **28**, 100677 (2020).
[25]Hussein Farh, H. M., Ben Seghier, M. E. A., Taiwo, R. & Zayed, T. Analysis and ranking of corrosion causes for water pipelines: A critical review. *NPJ Clean Water* **6**, 65 (2023).
[26]Kazi, S., Duffy, G. & Chen, X. Mineral scale formation and mitigation on metals and a polymeric heat exchanger surface. *Applied Thermal Engineering* **30**, 2236-2242 (2010).
[27]Vishwakarma, V. Impact of environmental biofilms: Industrial components and its remediation. *Journal of basic microbiology* **60**, 198-206 (2020).
[28]Karassik, I. & McGuire, J. T. *Centrifugal pumps*. (Springer Science & Business Media, 2012).
[29]Grundfos. *NBG 300-250-350/370 technical specifications*, < https://product-selection.grundfos.com/au/products/nbg-nbge/nbg/nbg-300-250-350370-98307051?pumpsystemid=2625522505&tab=variant-specifications> (
[30]Kaya, D., Çanka Kılıç, F. & Öztürk, H. H. in *Energy Management and Energy Efficiency in Industry: Practical Examples* 329-374 (Springer, 2021).
[31]World, I. *Agreements signed for US$22.6 billion independent water transmission pipeline in Saudi Arabia*, <https://www.infrapppworld.com/news/agreements-signed-for-us226-billion-independent-water-transmission-pipeline-in-saudi-arabia> (
[32]Anderson, M. B., Petrie, M., Alier, M. M., Cangiano, M. M. & Hemming, M. R. *Public-private partnerships, government guarantees, and fiscal risk*. (International Monetary Fund, 2006).
[33]Grimsey, D. & Lewis, M. K. in *Public Private Partnerships* (Edward Elgar Publishing, 2004).
[34]SolarOnline. *Solar Online Australia*, <https://solaronline.com.au/solar-panels.html?srsltid=AfmBOoqW_orA8GK32qazMwM1-qUj5MJKzmd-_egge_PacruN6EVsfS9d> (
[35]Sutzkover-Gutman, I. & Hasson, D. Feed water pretreatment for desalination plants. *Desalination* **264**, 289-296 (2010).
[36]Flemming, H.-C. Microbial biofouling: unsolved problems, insufficient approaches, and possible solutions. *Biofilm highlights*, 81-109 (2011).
[37]Shi, X., Tal, G., Hankins, N. P. & Gitis, V. Fouling and cleaning of ultrafiltration membranes: A review. *Journal of Water Process Engineering* **1**, 121-138 (2014).
[38]Kraak, D. *et al.* in *2018 IEEE 23rd European Test Symposium (ETS).* 1-10 (IEEE).

Supplementary

R code of Figure 1:

```
 # ===== Install and load required R packages (if not already installed) =====
packages <- c("ggplot2", "ggmap", "sf", "elevatr", "ggspatial", "dplyr", "gridExtra")
new_packages <- packages[!(packages %in% installed.packages()[,"Package"])]
if(length(new_packages)) install.packages(new_packages)

library(ggplot2)
library(ggmap)
library(sf)
library(elevatr)
library(ggspatial)
library(dplyr)
library(gridExtra)
library(cowplot)

# ===== Set your Google API Key (replace with your actual API Key) =====
register_google(key = "AIzaSyB__tXcP0Rp5XwCw8MfRcxDhkVZiJuZQ7U")  # Replace with your API Key

# 1. Define pipeline stations (Congo → Chad → Egypt)
locations <- data.frame(
  name = c("Congo River", "Chad", "Egypt"),
  lat = c(-4.267, 15.4542, 26.8206),
  lon = c(15.283, 18.7322, 30.8025)
)

# 2. Convert to sf object
locations_sf <- st_as_sf(locations, coords = c("lon", "lat"), crs = 4326)

# 3. Retrieve DEM (elevation) data
elev_data <- get_elev_point(locations_sf, prj = st_crs(4326), src = "aws")
locations_sf$elevation <- elev_data$elevation  # Add elevation info

# 4. Create pipeline line (connect stations)
pipeline_line <- st_sfc(st_linestring(as.matrix(locations[, c("lon", "lat")])), crs = 4326)
pipeline_sf <- st_sf(geometry = pipeline_line)

# 5. Retrieve basemap
bbox_africa <- c(left = -20, bottom = -35, right = 55, top = 40)
map_base_africa <- get_googlemap(center = c(lon = 17.5, lat = 2.5), zoom = 3, maptype = "terrain")

# 6. Plot global pipeline map
region_map <- ggmap(map_base_africa) +
  geom_sf(data = locations_sf, aes(color = elevation), size = 5, inherit.aes = FALSE) +
  geom_sf(data = pipeline_sf, color = "red", linewidth = 1.5, inherit.aes = FALSE) +
  scale_color_viridis_c(option = "magma", name = "Elevation (m)", guide = guide_colorbar(position = "right")) +
  coord_sf(xlim = c(-20, 55), ylim = c(-35, 40), expand = FALSE) +
  labs(title = "Detailed Pipeline Route", x = "Longitude", y = "Latitude") +
  annotation_scale(location = "bl", width_hint = 0.3) +
  annotation_north_arrow(location = "br", which_north = "true", height = unit(1, "cm")) +
  theme_minimal() +
  theme(
```

```
    legend.position = c(0.85, 0.2),
    panel.border = element_rect(color = "black", linewidth = 1, fill = NA))

# 7. Plot elevation inset map

# Ensure `locations` includes elevation data
locations$elevation <- elev_data$elevation

# Use the data frame `locations` instead of `locations_sf`
world_map <- ggplot() +
  borders("world", colour = "gray50", fill = "white", alpha = 0.5) +  # 50% transparent background
  geom_point(data = locations, aes(x = lon, y = lat, color = elevation), size = 3) +
  scale_color_viridis_c(option = "magma", name = "Elevation (m)", guide = "none") +  # Remove legend for inset
  theme_void() +
  annotation_scale(location = "bl", width_hint = 0.3) +
  annotation_north_arrow(location = "br", which_north = "true", height = unit(1, "cm")) +
  theme(
    legend.position = "none",  # Hide legend in inset
    panel.background = element_rect(fill = alpha("white", 0.5), color = "black"),  # 50% transparent background
    plot.background = element_rect(fill = alpha("white", 0.5), color = "black")  # Maintain border visibility
  )

# 8. Combine map and elevation inset
final_plot <- ggdraw() +
  draw_plot(region_map, x = 0, y = 0, width = 1, height = 1) +
  draw_plot(world_map, x = 0.15, y = 0.2, width = 0.25, height = 0.25)  # Adjust inset placement

# 9. Display the final plot
print(final_plot)

# ====== Save to PNG with 10:10 ratio ======
ggsave("combined_pipeline_plot_16x9.png", plot = final_plot, width = 10, height = 10, dpi = 400)
```

R code of Figure 2:

```
# Install and load required packages
if (!require(ggplot2)) install.packages("ggplot2")
if (!require(gridExtra)) install.packages("gridExtra")
if (!require(viridis)) install.packages("viridis")
library(ggplot2)
library(gridExtra)
library(viridis)

# ======= Fixed Parameters =======
L    <- 4094615      # Total pipeline length in meters
ks   <- 0.0001       # Pipe roughness in meters
nu   <- 1.007e-6     # Kinematic viscosity of water in m²/s
rho  <- 1000         # Density of water in kg/m³
g    <- 9.81         # Gravitational acceleration in m/s²
efficiency <- 0.7    # Pump efficiency (dimensionless)

# ======= Colebrook-White Iterative Function =======
colebrook <- function(Re, ks, D, tol = 1e-6, max_iter = 100) {
  f <- 0.02  # Initial guess for friction factor
```

```
  for (i in 1:max_iter) {
    lhs <- 1 / sqrt(f)
    rhs <- -2 * log10((ks / (3.7 * D)) + (2.51 / (Re * sqrt(f))))
    if (abs(lhs - rhs) < tol) break
    f <- 1 / (rhs^2)
  }
  return(f)
}

# ======= Function to Calculate Pressure Drop =======
calc_pressure_drop <- function(D, Q) {
  A <- pi * (D / 2)^2                # Cross-sectional area in m²
  v <- Q / A                         # Average flow velocity in m/s
  Re <- (v * D) / nu                 # Reynolds number
  f <- colebrook(Re, ks, D)          # Friction factor
  deltaP <- f * (L / D) * (rho * v^2 / 2)  # Pressure drop in Pascals
  return(deltaP)
}

# ======= Function to Calculate Pump Power =======
calc_pump_power <- function(D, Q) {
  deltaP <- calc_pressure_drop(D, Q)
  pump_power <- Q * deltaP / efficiency  # Pump power in Watts
  return(pump_power)
}

# ======= Sensitivity Analysis =======
D_values <- seq(2.5, 4.0, by = 0.1)      # Pipeline diameters (m)
Q_values <- seq(5, 15, by = 1)           # Flow rates (m³/s)
param_grid <- expand.grid(D = D_values, Q = Q_values)

param_grid$PressureDrop <- mapply(calc_pressure_drop, param_grid$D, param_grid$Q)
param_grid$PumpPower <- mapply(calc_pump_power, param_grid$D, param_grid$Q)

# Convert to more readable units
param_grid$PumpPower_MW <- param_grid$PumpPower / 1e6         # MW
param_grid$PressureDrop_bar <- param_grid$PressureDrop / 1e5  # bar

# ======= Plot: Pressure Drop =======
p1 <- ggplot(param_grid, aes(x = D, y = Q, z = PressureDrop_bar)) +
  geom_tile(aes(fill = PressureDrop_bar)) +
  geom_contour(color = "white", size = 0.8) +
  scale_fill_viridis_c(name = "Pressure Drop (bar)") +
  labs(title = "Effect of Pipeline Diameter and Flow Rate on Pressure Drop",
       x = "Pipeline Diameter (m)",
       y = "Flow Rate (m³/s)") +
  theme_minimal()

# ======= Plot: Pump Power =======
p2 <- ggplot(param_grid, aes(x = D, y = Q, z = PumpPower_MW)) +
  geom_tile(aes(fill = PumpPower_MW)) +
  geom_contour(color = "white", size = 0.8) +
  scale_fill_viridis_c(name = "Pump Power (MW)") +
```

```
  labs(title = "Effect of Pipeline Diameter and Flow Rate on Pump Power",
       x = "Pipeline Diameter (m)",
       y = "Flow Rate (m³/s)") +
  theme_minimal()

# ======= Show Both Plots One Above the Other =======
final_combined_plot <- grid.arrange(p1, p2, ncol = 2)

# ======= Save to PNG with 16:9 ratio =======
ggsave("combined_pipeline_plots_16x9.png", plot = final_combined_plot, width = 16, height = 9, dpi = 300)

# ======= Find Optimal Design (Minimum Pump Power) =======
min_idx <- which.min(param_grid$PumpPower_MW)
optimal <- param_grid[min_idx, ]
cat("Optimal Design:\n")
cat("Pipeline Diameter =", optimal$D, "m\n")
cat("Flow Rate =", optimal$Q, "m³/s\n")
cat("Pressure Drop =", optimal$PressureDrop_bar, "bar\n")
cat("Pump Power =", optimal$PumpPower_MW, "MW\n")
```

R code of Figure 3:

```
# ====== Ensure Required Packages Are Installed ======
required_packages <- c("sf", "ggplot2", "cowplot", "maps")

for (pkg in required_packages) {
  if (!require(pkg, character.only = TRUE)) {
    install.packages(pkg)
    library(pkg, character.only = TRUE)
  }
}

library(sf)
library(ggplot2)
library(cowplot)
library(maps)

# ------------------- 1. Define Key Locations and Create Pipeline Line -------------------
locations <- data.frame(
  name = c("Congo River", "Chad", "Egypt"),
  lon  = c(15.283, 18.7322, 30.8025),
  lat  = c(-4.267, 15.4542, 26.8206)
)

locations_sf <- st_as_sf(locations, coords = c("lon", "lat"), crs = 4326)
pipeline_line <- st_sfc(st_linestring(as.matrix(locations[, c("lon", "lat")])), crs = 4326)

# ------------------- 2. Example Elevation Profile Data -------------------
L_total <- 4094615  # Total pipeline length in meters
dist_seq <- seq(0, L_total, length.out = 1000)
control_dist <- c(0, L_total/2, L_total)
control_elev <- c(280, 330, 290)
elev_profile <- approx(x = control_dist, y = control_elev, xout = dist_seq)$y
```

```
elev_df <- data.frame(
  Distance_km = dist_seq / 1000,
  Elevation_m = elev_profile
)

# Key locations on the elevation profile (marked in black)
locations_elev <- data.frame(
  name = c("Congo River", "Chad", "Egypt"),
  Distance_km = c(0, (L_total/2)/1000, L_total/1000),
  Elevation_m = c(280, 330, 290)
)

# ------------------- 3. Determine Pump Station Positions on the Elevation Profile -------------------
# Assume 4 pump stations located at 1/5, 2/5, 3/5, 4/5 of the pipeline length
pump_station_fraction <- c(1/5, 2/5, 3/5, 4/5)
pump_station_dist <- pump_station_fraction * L_total
pump_station_elev <- approx(x = dist_seq, y = elev_profile, xout = pump_station_dist)$y

pump_elev_df <- data.frame(
  Distance_km = pump_station_dist / 1000,
  Elevation_m = pump_station_elev,
  station     = paste("Pump", 1:length(pump_station_elev))
)

# ------------------- 4. Create the Elevation Profile Plot (p_elev) -------------------
p_elev <- ggplot(elev_df, aes(x = Distance_km, y = Elevation_m)) +
  geom_line(color = "blue") +
  # Key locations (black points)
  geom_point(data = locations_elev, aes(x = Distance_km, y = Elevation_m),
             color = "black", size = 3) +
  # Pump stations (red points)
  geom_point(data = pump_elev_df, aes(x = Distance_km, y = Elevation_m),
             color = "red", size = 3) +
  labs(title = "Pipeline Elevation Profile",
       x = "Distance (km)", y = "Elevation (m)") +
  theme_minimal() +
  # Set background to white with 50% transparency
  theme(
    plot.background  = element_rect(fill = alpha("white", 0.5), color = NA),
    panel.background = element_rect(fill = alpha("white", 0.5), color = NA)
  )

# ------------------- 5. Prepare Map Data: Pipeline and Pump Stations -------------------
world_map <- map_data("world")

# Sample pipeline points for the route (blue line)
pipeline_line_proj <- st_transform(pipeline_line, 3857)
pipeline_points_proj <- st_line_sample(pipeline_line_proj, sample = seq(0, 1, length.out = 1000))
pipeline_points <- st_transform(pipeline_points_proj, 4326)
pipeline_coords <- st_coordinates(pipeline_points)
pipeline_df <- data.frame(pipeline_coords)
names(pipeline_df) <- c("lon", "lat")
```

```
# Example pump stations on the geographic map
pump_stations_proj <- st_line_sample(pipeline_line_proj, sample = pump_station_fraction)
pump_stations <- st_transform(pump_stations_proj, 4326)
pump_stations_coords <- st_coordinates(pump_stations)
pump_stations_df <- data.frame(
  lon = pump_stations_coords[,1],
  lat = pump_stations_coords[,2],
  station = paste("Pump", 1:nrow(pump_stations_coords))
)

# ------------------- 6. Create the Geographical Map Plot (p_map) -------------------
xlim_vals <- c(10, 32)
ylim_vals <- c(-5, 30)

p_map <- ggplot() +
  geom_polygon(data = world_map, aes(x = long, y = lat, group = group),
               fill = "gray95", color = "gray70") +
  geom_path(data = pipeline_df, aes(x = lon, y = lat),
            color = "blue", size = 1) +
  geom_point(data = pump_stations_df, aes(x = lon, y = lat),
             color = "red", size = 3) +
  geom_text(data = pump_stations_df, aes(x = lon, y = lat, label = station),
            color = "red", vjust = -1, size = 3) +
  geom_point(data = locations, aes(x = lon, y = lat),
             color = "black", size = 3) +
  geom_text(data = locations, aes(x = lon, y = lat, label = name),
            color = "black", vjust = 1.5, size = 3) +
  coord_fixed(ratio = 1, xlim = xlim_vals, ylim = ylim_vals) +
  labs(title = "Geographical Map of Pipeline and Pump Stations",
       x = "Longitude", y = "Latitude") +
  theme_minimal()

# ------------------- 7. Combine the Plots: Embed p_elev into p_map -------------------
final_plot <- ggdraw() +
  draw_plot(p_map, x = 0, y = 0, width = 1, height = 1) +
  # Embed the elevation plot at the left-top corner (adjust x, y, width, height as needed)
  draw_plot(p_elev, x = 0.2, y = 0.65, width = 0.25, height = 0.25)

# Display the final combined plot
print(final_plot)

# ------------------- 7. Combine the Plots: Embed p_elev into p_map -------------------
final_plot <- ggdraw() +
  draw_plot(p_map, x = 0, y = 0, width = 1, height = 1) +
  draw_plot(p_elev, x = 0.35, y = 0.72, width = 0.2, height = 0.2)

# Display the final combined plot
print(final_plot)

# ====== Save to PNG with 16:9 ratio ======
ggsave("combined_pipeline_plot_16x9.png", plot = final_plot, width = 16, height = 9, dpi = 300)
```

R code of Figure 4:

```
# ===== 1. Install and load all required packages =====
install.packages(c("terra", "sf", "ggplot2", "ggspatial", "viridis", "rnaturalearth", "rnaturalearthdata"))

library(terra)          # For handling raster data
library(sf)             # For handling spatial vector data
library(ggplot2)        # Core plotting package
library(ggspatial)      # For scale bar and north arrow
library(viridis)        # High-quality color scales
library(rnaturalearth)   # Optional background geographic data
library(rnaturalearthdata)

# ===== 2. Set your data directory path =====
downloads_path <- "~/Downloads/"  #  Modify based on your actual file path

# ===== 3. Load temperature and precipitation raster data ( please change filepath to your computer specifically
) =====
temp_files <- sort(list.files(path = downloads_path, pattern = "wc2.1_30s_tavg_.*\\.tif$", full.names = TRUE))
precip_files <- sort(list.files(path = downloads_path, pattern = "wc2.1_30s_prec_.*\\.tif$", full.names = TRUE))

# Read raster stacks
temp_raster <- rast(temp_files)
precip_raster <- rast(precip_files)

# ===== 4. Calculate annual mean temperature and total precipitation =====
avg_temp_raster <- mean(temp_raster)            # °C
total_precip_raster <- sum(precip_raster)       # mm/year

# ===== 5. Evaporation loss estimation model =====
k <- 0.7
T_min <- 10
P_max <- 2000

evap_raster <- k * (avg_temp_raster - T_min) * (1 - (total_precip_raster / P_max))

# ===== 6. Define pipeline stations (Congo → Chad → Egypt) =====
pipeline_stations <- data.frame(
  station = c("Congo", "Chad", "Egypt"),
  lat = c(-1.45, 12.1, 30.0),
  lon = c(15.0, 18.7, 31.2)
)

pipeline_sf <- st_as_sf(pipeline_stations, coords = c("lon", "lat"), crs = 4326)
pipeline_route <- st_linestring(matrix(unlist(pipeline_stations[, c("lon", "lat")]), ncol = 2, byrow = FALSE))
pipeline_route_sf <- st_sfc(pipeline_route, crs = 4326)

# ===== 7. Expand bounding box and crop evaporation raster =====
pipeline_bbox <- st_bbox(pipeline_sf)
expand_factor <- 0.1
x_range <- (pipeline_bbox$xmax - pipeline_bbox$xmin) * expand_factor
y_range <- (pipeline_bbox$ymax - pipeline_bbox$ymin) * expand_factor

expanded_bbox <- terra::ext(
  pipeline_bbox$xmin - x_range,
```

```
  pipeline_bbox$xmax + x_range,
  pipeline_bbox$ymin - y_range,
  pipeline_bbox$ymax + y_range
)

evap_raster_expanded <- crop(evap_raster, expanded_bbox)

# ===== 8. Convert raster to dataframe for ggplot2 visualization =====
evap_df_expanded <- as.data.frame(evap_raster_expanded, xy = TRUE)
colnames(evap_df_expanded)[3] <- "evap"  # Rename column for ggplot fill aesthetic

# ===== 9. Create evaporation loss map with pipeline overlay =====
ggplot() +
  geom_raster(data = evap_df_expanded, aes(x = x, y = y, fill = evap)) +
  scale_fill_viridis(name = "Evaporation (mm/year)", option = "inferno") +
  geom_sf(data = pipeline_sf, color = "black", size = 3) +
  geom_sf(data = pipeline_route_sf, color = "blue", size = 1.2) +
  annotation_scale(location = "bl", width_hint = 0.3) +
  annotation_north_arrow(location = "tl", which_north = "true",
                         style = north_arrow_fancy_orienteering) +
  labs(title = "Pipeline Route with Evaporation Loss",
       x = "Longitude", y = "Latitude") +
  theme_minimal()

# ====== Save evaporation map to PNG with 10:8 ratio ======
ggsave("pipeline_evaporation_map_.png",
       width = 10, height = 8, dpi = 400)
```

R code of Figure 5:

```
# ====== Install and Load Required Package ======
if (!require(ggplot2)) install.packages("ggplot2")
library(ggplot2)

# ====== Create the Base Risk Matrix ======
risk_matrix <- expand.grid(
  Likelihood = factor(c("Very Unlikely", "Unlikely", "Possible", "Likely", "Very Likely"),
                      levels = c("Very Unlikely", "Unlikely", "Possible", "Likely", "Very Likely")),
  Impact = factor(c("Negligible", "Minor", "Moderate", "Significant", "Severe"),
                  levels = c("Negligible", "Minor", "Moderate", "Significant", "Severe"))
)

# ====== Calculate Risk Scores ======
risk_matrix$Score <- as.numeric(risk_matrix$Likelihood) * as.numeric(risk_matrix$Impact)

# ====== Define Risk Levels ======
risk_matrix$RiskLevel <- cut(risk_matrix$Score,
                             breaks = c(0, 3, 6, 14, 25),
                             labels = c("Low", "Moderate", "High", "Extreme"))

# ====== Color Scheme for Risk Levels ======
risk_colors <- c("Low" = "#A8D08D",       # green
                 "Moderate" = "#FFD966",  # yellow
                 "High" = "#F4B084",      # orange
```

```
                "Extreme" = "#C00000")   # red

  # ====== Define Risk Events and Positions ======
  events <- data.frame(
    Event = c("Corrosion", "Scaling", "Biofouling", "External"),
    Likelihood = factor(c("Likely", "Possible", "Very Likely", "Unlikely"),
                        levels = c("Very Unlikely", "Unlikely", "Possible", "Likely", "Very Likely")),
    Impact = factor(c("Moderate", "Significant", "Minor", "Severe"),
                    levels = c("Negligible", "Minor", "Moderate", "Significant", "Severe"))
  )

  # ====== Plot the Risk Matrix (larger fonts) ======
  risk_plot <- ggplot(risk_matrix, aes(x = Impact, y = Likelihood, fill = RiskLevel)) +
    geom_tile(color = "white") +
    scale_fill_manual(values = risk_colors, name = "Risk Level") +
    geom_text(aes(label = Score), color = "black", size = 8) +  # score size further increased
    geom_text(data = events, aes(x = Impact, y = Likelihood, label = Event),
              color = "black", fontface = "bold", size = 8, vjust = -1.2, inherit.aes = FALSE) +
    theme_minimal(base_size = 18) +  # further increase base font size
    labs(title = "Risk Matrix",
         x = "Impact",
         y = "Likelihood") +
    theme(
      axis.text.x = element_text(angle = 30, hjust = 1, size = 18),
      axis.text.y = element_text(size = 18),
      axis.title = element_text(size = 20, face = "bold"),
      legend.title = element_text(size = 18),
      legend.text = element_text(size = 16),
      plot.title = element_text(hjust = 0.5, size = 24, face = "bold")
    )

  # ====== Display the Plot ======
  print(risk_plot)

  # ====== Save to PNG with 16:9 ratio ======
ggsave("risk_matrix_16x9_largest_font.png", plot = risk_plot, width = 16, height = 9, dpi = 300)
```

R code of Graphical Abstract

```
  # ====== 1.Install and load required R packages (if not already installed) ======
  packages <- c("ggplot2", "ggmap", "sf", "elevatr", "ggspatial", "dplyr", "gridExtra")
  new_packages <- packages[!(packages %in% installed.packages()[,"Package"])]
  if(length(new_packages)) install.packages(new_packages)

  library(ggplot2)
  library(ggmap)
  library(sf)
  library(elevatr)
  library(ggspatial)
  library(dplyr)
  library(gridExtra)
  library(cowplot)

  # ====== 2.Register your Google Maps API key ======
```

```
register_google(key = "AIzaSyB__tXcP0Rp5XwCw8MfRcxDhkVZiJuZQ7U")  # Replace with your own API key

# ====== 3.Define pipeline locations (Congo → Chad → Egypt) ======
locations <- data.frame(
  name = c("Congo River", "Chad", "Egypt"),
  lat = c(-4.267, 15.4542, 26.8206),
  lon = c(15.283, 18.7322, 30.8025)
)

# ====== 4.Convert location points to sf format ======
locations_sf <- st_as_sf(locations, coords = c("lon", "lat"), crs = 4326)

# ====== 5.Retrieve elevation data for each point ======
elev_data <- get_elev_point(locations_sf, prj = st_crs(4326), src = "aws")
locations_sf$elevation <- elev_data$elevation
locations$elevation <- elev_data$elevation  # Also add to original dataframe

# ====== 6.Create a line connecting the pipeline points ======
pipeline_line <- st_sfc(st_linestring(as.matrix(locations[, c("lon", "lat")])), crs = 4326)
pipeline_sf <- st_sf(geometry = pipeline_line)

# ======7. Download base map using ggmap ======
bbox_africa <- c(left = -20, bottom = -35, right = 55, top = 40)
map_base_africa <- get_googlemap(center = c(lon = 17.5, lat = 2.5), zoom = 3, maptype = "terrain")

# ====== 8.Create main pipeline map (region_map) ======
region_map <- ggmap(map_base_africa) +
  geom_sf(data = locations_sf, aes(color = elevation), size = 5, inherit.aes = FALSE) +
  geom_sf(data = pipeline_sf, color = "red", linewidth = 1.5, inherit.aes = FALSE) +
  scale_color_viridis_c(option = "magma", name = "Elevation (m)", guide = guide_colorbar(position = "right")) +
  coord_sf(xlim = c(-20, 55), ylim = c(-35, 40), expand = FALSE) +
  labs(title = "Detailed Pipeline Route", x = "Longitude", y = "Latitude") +
  annotation_scale(location = "bl", width_hint = 0.3) +
  annotation_north_arrow(location = "br", which_north = "true", height = unit(1, "cm")) +
  theme_minimal() +
  theme(
    legend.position = c(0.85, 0.2),
    panel.border = element_rect(color = "black", linewidth = 1, fill = NA)
  )

# ====== 9.Create an inset world map with elevation points ======
world_map <- ggplot() +
  borders("world", colour = "gray50", fill = "white", alpha = 0.5) +
  geom_point(data = locations, aes(x = lon, y = lat, color = elevation), size = 3) +
  scale_color_viridis_c(option = "magma", name = "Elevation (m)", guide = "none") +
  theme_void() +
  annotation_scale(location = "bl", width_hint = 0.3) +
  annotation_north_arrow(location = "br", which_north = "true", height = unit(1, "cm")) +
  theme(
    legend.position = "none",
    panel.background = element_rect(fill = alpha("white", 0.5), color = "black"),
    plot.background = element_rect(fill = alpha("white", 0.5), color = "black")
  )
```

```
# ====== 10.Combine main map and inset map (final_plot_1) ======
final_plot <- ggdraw() +
  draw_plot(region_map, x = 0, y = 0, width = 1, height = 1) +
  draw_plot(world_map, x = 0.15, y = 0.2, width = 0.25, height = 0.25)

print(final_plot)
final_plot_1 <- final_plot  # Save as plot A

# ======11. Load additional packages for second plot ======
required_packages <- c("sf", "ggplot2", "cowplot", "maps")
for (pkg in required_packages) {
  if (!require(pkg, character.only = TRUE)) {
    install.packages(pkg)
    library(pkg, character.only = TRUE)
  }
}
library(sf)
library(ggplot2)
library(cowplot)
library(maps)

# ======12. Define same pipeline locations for consistency ======
locations <- data.frame(
  name = c("Congo River", "Chad", "Egypt"),
  lon  = c(15.283, 18.7322, 30.8025),
  lat  = c(-4.267, 15.4542, 26.8206)
)

locations_sf <- st_as_sf(locations, coords = c("lon", "lat"), crs = 4326)
pipeline_line <- st_sfc(st_linestring(as.matrix(locations[, c("lon", "lat")])), crs = 4326)

# ====== 13.Generate synthetic elevation profile ======
L_total <- 4094615
dist_seq <- seq(0, L_total, length.out = 1000)
control_dist <- c(0, L_total/2, L_total)
control_elev <- c(280, 330, 290)
elev_profile <- approx(x = control_dist, y = control_elev, xout = dist_seq)$y
elev_df <- data.frame(Distance_km = dist_seq / 1000, Elevation_m = elev_profile)

# ====== 14.Define key elevation and pump station points ======
locations_elev <- data.frame(
  name = c("Congo River", "Chad", "Egypt"),
  Distance_km = c(0, (L_total/2)/1000, L_total/1000),
  Elevation_m = c(280, 330, 290)
)

pump_station_fraction <- c(1/5, 2/5, 3/5, 4/5)
pump_station_dist <- pump_station_fraction * L_total
pump_station_elev <- approx(x = dist_seq, y = elev_profile, xout = pump_station_dist)$y
pump_elev_df <- data.frame(
  Distance_km = pump_station_dist / 1000,
  Elevation_m = pump_station_elev,
```

```
  station     = paste("Pump", 1:length(pump_station_elev))
)

# ====== 15.Create elevation profile plot (p_elev) ======
p_elev <- ggplot(elev_df, aes(x = Distance_km, y = Elevation_m)) +
  geom_line(color = "blue") +
  geom_point(data = locations_elev, aes(x = Distance_km, y = Elevation_m), color = "black", size = 3) +
  geom_point(data = pump_elev_df, aes(x = Distance_km, y = Elevation_m), color = "red", size = 3) +
  labs(title = "Pipeline Elevation Profile", x = "Distance (km)", y = "Elevation (m)") +
  theme_minimal() +
  theme(
    plot.background  = element_rect(fill = alpha("white", 0.5), color = NA),
    panel.background = element_rect(fill = alpha("white", 0.5), color = NA)
  )

# ====== 16.Prepare geographic pipeline map (p_map) ======
world_map <- map_data("world")
pipeline_line_proj <- st_transform(pipeline_line, 3857)
pipeline_points_proj <- st_line_sample(pipeline_line_proj, sample = seq(0, 1, length.out = 1000))
pipeline_points <- st_transform(pipeline_points_proj, 4326)
pipeline_coords <- st_coordinates(pipeline_points)
pipeline_df <- data.frame(pipeline_coords)
names(pipeline_df) <- c("lon", "lat")

pump_stations_proj <- st_line_sample(pipeline_line_proj, sample = pump_station_fraction)
pump_stations <- st_transform(pump_stations_proj, 4326)
pump_stations_coords <- st_coordinates(pump_stations)
pump_stations_df <- data.frame(
  lon = pump_stations_coords[,1],
  lat = pump_stations_coords[,2],
  station = paste("Pump", 1:nrow(pump_stations_coords))
)

xlim_vals <- c(10, 32)
ylim_vals <- c(-5, 30)

p_map <- ggplot() +
  geom_polygon(data = world_map, aes(x = long, y = lat, group = group), fill = "gray95", color = "gray70") +
  geom_path(data = pipeline_df, aes(x = lon, y = lat), color = "blue", size = 1) +
  geom_point(data = pump_stations_df, aes(x = lon, y = lat), color = "red", size = 3) +
  geom_text(data = pump_stations_df, aes(x = lon, y = lat, label = station), color = "red", vjust = -1, size = 3) +
  geom_point(data = locations, aes(x = lon, y = lat), color = "black", size = 3) +
  geom_text(data = locations, aes(x = lon, y = lat, label = name), color = "black", vjust = 1.5, size = 3) +
  coord_fixed(ratio = 1, xlim = xlim_vals, ylim = ylim_vals) +
  labs(title = "Geographical Map of Pipeline and Pump Stations", x = "Longitude", y = "Latitude") +
  theme_minimal()

# ====== 17.Combine map and elevation into final_plot_2 ======
final_plot <- ggdraw() +
  draw_plot(p_map, x = 0, y = 0, width = 1, height = 1) +
  draw_plot(p_elev, x = 0.2, y = 0.65, width = 0.25, height = 0.25)

print(final_plot)
```

```
final_plot_2 <- final_plot  # Save as plot B

# ====== 18.Combine plot A and B into final horizontal 16:9 layout ======
if (!require("cowplot")) install.packages("cowplot")
library(cowplot)

map_with_inset <- final_plot_1  # Plot A
map_with_elevation <- final_plot_2  # Plot B

final_combined_plot <- plot_grid(
  map_with_inset, map_with_elevation,
  labels = c("A", "B"),
  ncol = 2,
  rel_widths = c(1, 1)
)

# ====== 19.Display combined plot ======
print(final_combined_plot)

# ======20 Save to PNG with 16:9 ratio ======
ggsave("combined_pipeline_plot_16x9.png", plot = final_combined_plot, width = 16, height = 9, dpi = 300)
```