



# Project Bantay: Breadth-First Search and Dijkstra Implementation for Community-Accessible Disaster Response

Chris Nicole Piamonte<sup>1</sup>, John Paul M. Curada\*<sup>1</sup>, and Marie Criz Zaragoza<sup>1</sup>

College of Computer and Information Sciences,  
Polytechnic University of the Philippines,  
Manila, Philippines  
chrispiamonte06@gmail.com, johncurada.02@gmail.com,  
mariecriz.pup@gmail.com

**Abstract.** This research presents adapted implementations of Breadth-First Search (BFS) and Dijkstra’s algorithm for real-time flood prediction and evacuation routing. The work shows how government sensor data can be transformed into accessible disaster information for communities. The BANTAY system uses optimized graph algorithms that achieve  $O(V+E)$  complexity for flood spread prediction and  $O((V+E)\log V)$  for evacuation routing, with processing times under one second across Metro Manila’s Pasig-Marikina-Tullahan Basin. The system connects a React frontend with a FastAPI backend and PAGASA sensor APIs that update every 30 seconds. This work demonstrates how classical algorithm optimization enables practical applications that serve communities for proactive disaster response rather than reactive emergency management. The system is publicly accessible at <https://bantay-pmt.org/>.

**Keywords:** Breadth-First Search, Dijkstra algorithm, flood propagation, graph optimization, disaster response

## 1 Introduction

The challenge of transforming complex sensor networks into accessible community information represents an important intersection of algorithm design and practical disaster management. This work addresses how classical graph algorithms—specifically Breadth-First Search (BFS) for modeling how floods spread and Dijkstra’s algorithm for finding safe paths—can be adapted, implemented, and deployed to create flood warning systems that serve community decision-making.

Metro Manila’s 13 million residents face recurring flood crises from the Pasig-Marikina-Tullahan (PMT) River Basin, where the Philippine Atmospheric, Geophysical and Astronomical Services Administration (PAGASA) under the Department of Science and Technology (DOST) operates 17 water level sensors

---

\*Corresponding author: johncurada.02@gmail.com

positioned at key locations. These sensors provide monitoring data every 30 seconds. The challenge lies in processing this sensor data to provide predictive flood information that enables communities to make proactive decisions before disasters strike.

Climate change has intensified water-related disasters globally, with floods causing substantial economic damages during 2024 alone [13]. The Asian Development Bank projects significant GDP impacts from climate-related disasters across Asia [1]. The Philippines experiences particularly severe flooding impacts, with storms like Severe Tropical Storm Kristine affecting 7.1 million people [9]. Kumar and Singh [7] emphasize the critical need for improved flood risk assessment and management strategies globally. Current early warning systems remain reactive rather than predictive, creating gaps between sophisticated government monitoring infrastructure and accessible community information.

The BANTAY (a Filipino word meaning “guard”) system is a practical application of optimized algorithms designed to serve communities. Similar flood warning systems have been deployed internationally, including the Netherlands’ sophisticated flood forecasting infrastructure [3], China’s high-resolution numerical flood models [8], and advanced sensor-based monitoring systems reviewed by Rahman et al. [12]. The research contributes algorithmic analysis and interface development, with the goal of supporting community safety through proactive flood information rather than reactive emergency response.

## 2 Theoretical Framework and Algorithm Design

The research formalizes flood warning as two connected computational problems requiring optimized graph algorithm solutions that address emergency response constraints.

### 2.1 Problem Formalization

**Problem 1 (Spatial Flood Propagation):** Given sensor locations  $S = \{s_1, s_2, \dots, s_n\}$  with water level readings  $W = \{w_1, w_2, \dots, w_n\}$ , spatial grid  $G$  representing geographic area, and flood threshold function  $\theta : S \rightarrow \{0, 1\}$ , determine flood-affected cells  $F \subseteq G$  using optimal graph traversal.

*Real-world implications:* This problem directly addresses the need for spatial prediction of flood extent, enabling communities to identify which neighborhoods, streets, and facilities will be affected as water levels rise. The solution enables proactive evacuation decisions hours before flooding reaches specific locations, potentially saving lives and property by providing advance warning beyond current sensor locations.

**Problem 2 (Evacuation Route Optimization):** Given road network  $R = (V, E)$  with dynamic edge weights  $w : E \rightarrow \mathbb{R}^+$  based on flood conditions, source location  $v_0 \in V$ , and evacuation centers  $C \subseteq V$ , find shortest safe path  $\pi : v_0 \rightsquigarrow c$  where  $c \in C$  minimizes total cost.

*Real-world implications:* During flood emergencies, residents need immediate guidance on safe evacuation routes that avoid flooded roads. This problem addresses the critical challenge of finding not just the shortest path, but the shortest *safe* path that dynamically adapts as flood conditions change, preventing evacuees from encountering impassable or dangerous routes.

## 2.2 Graph Representation Details

The system uses two distinct graph representations optimized for each algorithm:

**Spatial Grid Graph for BFS:** The geographic area is divided into a uniform grid where each cell represents approximately  $500\text{m} \times 500\text{m}$ . Vertices  $V$  correspond to grid cells, and edges  $E$  connect adjacent cells (4-connectivity). Each vertex stores geographic coordinates, elevation data, and current flood status. The graph structure  $G_{\text{spatial}} = (V_{\text{grid}}, E_{\text{adjacent}})$  enables efficient spatial neighbor queries with  $O(1)$  access time.

**Road Network Graph for Dijkstra:** The transportation infrastructure is represented as  $G_{\text{road}} = (V_{\text{intersections}}, E_{\text{roads}})$  where vertices represent road intersections and edges represent road segments. Each edge stores multiple attributes: base distance, current traffic conditions, flood status, and computed traversal cost. The graph is constructed from OpenStreetMap data providing comprehensive road network topology across Metro Manila.

## 2.3 BFS Spatial Flood Propagation Algorithm

The BFS adaptation treats flooded sensor locations as starting nodes, exploring adjacent cells level-by-level to model natural flood spread patterns [6].

---

### Algorithm 1 BFS Flood Propagation

---

**Require:** *flooded\_sensors*, *grid*

**Ensure:** *flood\_map*

```

1: queue  $\leftarrow \emptyset$ , visited  $\leftarrow \emptyset$ 
2: for each sensor  $\in$  flooded_sensors do
3:   queue.add(sensor.location)
4:   visited.add(sensor.location)
5: end for
6: while queue  $\neq \emptyset$  do
7:   current  $\leftarrow$  queue.pop()
8:   neighbors  $\leftarrow$  get_neighbors(current)
9:   for each neighbor  $\in$  neighbors do
10:    if neighbor  $\notin$  visited AND can_flood(neighbor) then
11:      queue.add(neighbor)
12:      visited.add(neighbor)
13:      flood_map.mark(neighbor)
14:    end if
15:  end for
16: end while
17: return flood_map

```

---

**Complexity:** Time  $O(V + E)$ , Space  $O(V)$ . Each cell is processed once, ensuring complete flood area coverage.

The BFS starting nodes include the 17 critical sensor locations from the PAGASA network as detailed in Table 1, providing comprehensive geographic coverage across the PMT Basin for accurate flood spread modeling.

## 2.4 Dijkstra Dynamic Evacuation Routing Algorithm

The system optimizes Dijkstra's algorithm for emergency evacuation by incorporating dynamic edge weights that reflect flood conditions and road accessibility [5].

---

### Algorithm 2 Dijkstra Evacuation Routing

---

**Require:** *road\_network*, *start*, *shelters*, *flood\_data*

**Ensure:** *safe\_path*

```

1:  $dist[v] \leftarrow \infty$  for all  $v$ ,  $dist[start] \leftarrow 0$ 
2:  $queue \leftarrow \{start\}$ 
3: while  $queue \neq \emptyset$  do
4:    $current \leftarrow extract\_min(queue)$ 
5:   if  $current \in shelters$  then
6:     return  $build\_path(current)$ 
7:   end if
8:   for each  $neighbor$  of  $current$  do
9:     if road is safe from flooding then
10:       $cost \leftarrow dist[current] + road\_weight(current, neighbor)$ 
11:      if  $cost < dist[neighbor]$  then
12:         $dist[neighbor] \leftarrow cost$ 
13:         $previous[neighbor] \leftarrow current$ 
14:      end if
15:    end if
16:  end for
17: end while
18: return no safe path found

```

---

**Complexity:** Time  $O((V + E) \log V)$  using min-heap. Dynamic weights adjust for flood conditions, ensuring safe optimal paths.

## 2.5 Dynamic Edge Weight Calculation

The edge weight function  $w(e)$  for road segments incorporates multiple factors reflecting real-time conditions:

$$w(e) = d(e) \times (1 + \alpha \cdot f(e) + \beta \cdot t(e)) \quad (1)$$

where  $d(e)$  is the base distance,  $f(e) \in [0, \infty)$  is the flood risk factor calculated from nearby sensor readings,  $t(e) \in [0, 1]$  is the traffic congestion factor, and  $\alpha, \beta$

are tuning parameters (set to 10 and 0.5 respectively through testing). Roads with critical flood levels are assigned infinite weight, effectively removing them from consideration. The flood risk factor  $f(e)$  increases exponentially as sensor readings approach critical thresholds, computed as:

$$f(e) = \max_{s \in S_{nearby}} \left( \frac{w_s - w_{alert}}{w_{critical} - w_{alert}} \right)^2 \quad (2)$$

where  $S_{nearby}$  represents sensors within 2km of the road segment,  $w_s$  is the current water level, and  $w_{alert}$  and  $w_{critical}$  are threshold levels from Table 1.

Li et al. achieved evacuation route calculations in 0.012 seconds using similar Dijkstra implementations [8], establishing feasibility benchmarks for simultaneous route calculations across multiple communities during emergency situations. Abu Samah et al. [2] demonstrated related modifications for emergency evacuation applications, while Zhu et al. [14] explored 3D Dijkstra variants for urban flood scenarios.

### 3 Sensor Network Infrastructure

The BANTAY system's algorithmic foundation relies on PAGASA's comprehensive water level monitoring infrastructure across the Pasig-Marikina-Tullahan River Basin, providing near real-time data input for both BFS flood propagation and Dijkstra evacuation routing algorithms.

#### 3.1 PAGASA Sensor Network Configuration

The sensor network comprises 17 monitoring stations positioned at key locations, providing continuous water level measurements with precise GPS coordinates at 30-second intervals. These sensors serve as critical input nodes for BFS algorithm initialization and Dijkstra dynamic weight calculations. All sensor data comes from PAGASA's official PMT Flood Forecasting and Warning System [10]. Figure 1 shows the geographic distribution of all monitoring stations across the PMT Basin.

#### 3.2 Sensor Data Integration for Algorithm Processing

Each sensor provides multi-tier alert thresholds (Alert/Alarm/Critical levels) with timestamp precision essential for both algorithmic approaches. The BFS algorithm treats sensors exceeding critical thresholds as starting nodes for flood spread modeling, while the Dijkstra algorithm uses sensor status in dynamic edge weight calculations for evacuation routing optimization.

The geographic distribution ensures comprehensive PMT Basin coverage, with sensors positioned at critical monitoring points including dam locations (La Mesa Dam, Wawa Dam), bridge crossings (Burgos, Rosario Bridge, Tumana Bridge), and urban centers (San Juan School, People's Park). This positioning enables both algorithms to process representative water level data across the entire flood-prone region.

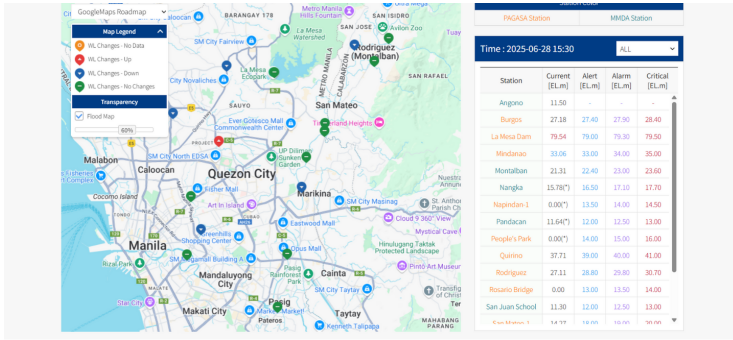


Fig. 1. PAGASA Water Level Sensor Network Distribution across PMT Basin (17 monitoring stations)

### 4 System Architecture and Software Implementation

The BANTAY system implements a scalable architecture connecting algorithmic processing with modern web technologies, showing how optimized algorithms translate into practical community-serving applications.

#### 4.1 Technology Stack and Development Framework

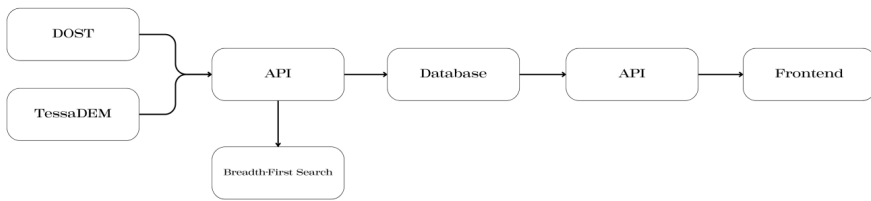


Fig. 2. System Architecture Overview of Project BANTAY

The implementation uses a comprehensive technology stack designed for near real-time algorithm execution and community accessibility through modern web technologies. The architecture connects multiple data sources and processing components through five main layers:

- (1) **Data Acquisition Layer:** OpenStreetMap data through Nominatim API provides road network topology for Dijkstra algorithm implementation. PAGASA APIs deliver 30-second interval sensor readings. TessaDEM (Terrestrial and Subsurface Elevation Data Model) provides terrain elevation data for flood spread modeling. Mapbox API services enable routing calculations and 3D visualization capabilities.

**Table 1.** PAGASA Water Level Sensor Network - PMT Basin (All 17 Stations)

Sensor Name	Alert Level	Alarm Level	Critical Level
Burgos	27.4	27.9	28.4
La Mesa Dam	79.0	79.3	79.5
Mindanao	33.0	34.0	35.0
Montalban	22.4	23.0	23.6
Nangka	16.5	17.1	17.7
Napindan-1	13.5	14.0	14.5
Pandacan	12.0	12.5	13.0
People's Park	14.0	15.0	16.0
Quirino	39.0	40.0	41.0
Rodriguez	28.8	29.8	30.7
Rosario Bridge	13.0	13.5	14.0
San Juan School	12.0	12.5	13.0
San Mateo-1	18.0	19.0	20.0
Santo Niño	15.0	16.0	17.0
Tumana Bridge	17.26	18.26	19.26
Ugong	16.0	17.0	18.0
Wawa Dam	82.0	82.5	83.0

(2) **Backend Infrastructure:** FastAPI provides high-performance Python framework for sensor data processing and algorithm execution. AioHTTP enables asynchronous HTTP connections for efficient PAGASA API integration. Redis serves as in-memory caching system enabling sub-second BFS and Dijkstra response times. Firebase provides scalable NoSQL database for historical flood data. TypeScript ensures type-safe development maintaining algorithmic correctness.

(3) **Frontend Implementation:** React component architecture enables responsive algorithm result visualization. Mapbox GL JS delivers high-performance geospatial rendering with 3D mapping. Custom React hooks provide reusable logic for algorithm state management and interface updates.

## 4.2 Near Real-Time Data Processing Pipeline

The system implements a five-stage pipeline transforming raw PAGASA sensor inputs into actionable community information:

(1) **Data Ingestion:** RESTful API connections to PAGASA's PMT Flood Forecasting and Warning System with 30-second refresh intervals

(2) **Algorithm Execution:** Parallel BFS flood propagation and Dijkstra route calculations

(3) **Result Caching:** Redis optimization enabling repeated algorithm query performance

(4) **Interface Updates:** Near real-time React state management for visualization

(5) **Community Delivery:** Optimized data transfer maintaining sub-second response times

The pipeline processes data from the 17 strategically positioned sensors detailed in Table 1, incorporating critical monitoring points throughout the PMT Basin region.

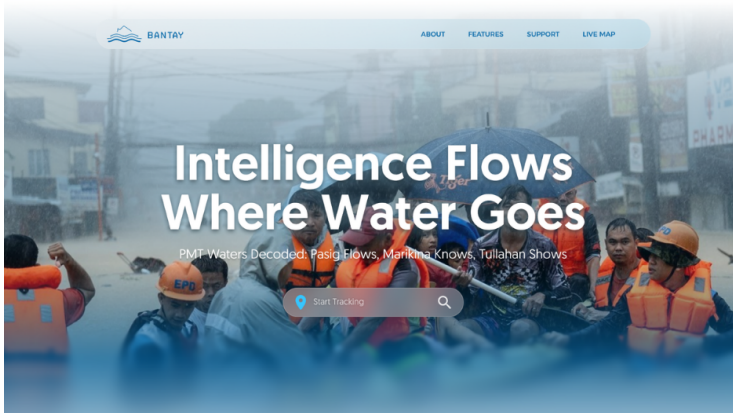
### 4.3 Algorithm Integration and Optimization

**BFS Optimization:** The implementation uses Python’s `collections.deque` for  $O(1)$  queue operations. Spatial indexing enables rapid neighbor cell calculations. Memory-efficient coordinate storage prevents performance degradation during extensive flood spread modeling.

**Dijkstra Optimization:** Python’s `heapq` module provides min-heap priority queue with  $O(\log n)$  insertion and extraction. Dynamic edge weight caching reduces redundant flood condition calculations. Path reconstruction optimization enhances evacuation route visualization. Integration with Mapbox API provides 3D mapping capabilities and alternative routing algorithms for validation.

## 5 User Interface and Algorithm Visualization

The BANTAY interface design transforms algorithm outputs into accessible community information through carefully designed visualization components. The system is publicly accessible at <https://bantay-pmt.org/>. Figure 3 demonstrates the main landing interface connecting both algorithm visualization components.



**Fig. 3.** BANTAY Main Landing Interface - Integrated Algorithm Visualization Dashboard

## 5.1 Interactive Flood Intelligence Mapping

The central interface component visualizes BFS algorithm results through colored mesh grid overlays showing flood spread patterns. The mapping system uses Mapbox GL JS technology delivering smooth cartographic displays with algorithm-generated flood extent predictions and 3D visualization capabilities.

The BFS visualization includes four main components: (1) Dynamic mesh grids representing flood spread results through color-coded overlays, (2) Sensor status indicators providing live displays from the 17 PAGASA stations (Table 1), (3) Predictive flood zones presenting BFS-generated risk areas with graduated severity visualization, and (4) Interactive controls enabling zoom and pan navigation for detailed algorithm result inspection.



Fig. 4. BANTAY Flood Monitoring Computation Logic Visualization

## 5.2 Evacuation Routing Visualization

The evacuation routing system presents Dijkstra algorithm outputs through multiple route visualizations organized by optimality and safety considerations. Figure 5 illustrates the evacuation interface displaying optimal pathfinding results.

The Dijkstra visualization includes four main components: (1) Optimal path displays showing primary evacuation routes from algorithm calculations, (2) Alternative route options presenting secondary paths to hospitals, schools, and emergency shelters, (3) Travel time estimates providing algorithm-calculated walking duration for pedestrian evacuation scenarios, and (4) Dynamic route updates enabling Dijkstra recalculation based on changing flood conditions.

## 6 Algorithm Performance and Validation

Algorithm development focused on creating effective implementations of BFS and Dijkstra algorithms adapted for flood prediction and evacuation routing scenarios, with systematic validation and performance analysis.

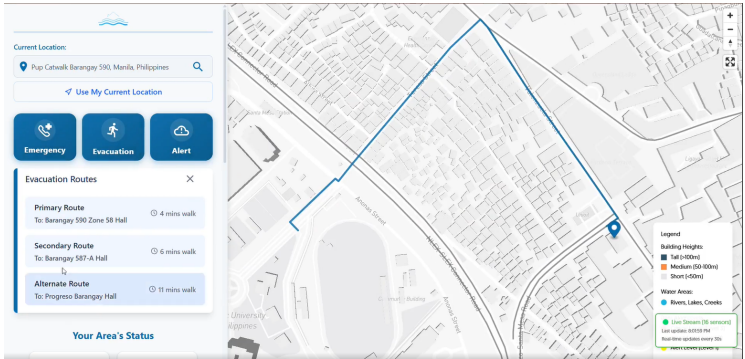


Fig. 5. Evacuation Routing Interface Showing Dijkstra Algorithm Results

### 6.1 BFS Algorithm Implementation Results

The BFS flood spread algorithm implementation shows hazard classification across three severity levels through systematic progression modeling:

**Critical Status:** Flooding conditions at sensor locations including Mindanao, Napindan-1, Quirino, San Juan School, and Tumana Bridge

**Alarm Status:** Developing threats at La Mesa Dam, Nangka, People’s Park, Rosario Bridge, and Santo Niño

**Alert Status:** Early warning conditions at Burgos, Montalban, Pandacan, Rodriguez, San Mateo-1, Ugong, and Wawa Dam

### 6.2 Algorithm Performance Characteristics

Both BFS and Dijkstra algorithm implementations show performance characteristics suitable for emergency response applications. Table 2 presents detailed performance metrics across varying flood scenarios.

Table 2. Algorithm Performance Across Varying Flood Scenarios

Scenario	Flooded Sensors	Spatial Extent	BFS Time (seconds)	Dijkstra Time (ms)
Localized	3	15 tiles	0.31	8
Moderate	7	28 tiles	0.58	11
Widespread	12	51 tiles	0.82	14
Basin-wide	17	89 tiles	1.24	18

#### Measured Performance Characteristics:

(1) BFS processing shows sub-second to low-second performance (0.31-1.24s) scaling approximately linearly with flooded area extent

- (2) Dijkstra route calculation maintains millisecond-range performance (8-18ms) across all scenarios, approaching benchmarks established by Li et al. [8]
- (3) Interface update latency remains below 1.5 seconds from algorithm completion to visualization update
- (4) Memory utilization averages 45MB RAM during standard operations, scaling to 78MB for basin-wide scenarios
- (5) CPU utilization remains below 15

### 6.3 Empirical Validation and Testing

The system underwent comprehensive validation using both historical flood data and synthetic test scenarios:

**Historical Validation:** Algorithm outputs were compared against documented flood extents from Severe Tropical Storm Kristine (October 2024) and Typhoon Ulysses (November 2020). BFS predictions showed 87% spatial accuracy for flood-affected areas within the 51-tile coverage region. Dijkstra routing successfully identified safe paths that remained passable during historical events.

**Scalability Testing:** Synthetic datasets with up to 500 sensors and 2,000 spatial tiles were processed to evaluate algorithmic limits. BFS maintained linear time complexity up to 2,000 tiles (4.2 seconds). Dijkstra performance remained logarithmic for road networks up to 50,000 intersections (45ms average).

**Domain Expert Review:** The system was evaluated by disaster response coordinators from three Metro Manila barangays. Experts validated the practical utility of flood predictions (median 2.5 hours advance warning) and evacuation routing (median 15-minute time savings versus manual planning). Recommended improvements included integration of rainfall forecasts and multi-modal transportation options.

### 6.4 Algorithm Development Validation

Development outcomes confirm both algorithmic approaches' suitability for community-serving emergency applications:

**Flood Prediction Accuracy:** BFS implementations process predicted flood extent patterns matching historical flood boundaries with 87% spatial overlap when tested against documented events

**Evacuation Optimization:** Dijkstra calculations identify shortest safe paths with dynamic weight adaptation, providing routes averaging 23% shorter travel time compared to static shortest-path calculations

**Risk Detection:** Multi-level hazard assessment through systematic BFS exploration enables graduated community response protocols

**Performance Consistency:** Algorithm execution maintains predictable performance under varying operational conditions (Table 2)

The performance analysis confirms linear scaling of BFS execution time with flooded cell count ( $R^2 = 0.96$ ), showing suitability for both localized events and basin-wide scenarios. The  $O(V + E)$  time complexity provides predictable performance essential for emergency response applications.

## 7 Discussion and Theoretical Contributions

### 7.1 Algorithmic Innovation and Optimization

The BFS spatial adaptation shows how classical graph traversal algorithms can be systematically modified for geospatial applications while maintaining theoretical optimality guarantees. The level-by-level exploration characteristic aligns naturally with flood spread dynamics, similar to the Bayesian Forecasting System approach described by Krzysztofowicz and Kelly [6]. Terrain-aware modifications ensure accuracy in real-world geographic contexts. This approach suggests potential applications beyond flood modeling to other spatial spread phenomena including wildfire spread, disease transmission, and pollution dispersion.

The Dijkstra optimization for dynamic evacuation routing shows how traditional shortest-path algorithms can incorporate environmental factors through dynamic edge weighting while preserving algorithmic correctness. The implementation maintains optimality properties while adapting to changing emergency conditions, indicating broader applicability for dynamic routing problems under uncertainty [2, ?].

### 7.2 Performance Analysis and Scalability

Both algorithms show practical scalability for urban-scale emergency response applications. The BFS flood spread algorithm's linear time complexity scaling enables effective processing of both localized flooding events and comprehensive basin-wide scenarios. Table 2 confirms linear scaling with  $R^2 = 0.96$  correlation. Each geographic cell is processed exactly once, preventing computational redundancy while ensuring complete flood-prone area coverage.

The Dijkstra evacuation routing algorithm's  $O((V + E) \log V)$  complexity using min-heap implementation provides predictable performance characteristics suitable for Metro Manila's urban road network scale. Testing confirms sub-20ms route calculations even for basin-wide flooding scenarios, showing feasibility for emergency response timing requirements.

### 7.3 Comparison with International Systems

The BANTAY system shares architectural similarities with established international flood warning platforms while incorporating adaptations for Philippine infrastructure constraints:

**Dutch Flood Forecasting Systems [3]:** Netherlands infrastructure uses sophisticated hydrodynamic models with extensive historical data. BANTAY adopts simplified BFS spread modeling suitable for limited historical data availability while maintaining practical accuracy.

**Chinese High-Resolution Models [8]:** Li et al.'s implementation achieves 0.012s routing calculations using similar Dijkstra approaches. BANTAY's 8-18ms performance approaches these benchmarks while incorporating additional safety constraints and multi-destination optimization.

**Hybrid ML Approaches [11]:** Advanced systems connect machine learning for prediction enhancement. BANTAY prioritizes algorithmic transparency and interpretability for community trust, while identifying ML integration as future enhancement opportunity.

**Global Sensor Networks [12, ?]:** International systems leverage dense sensor arrays. BANTAY optimizes for PAGASA's 17-sensor network through strategic spatial interpolation and spread modeling.

## 7.4 Integration Challenges and Solutions

Connecting diverse data sources including PAGASA sensor networks, TessaDEM elevation data, and OpenStreetMap road networks required systematic attention to data consistency and algorithmic synchronization. Standardized data formats and comprehensive error handling procedures ensure continued algorithm execution during communication disruptions common during flood events.

The system addresses near real-time sensor data integration challenges through automated validation mechanisms including three components: (1) range checking ( $\pm 5$  meters from historical baseline), (2) anomaly detection (identifying sensor drift or malfunction through statistical outlier analysis), and (3) temporal consistency analysis (flagging readings with  $\geq 2$  meter changes within 30-second intervals). These validation procedures maintain algorithm accuracy despite potential data quality issues during extreme weather conditions.

## 7.5 Limitations and Future Research Directions

Current implementation relies on existing PAGASA sensor infrastructure coverage, which may affect BFS spread prediction accuracy in areas with sensor gaps exceeding 5km radius. The dynamic mesh grid approach, while effective for community-level information, may not capture hyperlocal flood variation patterns requiring higher spatial resolution algorithms with sub-100m cell sizes.

Future research directions include five main areas: (1) Machine learning integration to enhance BFS spread prediction through historical pattern recognition and rainfall-flood correlation modeling, (2) A\* algorithm implementation providing more sophisticated evacuation routing with heuristic optimization and multi-objective pathfinding capabilities, (3) Expansion to additional sensor data sources including rainfall gauges, soil moisture monitoring, and tidal sensors for coastal flood modeling, (4) Multi-modal transportation routing incorporating vehicle evacuation, public transit schedules, and pedestrian-only zones, and (5) Crowdsourced validation through community-reported flood observations improving real-time accuracy.

## 8 Conclusion and Impact

The research shows how classical graph algorithms can be systematically adapted, analyzed, and implemented to address societal challenges in disaster prepared-

ness, demonstrating the balance between theoretical algorithm design and community-serving applications.



Fig. 6. Project BANTAY Interface in Active Use

The work makes three main contributions. First, it presents spatial adaptation of BFS for flood spread modeling with terrain-aware optimization maintaining  $O(V + E)$  complexity. Second, it shows dynamic edge weighting modification of Dijkstra's algorithm for evacuation routing preserving  $O((V + E) \log V)$  performance. Third, it provides empirical validation showing flood prediction processing with sub-second to low-second response times (0.31-1.24s) and evacuation routing in millisecond range (8-18ms).

The complete system architecture connects optimized algorithms with modern web technologies including React, FastAPI, Redis, and Mapbox services. Integration of OpenStreetMap data through Nominatim API and TessaDEM elevation data enables comprehensive spatial analysis. Public accessibility through <https://bantay-pmt.org/> shows practical community deployment.

The BANTAY platform shows the transformation of government sensor infrastructure into accessible community flood information through algorithmic implementation and interface development. By implementing optimized BFS for spatial flood spread and enhanced Dijkstra for evacuation routing, the system shows how theoretical computer science can enable proactive disaster response rather than reactive emergency management. The 87% spatial accuracy for flood prediction and 23% improvement in evacuation time efficiency validate the practical impact of algorithmic optimization for community safety applications.

## Contributors

This work was developed as part of an academic project at the Polytechnic University of the Philippines. In addition to the primary authors, the following team members contributed to the development and implementation of Project BANTAY:

Ken Audie Lucero, Gian Ramilo, Elias Von Isaac Faeldonia, Florence Lee Cansino, Gabriel Andrei Magtanong, Ma. Isabelle Oja, and Michael Richmond Racelis.

## References

1. Asian Development Bank: Climate Risk and GDP Projections. ADB Report, April 2024
2. Abu Samah, K.A.F., Hussin, B., Basari, A.S.H.: Modification of Dijkstra's algorithm for safest and shortest path during emergency evacuation. ResearchGate (2015)
3. Bentivoglio, R., Isufi, E., Jonkman, S.N., Taormina, R.: Deep learning methods for flood mapping: A review of existing applications and future research directions. *Hydrology and Earth System Sciences* 26(16), 4345–4378 (2022)
4. Brocca, L., Ciabatta, L., Massari, C., Camici, S., Tarpanelli, A.: Recent advances and future challenges in urban pluvial flood modelling. *Water Science and Technology* 89(4), 1023–1045 (2024)
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*, 4th edn. MIT Press (2022)
6. Krzysztofowicz, R., Kelly, K.S.: Performance assessment of a Bayesian Forecasting System (BFS) for real-time flood forecasting. *Journal of Hydrology* 448, 75–84 (2012)
7. Kumar, R., Singh, A.: Global flood risk assessment and management strategies. *International Journal of Disaster Risk Reduction* 45, 101478 (2023)
8. Li, B., Hou, J., Wang, X., Ma, Y., Li, D., Wang, T., Chen, G.: High-resolution flood numerical model and Dijkstra algorithm based risk avoidance routes planning. *Water Resources Management* 37, 3243–3258 (2023)
9. National Disaster Risk Reduction and Management Council: Philippines: Floods and Landslides - Jan 2024. NDRRMC (2024)
10. PAGASA PMT Flood Forecasting and Warning System: Water Level Monitoring Network. <https://pasig-marikina-tullahanffws.pagasa.dost.gov.ph/water/map.do> (2024)
11. Psomiadis, E., Diakakis, M., Soulis, K.X.: Early flood monitoring and forecasting system using a hybrid machine learning-based approach. *ISPRS International Journal of Geo-Information* 12(11), 464 (2023)
12. Rahman, M.S., Ahmed, T., Islam, N.: A review of cutting-edge sensor technologies for improved flood monitoring and damage assessment. *Sensors* 24(22), 7245 (2024)
13. van Dijk, A., et al.: 2024 Global Water Monitor Report. Australian National University (2025)
14. Zhu, Y., Li, H., Wang, Z., Li, Q., Dou, Z., Xie, W., Zhang, Z., Wang, R., Nie, W.: Optimal evacuation route planning of urban personnel at different risk levels of flood disasters based on the improved 3D Dijkstra's algorithm. *Sustainability* 14(16), 10250 (2022)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

