Optimization Semester Report

MAT2407

Tran Dang Quang Minh, K68A2 Mathematics and Computer Science

Vietnam National University
Hanoi University of Science
Faculty of Mathematics, Mechanics and Informatics

Hanoi, December 2025

Contents

Report Topic

Report Topic

Introduction

Report Topic

Introduction

Optimal Path-finding Model under time-dependent rainfall and local flooding conditions considering the continuous movement of convective clouds.

Report Topic

Introduction

Optimal Path-finding Model under time-dependent rainfall and local flooding conditions considering the continuous movement of convective clouds.

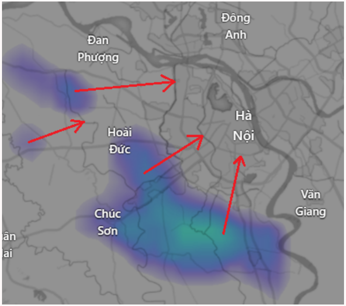


Figure 1.1: *Convective clouds illustrated on windy.com*

Report Topic

Inspiration

Report Topic

Inspiration

Heavy flooding in Hanoi on October 1st, 2025.

Report Topic

Inspiration

Heavy flooding in Hanoi on October 1st, 2025.

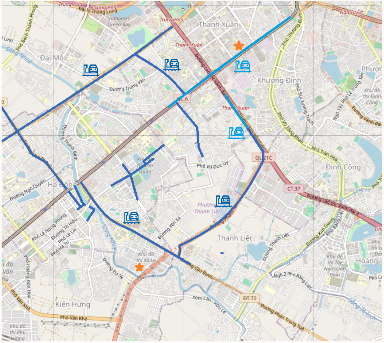


Figure 1.2: *Flooded (light blue) and Heavily flooded (blue) streets on October 1st, 2025*

Report Topic

Inspiration

Heavy flooding in Hanoi on October 1st, 2025.

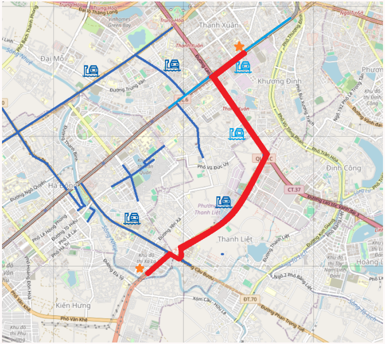


Figure 1.3: *Normal optimal route from HUS to Xala Urban Area (red line)*

Report Topic

Inspiration

Heavy flooding in Hanoi on October 1st, 2025.

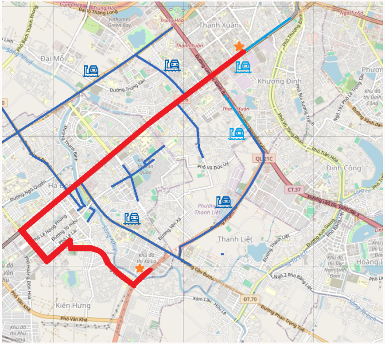


Figure 1.4: *Optimal route from HUS to Xala Urban Area in flooding conditions (red line)*

Report Topic

Simplifying Given Problem Statements

Report Topic

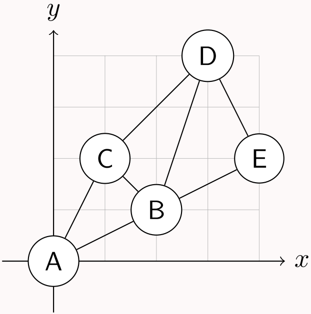
Simplifying Given Problem Statements

Convert the map to a Flat Undirected Graph with vertices set V and edges set E.

Report Topic

Simplifying Given Problem Statements

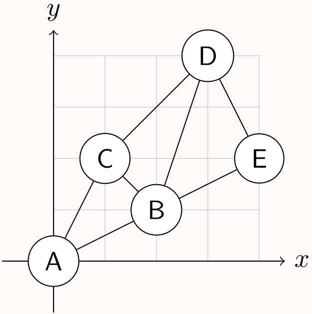
Convert the map to a Flat Undirected Graph with vertices set V and edges set E.



Report Topic

Simplifying Given Problem Statements

Convert the map to a Flat Undirected Graph with vertices set V and edges set E.



Each vertex i is located at (x_i, y_i).

Each edge e_{ij}, which connects vertices $i(x_i, y_i)$ and $j(x_j, y_j)$, has length:

$$l_{ij} = l_{ji} = 1000 \cdot \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

Report Topic

Simplifying Given Problem Statements

Convert the map to a Flat Undirected Graph with vertices set V and edges set E.
- Each vertex i is located at (x_i, y_i).
- Each edge e_{ij}, which connects vertices $i(x_i, y_i)$ and $j(x_j, y_j)$, has properties:
  + $l_{ij} = l_{ji} = 1000 \cdot \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$: Length of the street (m),

Report Topic

Simplifying Given Problem Statements

Convert the map to a Flat Undirected Graph with vertices set V and edges set E.

- Each vertex i is located at (x_i, y_i).
- Each edge e_{ij}, which connects vertices $i(x_i, y_i)$ and $j(x_j, y_j)$, has properties:
  - + $l_{ij} = l_{ji} = 1000 \cdot \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$: Length of the street (m),
  - + $v_{ij} = v_{ji}$: Average speed of vehicles on the street (m/s),

Report Topic

Simplifying Given Problem Statements

Convert the map to a Flat Undirected Graph with vertices set V and edges set E.

- Each vertex i is located at (x_i, y_i).
- Each edge e_{ij}, which connects vertices $i(x_i, y_i)$ and $j(x_j, y_j)$, has properties:
  + $l_{ij} = l_{ji} = 1000 \cdot \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$: Length of the street (m),
  + $v_{ij} = v_{ji}$: Average speed of vehicles on the street (m/s),
  + $d_{ij} = d_{ji}$: Total drainage capacity of the street (m^3/h),

Report Topic

Simplifying Given Problem Statements

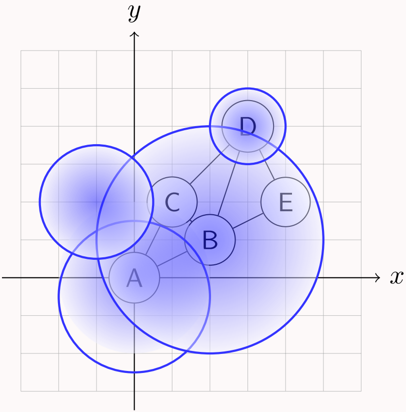
Convert the map to a Flat Undirected Graph with vertices set V and edges set E.

- Each vertex i is located at (x_i, y_i).
- Each edge e_{ij}, which connects vertices $i(x_i, y_i)$ and $j(x_j, y_j)$, has properties:
  - $+ \; l_{ij} = l_{ji} = 1000 \cdot \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$: Length of the street (m),
  - $+ \; v_{ij} = v_{ji}$: Average speed of vehicles on the street (m/s),
  - $+ \; d_{ij} = d_{ji}$: Total drainage capacity of the street (m^3/h),
  - $+ \; w_{ij} = w_{ji}$: Width of the street (m),

Report Topic

Simplifying Given Problem Statements

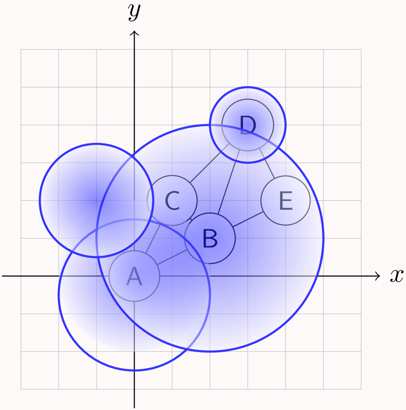
Considering the i-th convective cloud as a circle $B_{r_i}(x_i, y_i)$.



Report Topic

Simplifying Given Problem Statements

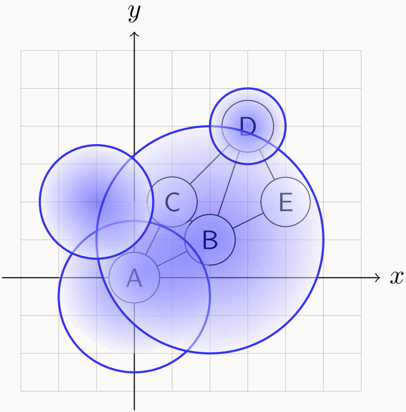
The i-th convective cloud has a rainfall rate of $R_i(mm/h)$ at every point.



Report Topic

Simplifying Given Problem Statements

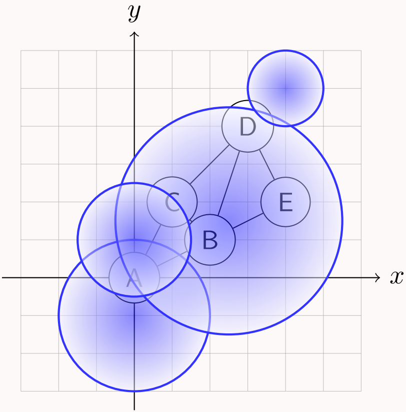
The i-th cloud continuously moving towards vector $v_i = (\Delta_x, \Delta_y)$ every second.



Report Topic

Simplifying Given Problem Statements

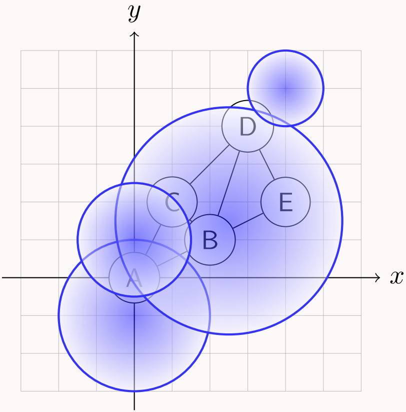
The i-th cloud continuously moving towards vector $v_i = (\Delta_x, \Delta_y)$ every second.



Report Topic

Simplifying Given Problem Statements

Let $K_{ij}(t_0)$ be the set of clouds that cross the edge e_{ij} at time t_0.



Report Topic

Simplifying Given Problem Statements

Consider the rainfall rate on edge e_{ij} at time t_0 defined as

$$r_{ij}(t_0) = \begin{cases} \max\limits_{k \in K_{ij}(t_0)} R_k, & \text{if } K_{ij}(t_0) \neq \varnothing, \\ 0, & \text{if } K_{ij}(t_0) = \varnothing. \end{cases}$$

Report Topic

Simplifying Given Problem Statements

Consider the rainfall rate on edge e_{ij} at time t_0 defined as

$$r_{ij}(t_0) = \begin{cases} \max\limits_{k \in K_{ij}(t_0)} R_k, & \text{if } K_{ij}(t_0) \neq \varnothing, \\ 0, & \text{if } K_{ij}(t_0) = \varnothing. \end{cases}$$

Now we have the full problem's input...

Report Topic

Simplifying Given Problem Statements

Convert the map to a Flat Undirected Graph with vertices set V and edges set E.
- Each vertex i is located at (x_i, y_i).
- Each edge e_{ij}, which connects vertices $i(x_i, y_i)$ and $j(x_j, y_j)$, has properties:
  + $l_{ij} = l_{ji} = 1000 \cdot \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$: Length of the street (m),
  + $v_{ij} = v_{ji}$: Average speed of vehicles on the street (m/s),
  + $d_{ij} = d_{ji}$: Total drainage capacity of the street (m^3/h),
  + $w_{ij} = w_{ji}$: Width of the street (m),
  + $r_{ij}(t_0) = r_{ji}(t_0)$: Rainfall rate on the street at time t_0 (mm/h),

Report Topic

Simplifying Given Problem Statements

Convert the map to a Flat Undirected Graph with vertices set V and edges set E.

- Each vertex i is located at (x_i, y_i).
- Each edge e_{ij}, which connects vertices $i(x_i, y_i)$ and $j(x_j, y_j)$, has properties:
  + $l_{ij} = l_{ji} = 1000 \cdot \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$: Length of the street (m),
  + $v_{ij} = v_{ji}$: Average speed of vehicles on the street (m/s),
  + $d_{ij} = d_{ji}$: Total drainage capacity of the street (m^3/h),
  + $w_{ij} = w_{ji}$: Width of the street (m),
  + $r_{ij}(t_0) = r_{ji}(t_0)$: Rainfall rate on the street at time t_0 (mm/h),
  + $K_{ij}(t_0) = K_{ji}(t_0)$: Set of convective clouds cross the street at time t_0.

Report Topic

Simplifying Given Problem Statements

Convert the map to a Flat Undirected Graph with vertices set V and edges set E.

- Each vertex i is located at (x_i, y_i).
- Each edge e_{ij}, which connects vertices $i(x_i, y_i)$ and $j(x_j, y_j)$, has properties:
  + $l_{ij} = l_{ji} = 1000 \cdot \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$: Length of the street (m),
  + $v_{ij} = v_{ji}$: Average speed of vehicles on the street (m/s),
  + $d_{ij} = d_{ji}$: Total drainage capacity of the street (m^3/h),
  + $w_{ij} = w_{ji}$: Width of the street (m),
  + $r_{ij}(t_0) = r_{ji}(t_0)$: Rainfall rate on the street at time t_0 (mm/h),
  + $K_{ij}(t_0) = K_{ji}(t_0)$: Set of convective clouds cross the street at time t_0.
- Each convective cloud $B_{r_i}(x_i, y_i)$ has properties:

Report Topic

Simplifying Given Problem Statements

Convert the map to a Flat Undirected Graph with vertices set V and edges set E.

- Each vertex i is located at (x_i, y_i).
- Each edge e_{ij}, which connects vertices $i(x_i, y_i)$ and $j(x_j, y_j)$, has properties:
  - + $l_{ij} = l_{ji} = 1000 \cdot \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$: Length of the street (m),
  - + $v_{ij} = v_{ji}$: Average speed of vehicles on the street (m/s),
  - + $d_{ij} = d_{ji}$: Total drainage capacity of the street (m^3/h),
  - + $w_{ij} = w_{ji}$: Width of the street (m),
  - + $r_{ij}(t_0) = r_{ji}(t_0)$: Rainfall rate on the street at time t_0 (mm/h),
  - + $K_{ij}(t_0) = K_{ji}(t_0)$: Set of convective clouds cross the street at time t_0.
- Each convective cloud $B_{r_i}(x_i, y_i)$ has properties:
  - + (x_i, y_i): The center of the cloud,

Report Topic

Simplifying Given Problem Statements

Convert the map to a Flat Undirected Graph with vertices set V and edges set E.

- Each vertex i is located at (x_i, y_i).
- Each edge e_{ij}, which connects vertices $i(x_i, y_i)$ and $j(x_j, y_j)$, has properties:
  + $l_{ij} = l_{ji} = 1000 \cdot \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$: Length of the street (m),
  + $v_{ij} = v_{ji}$: Average speed of vehicles on the street (m/s),
  + $d_{ij} = d_{ji}$: Total drainage capacity of the street (m^3/h),
  + $w_{ij} = w_{ji}$: Width of the street (m),
  + $r_{ij}(t_0) = r_{ji}(t_0)$: Rainfall rate on the street at time t_0 (mm/h),
  + $K_{ij}(t_0) = K_{ji}(t_0)$: Set of convective clouds cross the street at time t_0.
- Each convective cloud $B_{r_i}(x_i, y_i)$ has properties:
  + (x_i, y_i): The center of the cloud.
  + $(\Delta_{x_i}, \Delta_{y_i})$: The movement vector of the cloud every second (in meters).

Report Topic

Simplifying Given Problem Statements

Convert the map to a Flat Undirected Graph with vertices set V and edges set E.

- Each vertex i is located at (x_i, y_i).
- Each edge e_{ij}, which connects vertices $i(x_i, y_i)$ and $j(x_j, y_j)$, has properties:
  + $l_{ij} = l_{ji} = 1000 \cdot \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$: Length of the street (m),
  + $v_{ij} = v_{ji}$: Average speed of vehicles on the street (m/s),
  + $d_{ij} = d_{ji}$: Total drainage capacity of the street (m^3/h),
  + $w_{ij} = w_{ji}$: Width of the street (m),
  + $r_{ij}(t_0) = r_{ji}(t_0)$: Rainfall rate on the street at time t_0 (mm/h),
  + $K_{ij}(t_0) = K_{ji}(t_0)$: Set of convective clouds cross the street at time t_0.
- Each convective cloud $B_{r_i}(x_i, y_i)$ has properties:
  + (x_i, y_i): The center of the cloud,
  + $(\Delta_{x_i}, \Delta_{y_i})$: The movement vector of the cloud every second (in meters),
  + r_i: The radius of the cloud (m),

Report Topic

Simplifying Given Problem Statements

Convert the map to a Flat Undirected Graph with vertices set V and edges set E.
- Each vertex i is located at (x_i, y_i).
- Each edge e_{ij}, which connects vertices $i(x_i, y_i)$ and $j(x_j, y_j)$, has properties:
  + $l_{ij} = l_{ji} = 1000 \cdot \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$: Length of the street (m),
  + $v_{ij} = v_{ji}$: Average speed of vehicles on the street (m/s),
  + $d_{ij} = d_{ji}$: Total drainage capacity of the street (m^3/h),
  + $w_{ij} = w_{ji}$: Width of the street (m),
  + $r_{ij}(t_0) = r_{ji}(t_0)$: Rainfall rate on the street at time t_0 (mm/h),
  + $K_{ij}(t_0) = K_{ji}(t_0)$: Set of convective clouds cross the street at time t_0.
- Each convective cloud $B_{r_i}(x_i, y_i)$ has properties:
  + (x_i, y_i): The center of the cloud,
  + $(\Delta_{x_i}, \Delta_{y_i})$: The movement vector of the cloud every second (in meters),
  + r_i: The radius of the cloud (m),
  + R_i: The overall rainfall rate of the cloud (mm/h).

Report Topic

Simplifying Given Problem Statements

- The problem can now be stated as:

Find the optimal (in fuel usage) route from a vertex (x_i, y_i) *to another vertex* (x_j, y_j) *avoiding any streets that flooded higher than* $H_{max}(cm)$ *such that the overall travel time does not exceed* $T_{max}(h)$*. Suppose that the cost of fuel per kilometer is* $C = 0.75\ \$/km$*.*

Preprocessing Inputs

Preprocessing Inputs

Data Processing

Preprocessing Inputs

Data Processing

- The time cost of the street e_{ij} is:

$$t_{ij} = \frac{1}{3600} \cdot \frac{l_{ij}}{v_{ij}} \ (h)$$

-

Preprocessing Inputs

Data Processing

- The time cost of the street e_{ij} is:

$$t_{ij} = \frac{1}{3600} \cdot \frac{l_{ij}}{v_{ij}} \ (h)$$

- The area of the street e_{ij} is:

$$A_{ij} = l_{ij} \cdot w_{ij} \ (m^2)$$

Preprocessing Inputs

Data Processing

- The time cost of the street e_{ij} is:

$$t_{ij} = \frac{1}{3600} \cdot \frac{l_{ij}}{v_{ij}} \ (h)$$

- The area of the street e_{ij} is:

$$A_{ij} = l_{ij} \cdot w_{ij} \ (m^2)$$

- Therefore, the flood water volume per hour of the street e_{ij} at time t_0 is:

$$V_{ij}(t_0) = 0.001 \cdot r_{ij}(t_0) \cdot A_{ij} \ (m^3/h)$$

Preprocessing Inputs

Data Processing

- The time cost of the street e_{ij} is:

$$t_{ij} = \frac{1}{3600} \cdot \frac{l_{ij}}{v_{ij}} \ (h)$$

- The area of the street e_{ij} is:

$$A_{ij} = l_{ij} \cdot w_{ij} \ (m^2)$$

- Therefore, the flood water volume per hour of the street e_{ij} at time t_0 is:

$$V_{ij}(t_0) = 0.001 \cdot r_{ij}(t_0) \cdot A_{ij} \ (m^3/h)$$

- Then, the flood height of the street e_{ij} at time t_1 after $\Delta t = t_1 - t_0(h)$ is:

$$H_{ij}(t_1) = \begin{cases} 0, & t_1 = 0, \\ H_{ij}(t_0) + \Delta t \cdot \frac{V_{ij}(t_0) - d_{ij}}{A_{ij}}, & V_{ij}(t_0) > d_{ij}, \ \ (m) \\ \max(0, H_{ij}(t_0) - \Delta t \cdot \frac{d_{ij} - V_{ij}(t_0)}{A_{ij}}), & V_{ij}(t_0) \le d_{ij}. \end{cases}$$

Modeling Process

Modeling Process

Setting up Variables

Modeling Process

Setting up Variables

For each street e_{ij}, denotes a binary variables x_{ijk}:

$$x_{ijk} = \begin{cases} 1, & \text{the street } e_{ij} \text{ is in the optimal route and we go through it at } t = k, \\ 0, & \text{otherwise.} \end{cases}$$

Modeling Process

Setting up Variables

For each street e_{ij}, denotes a binary variables x_{ijk}:

$$x_{ijk} = \begin{cases} 1, & \text{the street } e_{ij} \text{ is in the optimal route and we go through it at } t = k, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{ik} = \begin{cases} 1, & \text{the traveler is at the } i\text{-th node at } t = k, \\ 0, & \text{otherwise.} \end{cases}$$

Denotes a discrete time variable $k \in T$, T is a dimension of time from 0 to T_{max}. Consider E_{ans} the set of streets in the optimal route. If there doesn't exist a valid route, returns an empty set. Otherwise, returns the set of streets from the start point to the destination point.

Modeling Process

Objective Function

Modeling Process

Objective Function

The objective is to determine the lowest fuel cost, so the Objective Function is:

$$\mathit{min} \quad \sum_{(i,j)\in E, k\in T} 0.001 \cdot x_{ijk} \cdot l_{ij} \cdot C$$

Modeling Process

Objective Function

The objective is to determine the lowest fuel cost, so the Objective Function is:

$$\textit{min} \quad \sum_{(i,j)\in E, k\in T} 0.001 \cdot x_{ijk} \cdot l_{ij} \cdot C$$

or to be simpler, $\textit{min} \quad \sum_{(i,j)\in E, k\in T} x_{ijk} \cdot l_{ij}$

Modeling Process

Constraints

Modeling Process

Constraints

1. Path Constraints:

Modeling Process

Constraints

1. Path Constraints:

The sum of the flows entering a node must equal the sum of the flows exiting that node, except for the source and destination.

Modeling Process

Constraints

1. Path Constraints:

The sum of the flows entering a node must equal the sum of the flows exiting that node, except for the source and destination.

- The source node s only has a single path out:

$$\sum_{(s,i)\in E, k\in T} x_{sik} = 1.$$

Modeling Process

Constraints

1. Path Constraints:

The sum of the flows entering a node must equal the sum of the flows exiting that node, except for the source and destination.

- The source node s only has a single path out:

$$\sum_{(s,i)\in E,k\in T} x_{sik} = 1.$$

- The destination node t only has a single path in:

$$\sum_{(j,t)\in E,k\in T} x_{jtk} = 1.$$

Modeling Process

Constraints

1. Path Constraints:

The sum of the flows entering a node must equal the sum of the flows exiting that node, except for the source and destination.

- The source node s only has a single path out:

$$\sum_{(s,i)\in E, k\in T} x_{sik} = 1.$$

- The destination node t only has a single path in:

$$\sum_{(j,t)\in E, k\in T} x_{jtk} = 1.$$

- For every node $v \neq s, t$, the flow constraint is:

$$\sum_{(i,v)\in E, k\in T} x_{ivk} - \sum_{(v,j)\in E, k\in T} x_{vjk} = 0.$$

Modeling Process

Constraints

1. Path Constraints:

The sum of the flows entering a node must equal the sum of the flows exiting that node, except for the source and destination.

- The source node s only has a single path out:

$$\sum_{(s,i)\in E,k\in T} x_{sik} = 1.$$

- The destination node t only has a single path in:

$$\sum_{(j,t)\in E,k\in T} x_{jtk} = 1.$$

- For every node $v \neq s, t$, the flow constraint is:

$$\sum_{(i,v)\in E,k\in T} x_{ivk} - \sum_{(v,j)\in E,k\in T} x_{vjk} = 0.$$

This confirms that there's only a single path from the source to the destination.

Modeling Process

Constraints

2. Time Constraints:

Modeling Process

Constraints

2. Time Constraints:

The total traveling time does not exceed $T_{max}(h)$.

Modeling Process

Constraints

2. Time Constraints:

The total traveling time does not exceed $T_{max}(h)$.

$$\sum_{(i,j)\in E,k\in T} t_{ij} \cdot x_{ijk} \leq T_{max}$$

Modeling Process

Constraints

2. Time Constraints:

The total traveling time does not exceed $T_{max}(h)$.

$$\sum_{(i,j)\in E, k\in T} t_{ij} \cdot x_{ijk} \leq T_{max}.$$

3. Flood Constraints:

Modeling Process

Constraints

2. Time Constraints:

The total traveling time does not exceed $T_{max}(h)$.

$$\sum_{(i,j)\in E, k\in T} t_{ij} \cdot x_{ijk} \leq T_{max}$$

3. Flood Constraints:

$$H_{ij}(t_{si}) \leq H_{max} \forall (i,j) \in E,$$

t_{si}: time to travel from source node to the i-th node.

Modeling Process

Constraints

2. Time Constraints:

The total traveling time does not exceed $T_{max}(h)$.

$$\sum_{(i,j)\in E, k\in T} t_{ij} \cdot x_{ijk} \leq T_{max}$$

.

3. Flood Constraints:

$$H_{ij}(t_{si}) \leq H_{max} \forall (i,j) \in E,$$

t_{si}: time to travel from source node to the i-th node.

Since there's a t_{si} variable in the constraint and the traveling time between nodes are different, it would be a nonlinear constraint.

Modeling Process

Constraints

2. Time Constraints:

The total traveling time does not exceed $T_{max}(h)$.

$$\sum_{(i,j)\in E,k\in T} t_{ij} \cdot x_{ijk} \leq T_{max}$$

.

3. Flood Constraints:

$$H_{ij}(t_{si}) \leq H_{max} \forall (i,j) \in E,$$

t_{si}: time to travel from source node to the i-th node.

Since there's a t_{si} variable in the constraint and the traveling time between nodes are different, it would be a nonlinear constraint.

Hence, we have to linearize the time by dt, such that dt is the smallest duration (in second) we store a state. For example, if $dt = 5$, we store the state of conditions every 5 seconds.

Modeling Process

Constraints

2. Time Constraints:

The total traveling time does not exceed $T_{max}(h)$.

$$\sum_{(i,j)\in E,k\in T} t_{ij} \cdot x_{ijk} \leq T_{max}$$

.

3. Flood Constraints:

$$H_{ij}(t_{si}) \leq H_{max} \forall (i,j) \in E,$$

t_{si}: time to travel from source node to the i-th node.

Since there's a t_{si} variable in the constraint and the traveling time between nodes are different, it would be a nonlinear constraint.

Hence, we have to linearize the time by dt, such that dt is the smallest duration (in second) we store a state. For example, if $dt = 5$, we store the state of conditions every 5 seconds.

Then, if the t_{si} isn't divisible by dt, we choose the maximum t that can be divided by dt.

Modeling Process

Constraints

4. Other Constraints:

Modeling Process

Constraints

4. Other Constraints:

$$y_s^0 = 1, \tag{1}$$

$$y_i^0 = 0, \quad \forall i \in V \setminus \{s\}. \tag{2}$$

$$x_{ijk} \le y_i^k, \quad \forall (i,j) \in E,\ \forall k \in T, \tag{3}$$

$$y_j^{k+t_{ij}} \ge x_{ijk}, \quad \forall (i,j) \in E,\ \forall k \text{ such that } k + t_{ij} \le K. \tag{4}$$

$$\sum_{i \in V} y_i^k \le 1, \quad \forall k = 0, 1, \ldots, K. \tag{5}$$

$$\sum_{k=0}^{K} y_i^k \le 1, \quad \forall i \in V. K = \left\lfloor \frac{T_{\max} \cdot 3600}{dt} \right\rfloor. \tag{6}$$

Modeling and Displaying

Modeling and Displaying

Modeling and Displaying

We can now use gurobipy to run the model on python, and matplotlib to display the map.

Modeling and Displaying

Modeling and Displaying

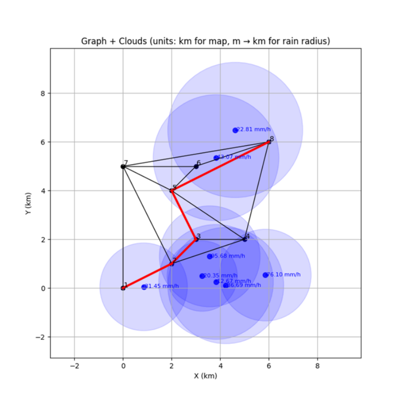


Figure 4.1: *An example of a way to go from the 1st Node to the 8th Node*

Modeling and Displaying

Modeling and Displaying

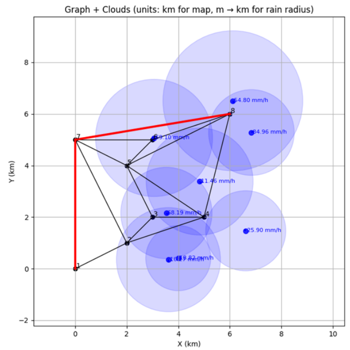


Figure 4.2: *Another way to go from the 1st Node to the 8th Node in a different scenario*

Modeling and Displaying

Modeling and Displaying

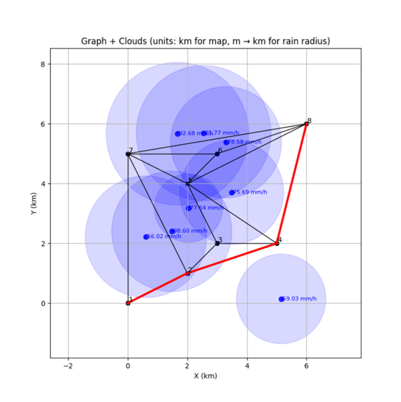


Figure 4.3: *Another way to go from the 1st Node to the 8th Node in a different scenario*

Conclusion

Conclusion

Conclusion

- This work formulates the route planning problem under rainfall-induced flooding as a **time-expanded Mixed Integer Linear Programming (MILP)** model.
- The proposed model explicitly incorporates **dynamic rainfall, cloud movement, road drainage capacity, and flood accumulation over time**.
- By discretizing time, the originally nonlinear flood constraint is successfully linearized, allowing the problem to be solved using standard MILP solvers such as **Gurobi**.
- The obtained solution guarantees a **single continuous path** from the source to the destination while satisfying both **travel time** and **flood safety constraints**.

Conclusion

Comparison with Other Approaches

- Classical shortest-path algorithms such as **Dijkstra or A*** are efficient for static networks but fail to capture **time-dependent flooding and dynamic constraints**.
- Heuristic or metaheuristic methods (e.g., **Genetic Algorithms, Ant Colony Optimization**) can handle dynamic environments but **do not guarantee optimality**.
- In contrast, the proposed MILP model provides a **globally optimal solution** under well-defined constraints, making it suitable for **risk-sensitive and safety-critical routing**.
- The trade-off lies in computational complexity, as MILP scalability is limited compared to graph-based heuristics.

Conclusion

Efficiency and Future Work

- Numerical experiments show that the MILP-based approach is effective for **small- to medium-scale networks** with fine temporal resolution.
- The model accurately avoids flooded streets by accounting for **rainfall accumulation and drainage dynamics**.
- However, the time-expanded formulation leads to a rapid growth in the number of variables and constraints as the network size or time horizon increases.
- Future work includes **model reduction techniques**, **rolling-horizon optimization**, and hybrid approaches combining **MILP with heuristic routing algorithms**.

Thank you!