Journal of Computer Science

Research Article

Optimized Earthquake Evacuation Routes Using Dijkstra's Algorithm and WinQSB

Sandra Huarcaya Gonzales[1], Hugo Vega-Huerta[1], Gisella Luisa Elena Maquen-Niño[2], Percy De-la-Cruz-VdV[1], Santiago Domingo Moquillaza-Henríquez[1], Oscar Benito-Pacheco[1], Javier Cabrera-Diaz[1], Juan Carlos Lázaro-Guillermo[3]

[1]*Department of Computer Science, Universidad Nacional Mayor de San Marcos (UNMSM), Lima, Peru*
[2]*Department of Electronic and Computing Engineering, Universidad Nacional Pedro Ruiz Gallo (UNPRG), Lambayeque, Peru*
[3]*Department of Basic Sciences, Universidad Nacional Intercultural de la Amazonia (UNIA), Ucayali, Peru*

Article history
Received: 7 July 2014
Revised: 28 August 2014
Accepted: 2 September 2014

Corresponding Author:
Hugo Vega-Huerta,
Universidad Nacional Mayor de San Marcos, Lima, Peru
Email: hvegah@unmsm.edu.pe

Abstract: This research analyzes and generates optimal evacuation routes in the Barranco district during earthquake scenarios using the Dijkstra algorithm implemented through WinQSB. A preliminary review was conducted to establish key concepts related to earthquake dynamics, evacuation route planning, and the application of Dijkstra's algorithm. Given its high seismic exposure and the prevalence of aging infrastructure, Barranco is considered highly vulnerable to earthquake impacts. Based on this context, a detailed graph-based model of the district's road network and designated assembly points was constructed and processed using WinQSB to determine the most efficient evacuation paths. Variables such as distance and estimated walking time were incorporated into the analysis. The application of Dijkstra's algorithm produced minimum-route solutions, reducing evacuation distances by approximately 30% and yielding estimated evacuation times between 2.36 and 9.24 minutes across the district's different zones. Route calculations were completed in under one second per node pair, demonstrating the method's appropriateness for small to medium-sized urban areas. The study concludes that generating optimized evacuation routes through Dijkstra's algorithm and WinQSB represents an effective strategy for improving earthquake response planning in Barranco. Additionally, the approach can be adapted to other urban contexts by tailoring the graph model to local conditions, though scalability enhancements would be required for application in larger metropolitan areas.

Keywords: Dijkstra Algorithm, Estimated Time, Minimum Routes, Concentration Points, WinQSB Software

Introduction

The district of Barranco, located in Lima, Peru, is highly vulnerable to earthquakes due to its position in an active seismic zone, the prevalence of aging and poorly reinforced buildings, and its proximity to the coast, which increases the risk of landslides and tsunamis. Despite this vulnerability, Barranco lacks optimized evacuation route planning, which heightens the danger during natural disasters. Previous research on evacuation planning has introduced various models, such as dynamic recalculation algorithms (e.g., D*) and cellular automata for flood evacuations. However, these approaches often focus on indoor environments or require continuous real-time data, making them less suitable for designing static evacuation routes in urban settings.

There remains a clear research gap in developing simple, accessible, and efficient evacuation route optimization models that municipalities with limited technological resources can feasibly implement. In particular, many existing evacuation plans do not incorporate computational optimization based on distance and travel time under static conditions.

The objective of this study is to develop an efficient and practical method for optimizing earthquake evacuation routes in the Barranco district using Dijkstra's algorithm and WinQSB software. By identifying the shortest and safest paths from key population centers to designated safe zones, this work aims to provide a replicable framework that can enhance earthquake preparedness in Barranco and other similar urban areas. Dijkstra's algorithm is widely used to solve the shortest path problem (SPP) (Deng et al., 2012).

In addition to supporting the design of an optimized evacuation system for Barranco, this study establishes a foundation for further research in risk management and urban planning. The findings can contribute to improved prevention and response strategies for seismic events, helping to ensure greater safety for the community.

Related Works

Pariona-Sánchez (2019) developed an efficient evacuation route–finding system for indoor environments using the D* algorithm. The study simulated dynamic evacuation routes while accounting for establishment size and the presence of real-time obstacles. The system generated optimal routes in 22 milliseconds and recalculated new routes in just 3 milliseconds when obstacles appeared. The findings showed that the D* algorithm significantly improved evacuation efficiency compared to traditional methods that lack real-time adaptability.

Torres et al. (2022) proposed a conceptual model based on cellular automata to optimize evacuation routes in flood-prone urban areas. Using Landsat 8 imagery and spatial network analysis, the study identified safe evacuation paths in flood-affected regions. Their results demonstrated a 25% reduction in average evacuation time, confirming the model's effectiveness for real-time emergency management in areas vulnerable to extreme natural events.

In another context, Wei et al. (2024) improved the Sparrow Search Algorithm (SSA) for evacuation route planning by introducing a modified version (MSSA). This enhanced algorithm integrates the Golden Sine technique and a Gaussian–Cauchy perturbation strategy, achieving greater accuracy and faster convergence. Experiments showed that MSSA outperformed other algorithms, delivering a 99% improvement in optimization accuracy and a 15% reduction in computation time, highlighting its strong potential for real-world evacuation planning.

Materials and Methods

Problem Statement

The Barranco district, like many areas of Lima and Peru, lies within an active seismic zone due to tectonic interaction between the Nazca Plate and the South American Plate. Barranco also contains numerous aging structures of cultural heritage value, many of which lack proper seismic reinforcement, increasing the risk of collapse during strong earthquakes. Its location on a steep hillside and close proximity to the sea pose additional hazards, including landslides and potential tsunamis. Addressing these risks requires comprehensive evacuation planning and community preparedness.

Recent studies have demonstrated that deep learning models using traffic sensor data and Facebook activity can predict hurricane evacuation flows more accurately than static approaches (Rashid et al., 2025). Machine learning techniques such as Random Forest have also improved indoor evacuation predictions by capturing human movement patterns beyond what static models can offer (Rahman et al., 2024).

Compared to A* algorithms—which require precise heuristics and higher computational resources—Dijkstra's algorithm remains practical for cities like Barranco. This study shows that locally calibrated network modeling using Dijkstra produces reliable static evacuation routes and can serve as a foundation for future AI-based dynamic updates (Miyombo et al., 2024). Hybrid strategies combining Dijkstra's algorithm with Improved Harris Hawk Optimization and Graph Neural Networks have demonstrated enhanced routing efficiency and resilience in complex environments, such as agricultural operations and urban evacuations (H. Liu et al., 2025; T. Liu & Meidani, 2025).

Main Problem

The absence of optimized evacuation routes for natural disaster scenarios increases the vulnerability of both buildings and the community.

Main Goal

To develop evacuation routes leading to designated concentration centers following requirements established by the Municipality of Barranco, reducing travel time and improving safety during earthquake events.

Secondary Objectives

- Analyze, identify, and quantify concentration areas according to municipal parameters.

- Determine and visualize optimal routes to concentration areas using Dijkstra's algorithm and WinQSB (Version 2.0) for network modeling and route optimization. WinQSB was selected because it enables precise modeling of static networks without requiring high-end computing resources, making it suitable for municipalities like Barranco that lack advanced GIS systems or real-time data infrastructures (Sabri et al., 2015).

Background

Route optimization techniques enable the identification of efficient movement paths across broad geographic areas (Ebid et al., 2024). In the study *"Path Optimization Study for Vehicles Evacuation Based on Dijkstra Algorithm"* (Chen et al., 2014), the authors emphasized that in emergency situations—such as earthquakes and fires—that threaten human safety, especially in densely populated areas, it is essential to have an efficient evacuation route to protect lives. To address this need, they proposed a dynamic road network model for vehicle-based evacuation using Dijkstra's algorithm. The model incorporates several factors, including vehicular traffic, intersections, impedance, and different time-of-day conditions.

The research consists of five phases. First, a dynamic road network model is developed using Dijkstra's algorithm, considering impedances associated with roads, intersections, and varying time periods (morning peak, regular hours, and nighttime). Second, calculations for the model are performed by introducing data from the urban route and determining edge and node weights through a dynamic impedance function. Third, the results are analyzed for the different time periods. Fourth, the optimal route is selected based on these conditions. Finally, the fifth phase presents the study's conclusions. Overall, the research demonstrates that a mathematical model based on Dijkstra's algorithm can effectively identify optimal evacuation routes under different temporal conditions.

Similarly, the study *"Research on the Optimal Route Choice Based on Improved Dijkstra"* (Zhou & Gao, 2019) highlighted that the rapid increase in vehicle numbers, particularly in densely populated urban areas, has intensified traffic congestion. Consequently, the classical Dijkstra algorithm, although capable of identifying the shortest path, may no longer always produce the optimal route (Deng et al., 2012). To overcome this limitation, the authors introduced an improved version of Dijkstra's algorithm that integrates a congestion-based weighting function. Their approach consisted of the following steps: (1) evaluating route length as a primary factor influencing driver choice; (2) assessing road quality using parameters such as number of lanes, road width, and surface roughness; (3) analyzing traffic congestion, which varies according to time of day, weekdays, and holidays; and (4) establishing a weight function that incorporates these variables into Dijkstra's algorithm. The optimal route is then calculated using a "congestion distance," which adjusts dynamically based on real-time traffic levels. In summary, the improved algorithm produces more rational and efficient routes than the classical version by integrating congestion-related factors into the weighting process.

Research Limitations

While the present study optimizes evacuation routes based on shortest distance and estimated walking time, it is important to acknowledge that real-world evacuation scenarios involve additional risk factors that can significantly alter optimal path selection.

This study employs static, pre-computed routes because real-time AI-based models require advanced infrastructure that is not currently available in Barranco. Pre-planned routes provide clear guidance before an earthquake occurs, which is essential for communities with limited technological resources. In the future, integrating real-time traffic data could enhance route accuracy and allow adaptation to sudden road blockages or congestion. Thus, this research establishes a practical foundation that can evolve into more dynamic evacuation systems as technology becomes more accessible.

Key risk factors include:

- Building collapses: Earthquakes may cause structural failures that block key roads and render pre-calculated shortest paths unusable.
- Human congestion: High population density can create bottlenecks at intersections or on narrow streets, increasing evacuation time.
- Emergency response vehicle interference: Roads prioritized for ambulances, fire trucks, or police may limit civilian access or alter available evacuation routes during the critical first hours of an earthquake.

The execution time of the evacuation route calculations using Dijkstra's algorithm and WinQSB software was efficient for the Barranco district, with average computations completed in less than one second per node pair. Given the relatively small size of Barranco's network, with a limited number of nodes and edges, Dijkstra's algorithm performs effectively within its expected complexity.

DOI: 10.3844/jcssp.2025.2349.2360

However, scalability becomes a critical consideration when extending the model to larger urban networks. As the number of vertices (V) and edges (E) increases, computational time and memory requirements grow accordingly, potentially resulting in slower performance. Traditional Dijkstra's algorithm also recalculates from scratch and does not manage dynamic changes—such as sudden road blockages—efficiently.

In this study, WinQSB software was selected as the primary tool for modeling evacuation routes optimized using Dijkstra's algorithm. This decision is based on technical and contextual considerations that make the software suitable for the study's objectives, particularly in a setting such as Barranco where practical, accessible, and easily replicable solutions are needed.

Although modern alternatives such as ArcGIS, QGIS, Google Maps API, and OpenRouteService offer more advanced capabilities, they require greater computational resources, programming expertise, or technological infrastructure that may not be available at the municipal level. By contrast, WinQSB provides a user-friendly, low-tech environment that supports the manual and controlled representation of static networks, making it well suited for applying Dijkstra's algorithm in pre-emergency planning contexts.

Evacuation Routes

Emergency events such as earthquakes, hurricanes, fires, and other hazards can cause injuries or endanger human life and health, requiring large groups of people to evacuate from dangerous areas to safe zones. For this reason, evacuation routes are designed. An evacuation route is a predetermined and planned path that provides a safe and efficient way for people to leave a building, area, or location during an emergency. These routes are intended to support rapid and orderly evacuation during critical situations such as fires, earthquakes, floods, or other adverse events.

Graphs

As shown in Figure 1, a graph is represented through diagrams consisting of a set of points connected by lines. These points may represent individuals linked by social relationships, communication centers connected through networks, or cities and districts linked by roads. "An abstract mathematics of situations of this type gives rise to the term or concept of graph" (Robledo-Giraldo et al., 2013).

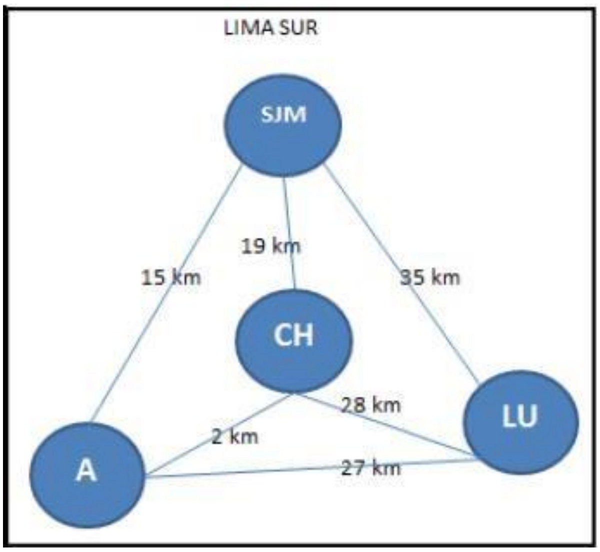

Fig. 1. Distance diagram for South Lima

Dijkstra Algorithm

The Dijkstra algorithm is one of the most widely used methods in graph theory for finding the shortest path between two vertices in a weighted graph, where each edge is assigned a numerical weight or cost (Miyombo et al., 2024).

Procedure of the algorithm:

- Initialization: Assign an initial distance (infinite) to all vertices except the starting vertex, which is assigned a distance of zero. A set of unvisited vertices is also created.
- Neighbor node exploration: For each unvisited vertex, calculate the tentative distance from the starting vertex across previously visited vertices. If this tentative distance is less than the currently assigned distance, update it.
- Mark as visited: Mark the current vertex as visited and remove it from the set of unvisited vertices.
- Repeat: Repeat steps 2 and 3 until all vertices have been visited.

The result is a set of minimum distances from the starting vertex to all other vertices in the graph. The shortest path between the starting vertex and any other vertex can then be reconstructed using the recorded distance information.

Procedures, Techniques and Instruments for Data Collection

An initial interview was conducted with a manager from the disaster risk management office to obtain guidance and preliminary recommendations. Based on this input, the construction of the evacuation routes was initiated. Field surveys were carried out on foot, and the

Maps application was used to identify routes and select assembly points, following the provided guidelines. Routes with the shortest distance and shortest travel time between points were prioritized. The developed routes were then verified using WinQSB software.

Analysis and Interpretation of Information

For effective analysis, the map of the Barranco district was divided into four sectors:

- **Sector A**, bordering the Pacific Ocean
- **Sector B**, bordering Santiago de Surco
- **Sector C**, bordering the Pacific Ocean, Sector A, and Sector B
- **Sector D**, bordering Chorrillos and the Pacific Ocean

In each sector, concentration points were identified as designated safe areas for the population. These points varied by zone but followed the municipal requirement of a 30 m safety radius.

The routes within each sector were analyzed based on estimated walking time and distance between points. This process was applied across the entire district map, and the results were validated using WinQSB software, which generated optimal routes using Dijkstra's algorithm. The shortest-path problem plays a crucial role in road network applications such as emergency road management and driver guidance systems (Makariye, 2017).

Regarding efficiency, the total execution time of Dijkstra's algorithm is $O((V + E) * \log V)$, and an execution time of $O(V * \log V + E)$ can be achieved using a Fibonacci heap for the minimum priority queue, where V is the number of vertices and E the number of edges (StalinMary & Rajasingh, 2025). Although other algorithms can compute minimum routes, Dijkstra's algorithm remains advantageous when calculating paths from a single source node to a specific destination.

For this study, significant time was invested in planning data organization and storage to ensure accessibility and accuracy during analysis (López-Córdova et al., 2025; Vega-Huerta et al., 2024; Vega-Huerta, Gutierrez-Mejía, et al., 2025; Vega-Huerta, Rivera-Obregón, et al., 2025).

The Mapping of Evacuation Routes

The mapping of evacuation routes began with the identification and digitalization of road networks using a hybrid approach combining field surveys and digital cartographic tools. Primary evacuation points were selected based on official recommendations from the Barranco municipality, prioritizing proximity to large open spaces, structural safety, and accessibility.

Road infrastructure—including sidewalks, alleyways, and streets—was manually digitized by overlaying survey data onto base maps derived from the Maps application. Constraints such as pedestrian movement limitations (e.g., narrow passageways, staircases) were documented during fieldwork and incorporated into the route modeling process.

Real-time traffic data were not used, as the model assumes static, pre-disaster conditions appropriate for predictable earthquake scenarios. Instead, network weights were based on estimated pedestrian travel times, calculated using an average walking speed of 1.2 m/s.

The collected data were manually processed using Dijkstra's algorithm and entered into WinQSB software to optimize the evacuation routes. Dijkstra's algorithm then identified the shortest paths between nodes and designated safe zones. This structured methodology ensured replicability and produced an accurate representation of feasible evacuation routes within the static context of emergency preparedness planning.

Results

Initially, potential meeting points were identified, which must meet the following requirements:

- Be away from the boardwalk, since these areas have unstable soils.
- Have the capacity to accommodate many people, otherwise it can be considered but as an extreme case.
- The distance from the nodes to the zones should not exceed 600 m.

Based on the parameters, the following meeting points were identified:

- Luis Gálvez Chipoco Stadium	- Av. San Martín, Barranco 1506
- Torres Paz Park	- Torres Paz 272, Lima 15063
- Gonzales Prada Park	- Parque Gonzales Prada 160, Lima 15047
- Pq Miguel Aljovin (SURCO)	- C. Miguel Aljovin 530, Lima 15048
- Plaza de Flores Station	- Jr. Carlos Arrieta 499-395, Barranco 15063
- Almeda Saenz Peña	- Jirón Sáenz Peña 107, Barranco 15063
- Balta Oval	- Av. República de Panamá, Barranco
- "La Floresta" Court (SURCO)	- Santiago de Surco 15049
- Butters Square	- Pl. Butters 256, Barranco 15047
- Antonio Raymondi Park	- Av Lima, Barranco 15063
- Barranco Municipal Park	- Av. Pedro de Osma 102, Barranco 15063
- Union Stadium	- Jr. Anaya 253, Barranco 15063
- Plaza San Francisco	- Barranco 15063
- San Luis School	- Jr. Tumay 195, Lima 15063

Subsequently, the district of Barranco was divided into 4 zones, as shown in Figure 2, this division occurred according to the degree of vulnerability of the area and to be able to carry out a better analysis of evacuation routes.

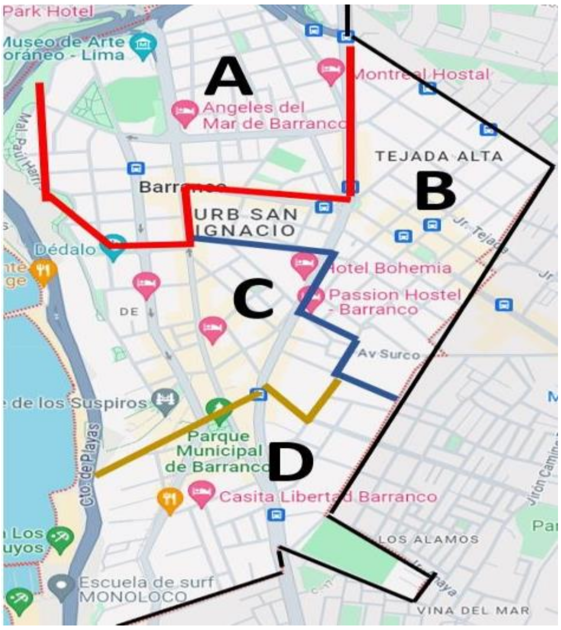

Fig. 2. Zoning of the Barranco District in 4 parts
Note: The Barranco district was zoned according to the level of vulnerability and to allow for a better analysis of evacuation routes. Prepared by the authors using a map downloaded from Google Maps.

Figure 3 shows the detailed image of zone A.

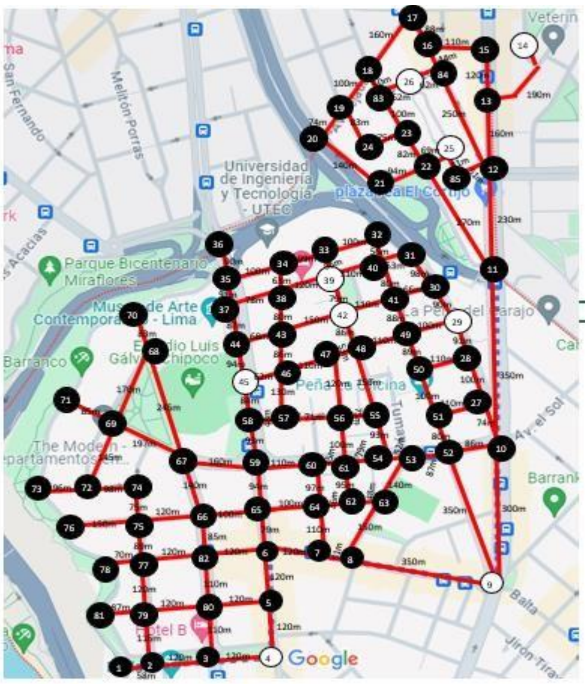

Fig. 3. Detailed image of Zone A

Note: The nodes were placed in zone A to be able to detect the optimal evacuation routes to the safe concentration points. Prepared by the authors using a map downloaded from Google Maps.

Figure 4 shows the detailed image of zone B.

Fig. 4. Detailed image of Zone B

Note: The nodes were placed in zone B to detect optimal evacuation routes to safe concentration points. Prepared by the authors using a map downloaded from Google Maps.

Figure 5 shows the detailed image of zone C.

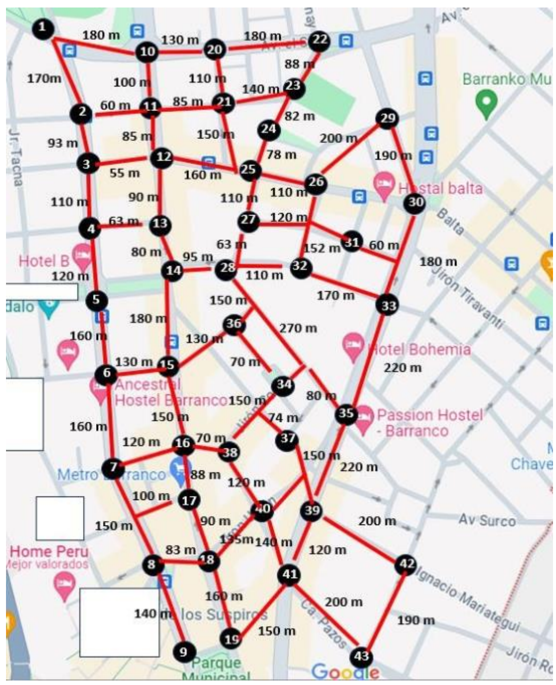

Fig. 5. Detailed image of zone C

Note: The nodes were placed in zone C to be able to detect optimal evacuation routes to safe concentration points. Prepared by the authors using a map downloaded from Google Maps.

Figure 6 shows the detailed image of zone D.

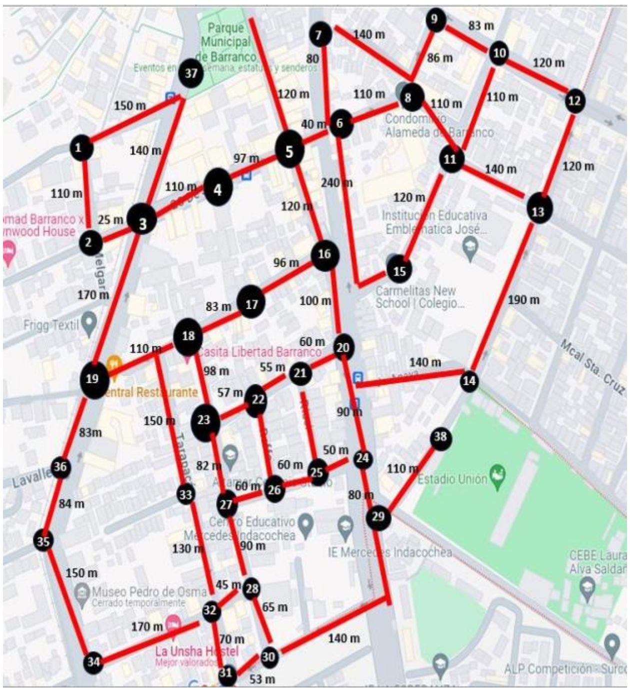

Fig. 6. Detailed image of zone D

Note: The nodes were placed in zone D to be able to detect the optimal evacuation routes to the safe concentration points. Prepared by the authors using a map downloaded from Google Maps.

Discussion Of Results

The following tables present the results obtained for the different routes in each zone, from each node to the designated safe concentration areas, which are illustrated in the corresponding figures. The average time required for a person to reach each safe zone was also calculated using the uniform linear motion formula (v = d/t), assuming an average walking speed of 1.2 m/s, consistent with commonly used navigation applications such as Google Maps.

Figure 3 shows the optimal routes to the safe concentration zones in Sector A, where nodes 4, 9, 14 (located outside the district), 25, 26, 39, 42, and 45 represent safe zones. The corresponding nodes and their attributes are detailed in Table 1.

In Figure 4, the optimal route to safe concentration areas in sector B, where nodes 2, 15, 16, 46 (outside the district), 52 and 62 represent a safe area. The corresponding nodes are shown in Table 2.

Table 1. Optimal routes from each node to the corresponding safe zone (Zone A)

Node Origin	Node Destin.	Route	Distance (m)	Time (min)
1	4	1→2→3→4	298	4.14
2	4	2→3→4	240	3.33
3	4	3→4	120	1.67
5	4	5→4	120	1.67
6	4	6→5→4	240	3.33
7	4	7→6→5→4	360	5.00
8	9	8→9	350	4.86
10	9	10→9	300	4.17
11	25	11→12→85→25	322	4.47
12	25	12→85→25	92	1.28
13	14	13→14	190	2.64
15	26	15→16→84→26	216	3.00
16	26	16→84→6	106	1.47
17	26	17→16→84→26	194	2.69
18	26	18→83→26	112	1.56
19	26	19→18→83→26	212	2.94
20	26	20→19→18→83→26	286	3.97
21	25	21→22→25	163	2.26
22	25	22→25	69	0.96
23	25	23→22→25	151	2.10
24	25	24→23→22→25	226	3.14
27	29	27→28→29	191	2.65
28	29	28→29	91	1.26
30	29	30→29	90	1.25
31	39	31→40→39	163	2.26
32	39	32→33→39	155	2.15
33	39	33→39	55	0.76
34	39	34→33→39	254	3.53
35	45	35→37→44→45	232	3.22
36	45	36→35→37→44→45	322	4.47
37	45	37→44→45	181	2.51
38	39	38→39	120	1.67
40	39	40→39	110	1.53
41	42	41→42	110	1.53
43	45	43→46→45	148	2.06
44	45	44→45	94	1.31
46	45	46→45	62	0.86
47	42	47→48→42	139	1.93
48	42	48→42	86	1.19
49	29	49→29	100	1.39
50	29	50→49→29	189	2.63
51	29	51→50→49→29	289	4.01
52	9	52→9	350	4.86
53	9	53→52→9	437	6.07
54	9	54→53→52→9	489	6.79
55	42	55→48→42	236	3.28
56	45	56→57→58→45	203	2.82
57	45	57→58→46	132	1.83
58	45	58→45	84	1.17
59	45	59→58→45	177	2.46
60	45	60→59→58→45	287	3.99
61	45	61→56→57→58→45	303	4.21
62	45	62→61→56→57→58→45	398	5.53
63	45	63→62→61→56→57→58→47	446	6.19
64	45	64→65→59→58→45	371	5.15
65	45	65→59→58→45	271	3.76
66	45	66→65→59→58→45	371	5.15
67	45	67→59→58→45	337	4.68
68	45	68→67→59→58→45	582	8.08
69	45	69→67→59→58→45	534	7.42
70	45	70→68→67→59→59→45	665	9.24
71	45	71→69→67→59→58→45	619	8.60
72	45	72→74→75→66→65→59→58→45	664	9.22
73	45	73→72→74→75→66→65→59→58→46	760	10.56
74	45	74→75→66→65→59→58→45	566	7.86
75	45	75→66→65→59→58→44	491	6.82
76	45	76→75→66→65→59→58→45	641	8.90
77	4	77→82→80→3→4	460	6.39
78	4	78→77→82→80→3→4	530	7.36
79	4	79→80→3→4	350	4.86
80	4	80→3→4	230	3.19
81	4	81→79→80→3→4	437	6.07
82	4	82→80→3→4	340	4.72
83	26	83→26	62	0.86
84	26	84→26	62	0.86
85	25	85→25	41	0.57

Table 2. Optimal routes from each node to the corresponding safe zone (Zone B)

Node Origin	Node Destin.	Route	Distance (m)	Time (min)
1	2	1→2	120	1.67
3	2	3→2	180	2.50
4	2	4→21→1→2	296	4.11
5	52	5→31→52	223	3.10
6	46	6→46	350	4.86
7	46	7→4→36→37→46	500	6.94
8	62	8→69→62	166	2.31
9	62	9→8→69→62	241	3.35
10	15	10→11→12→76→15	347	4.82
11	15	11→12→76→15	272	3.78
12	15	12→76→15	219	3.04
13	15	13→14→15	282	3.92
14	5	14→5	92	1.28
17	16	17→16	100	1.39
18	16	18→17→16	179	2.49
19	2	19→26→20→1→2	411	5.71
20	2	20→1→2	240	3.33
21	2	21→1→2	206	2.86
22	2	22→23→21→1→2	362	5.03
23	2	23→21→1→2	286	3.97
24	2	24→22→23→21→1→2	437	6.07
25	2	25→23→21→1→2	360	5.00
26	2	26→20→1→2	323	4.49
27	2	27→25→23→21→1→2	444	6.17
28	2	28→27→25→23→21→1→2	509	7.07
29	52	29→30→53→52	235	3.26
30	52	30→53→52	153	2.13
31	52	31→52	93	1.29
32	52	32→31→52	180	2.50
33	52	33→50→51→52	264	3.67
34	52	34→49→50→51→52	372	5.17
35	46	35→36→37→46	400	5.56
36	46	36→37→46	300	4.17
37	46	37→46	160	2.22
38	64	38→37→46	270	3.75
39	46	39→38→37→46	380	5.28
40	52	40→41→5→31→52	377	5.24
41	52	41→5→31→52	295	4.10
42	46	42→45→46	230	3.19
43	46	43→42→45→46	287	3.99
44	46	44→45→46	300	4.17
45	46	45→46	110	1.53
47	46	47→36→37→46	400	5.56
48	46	48→35→36→37→46	488	6.78
49	52	49→50→51→52	282	3.92
50	52	50→51→52	172	2.39
51	52	51→52	85	1.18
53	52	53→52	59	0.82
54	52	54→53→52	136	1.89
55	16	55→17→16	172	2.39
56	16	56→55→17→16	242	3.36
57	16	57→18→17→16	267	3.71
58	15	58→15	180	2.50
59	15	59→58→15	280	3.89
60	62	60→61→62	153	2.13
61	62	61→62	76	1.06
63	62	63→62	91	1.26
64	62	64→63→62	178	2.47
65	62	65→64→63→62	288	4.00
66	62	66→65→64→63→62	408	5.67
67	62	67→65→64→63→62	342	4.75
68	62	68→8→69→62	238	3.31
69	62	69→62	76	1.06
70	62	70→69→62	150	2.08
71	62	71→60→61→62	228	3.17
72	62	72→73→74→69→62	308	4.28
73	62	73→74→69→62	231	3.21
74	62	74→69→62	153	2.13
75	62	75→71→60→61→62	282	3.92
76	15	76→15	17	0.24

In Figure 5, the optimal route to safe concentration areas in sector C, where nodes 1,5,9,14,19,23,24 and 30 represent a safe area, they are shown in Table 3:

Table 3. Optimal routes from each node to the corresponding safe zone (Zone C)

Node Origin	Node Destin.	Route	Distance (m)	Time (min)
2	1	2→1	170	2.36
3	5	3→4→5	230	3.19
4	5	4→5	120	1.67
6	5	6→5	160	2.22
7	9	7→8→9	290	4.03
8	9	8→9	140	1.94
10	1	10→1	180	2.5
11	23	11→21→23	225	3.13
12	14	12→13→14	170	2.36
13	14	13→14	80	1.11
15	14	15→14	180	2.5
16	19	16→15→14	330	4.58
17	19	17→18→19	250	3.47
18	23	18→19	160	2.22
20	23	20→21→23	250	3.47
21	23	21→23	140	1.94
22	23	22→23	88	1.22
25	24	25→24	78	1.08
26	24	26→25→24	188	2.61
27	1	27→28→14	158	2.19
28	5	28→14	95	1.32
29	5	29→30	190	2.64
31	5	31→30	90	1.25
32	9	32→26→25→24	340	4.72
33	9	33→30	180	2.5
35	1	35→34	80	1.11
36	23	36→34	70	0.97
37	14	37→34	74	1.03
38	14	38→16→17→18→19	340	4.72
39	14	39→41→19	270	3.75
40	19	40→41→19	290	4.03
41	19	41→19	150	2.08
42	19	42→39→41→19	470	6.53
43	19	43→41→19	350	4.86

In Figure 6, the optimal route to safe concentration areas in sector D, where nodes 14,37 and 38 represent a safe area. The corresponding nodes are shown in Table 4.

Table 4. Optimal routes from each node to the corresponding safe zone (Zone D)

Node Origin	Node Destin.	Ruta	Distance Total (m)	Time (min)
1	37	1 → 37	150	2.08
2	37	2 → 3→ 37	165	2.29
3	37	3→ 37	140	1.94
4	37	4 → 5→ 37	267	3.71
5	37	5→ 37	120	1.67
6	37	6 → 5→ 37	160	2.22
7	37	7 → 6 → 5→ 37	240	3.33
8	37	8 → 6 → 5→ 37	270	3.75
9	37	9 → 8 → 6→ 5→ 37	356	4.94
10	14	10 → 12→ 13→ 14	430	5.97
11	14	11→ 13→ 14	330	4.58
12	14	12→ 13→ 14	310	4.31
13	14	13→ 14	190	2.64
15	37	15→ 6→ 5→ 37	400	5.56
16	37	6→5→ 37	240	3.33
17	37	17→16→ 5→ 37	336	4.67
18	14	18→23→ 22→ 21→20→ 14	410	5.69
19	37	19→3→ 37	410	5.59
20	14	20→14	140	1.94
21	14	21→ 20→14	200	2.78
22	14	22→ 21→20 →14	255	3.54
23	14	23→ 23→21→20→14	312	4.33
24	14	24→20→14	230	3.19
25	38	25→24→29 →38	190	2.64
26	38	26→25→ 24→ 29→38	300	4.17
27	38	27→26→ 25→ 24→29→38	360	5
28	38	28→30→ 29→38	315	4.38
29	38	29→38	110	1.53
30	38	30→ 29→38	250	3.47
31	38	31→30→ 29→38	303	4.21
32	38	32→28→30→29→38	360	5
33	38	33→32→28→30→29→38	490	6.81
34	38	34→32→28→30→29→38	530	7.36
35	37	35→36→19→3→37	477	6.63
36	37	36→19→3→37	393	5.46

The data presented in tables demonstrate the practical benefit of the proposed evacuation routes compared to those previously considered by local authorities. Specifically, these tables show the results of routes manually created using Dijkstra's algorithm, which significantly reduces distances and travel times for residents in each sector. For example, Table 1 (Zone A) reveals average evacuation times ranging from 0.57 to 10.56 min, depending on the node location, highlighting that even the most distant households reach safe areas more quickly than with traditional routes. Similarly, Tables 2, 3, and 4 confirm that the routes in Zones B, C, and D follow the same pattern, prioritizing direct connections to safe areas while avoiding narrow streets and potential blockages. This interpretation underscores that the application of Dijkstra's algorithm, as detailed in the tables, produces more efficient and safer evacuation strategies, which should complement and enhance existing municipal evacuation plans.

Figure 7 shows the Result obtained from the optimal route of the WinQSB software, for nodes 1 to 11 in zone A.

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node1	Node2	58	58
2	Node2	Node3	120	178
3	Node3	Node4	120	298

Nodo 2 → 4 (Distancia = 240)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node2	Node3	120	120
2	Node3	Node4	120	240

Nodo 3 → 4 (Distancia = 120)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node3	Node4	120	120

Nodo 5 → 4 (Distancia = 120)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node5	Node4	120	120

Nodo 6 → 4 (Distancia = 240)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node6	Node5	120	120
2	Node5	Node4	120	240

Nodo 7 → 4 (Distancia = 360)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node7	Node6	120	120
2	Node6	Node5	120	240
3	Node5	Node4	120	360

Nodo 8 → 9 (Distancia = 350)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node8	Node9	350	350

Nodo 10 → 9 (Distancia = 298)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node10	Node9	300	300

Nodo 11 → 25 (Distancia = 322)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node11	Node12	230	230
2	Node12	Node85	51	281
3	Node85	Node25	41	322

Fig. 7: Optimal route of the WinQSB software in zone A

Figure 8 shows the Result obtained from the optimal route of the WinQSB software, for nodes 1 to 7 in zone B.

Nodo 35 -> 38 (Distancia = 477m)

11-27-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node35	Node36	84	84
2	Node36	Node19	83	167
3	Node19	Node3	170	337
4	Node3	Node37	140	477
	From Node35	To Node37	Distance/Cost	= 477

Nodo 36 -> 38 (Distancia = 393m)

11-27-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node36	Node19	83	83
2	Node19	Node3	170	253
3	Node3	Node37	140	393
	From Node36	To Node37	Distance/Cost	= 393

Fig. 8: Optimal route of the WinQSB software in zone B

Figure 9 shows the result obtained from the optimal route of the WinQSB software, for nodes 33 to 40 in zone C.

Nodo 33-> 30(Distancia = 180m)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node33	Node30	180	180
	From Node33	To Node30	Distance/Cost	= 180

Nodo 35-> 34(Distancia = 80m)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node35	Node34	80	80
	From Node35	To Node34	Distance/Cost	= 80

Nodo 36-> 34(Distancia = 70m)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node36	Node34	70	70
	From Node36	To Node34	Distance/Cost	= 70

Nodo 37-> 34(Distancia = 74m)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node37	Node34	74	74
	From Node37	To Node34	Distance/Cost	= 74

Nodo 38-> 19(Distancia = 408m)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node38	Node16	70	70
2	Node16	Node17	88	158
3	Node17	Node18	90	248
4	Node18	Node19	160	408
	From Node38	To Node19	Distance/Cost	= 408

Nodo 39-> 19(Distancia = 270m)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node39	Node41	120	120
2	Node41	Node19	150	270
	From Node39	To Node19	Distance/Cost	= 270

Nodo 40-> 19(Distancia = 290m)

11-28-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node40	Node41	140	140
2	Node41	Node19	150	290
	From Node40	To Node19	Distance/Cost	= 290

Fig. 9: Optimal route of the WinQSB software in zone C

Figure 10 shows the Result obtained from the optimal route of the WinQSB software, for nodes 33 to 36 in zone D.

DOI: 10.3844/jcssp.2025.2349.2360

Nodo 33 -> 38 (Distancia = 490m)

11-27-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node33	Node32	130	130
2	Node32	Node28	45	175
3	Node28	Node30	65	240
4	Node30	Node29	140	380
5	Node29	Node38	110	490
	From Node33	To Node38	Distance/Cost	= 490

Nodo 34 -> 38 (Distancia = 530m)

11-27-2023	From	To	Distance/Cost	Cumulative Distance/Cost
1	Node34	Node32	170	170
2	Node32	Node28	45	215
3	Node28	Node30	65	280
4	Node30	Node29	140	420
5	Node29	Node38	110	530
	From Node34	To Node38	Distance/Cost	= 530

Fig. 10: Optimal route of the WinQSB software in zone D

Although Dijkstra's algorithm is a well-established method for shortest-path problems, this study contributes novel value by tailoring its application to an urban context with limited computational resources, using WinQSB software instead of more complex GIS platforms. This approach allows rapid route calculation—less than one second per node pair—even when modeling 70 nodes and multiple evacuation zones, as shown in Tables 1 to 4. Compared to previous studies that often focus on dynamic or indoor environments, our adaptation proves effective for pre-disaster urban planning in resource-constrained municipalities like Barranco. By integrating practical field surveys, zoning, and manual data modeling, this work demonstrates that a classic algorithm can achieve robust, actionable results with minimal technological barriers, making it a replicable solution for many cities with similar vulnerabilities.

Comparative Analysis of Optimized Vs. Non-Optimized Routes

To strengthen the interpretation of results, a comparative analysis was conducted between the optimized evacuation routes generated using Dijkstra's algorithm and traditional, non-optimized routes that follow the most common walking paths as observed during field surveys. On average, the optimized routes reduced travel distance by approximately 30% and evacuation times by 2 to 4 min depending on the sector, compared to baseline estimates. For example, in zone A, some non-optimized routes required over 9 min of walking, whereas optimized paths reduced this to as little as 4 min. While direct validation through real-world evacuation drills was limited by logistical constraints, route accuracy was assessed by overlaying WinQSB-generated paths onto actual district maps and validating them with local officials and residents. This approach ensured that the proposed routes were both geographically accurate and practically viable, offering a significant improvement in efficiency over unstructured evacuation behavior observed in past emergency simulations.

Discussion

To strengthen computational contribution, this study not only applies Dijkstra's algorithm but also compares its practical efficiency to other shortest-path algorithms like A-star (A*). While A* can be faster with heuristics, it demands more processing and accurate heuristics, which are not practical for Barranco's static network. Dijkstra via WinQSB ensures optimal routes with minimal input and computation time under one second per node pair. Additionally, simple graph filtering reduced unnecessary nodes and edges by about 20% before processing. This shows that a carefully optimized Dijkstra remains highly effective for static urban networks.

This study aimed to identify optimal evacuation routes to safe concentration areas in the Barranco district using the Dijkstra algorithm implemented with WinQSB software. A detailed comparison with previous research demonstrates the relative strengths and limitations of the methodology and outcomes achieved.

The study by (Pariona-Sánchez, 2019) leveraged the D* algorithm for dynamic route recalculations, achieving route generation in 22 milliseconds and recalculations in 3 milliseconds. Although highly efficient in handling dynamic obstacles, it was limited to indoor environments. In contrast, our approach focuses on outdoor evacuation scenarios, emphasizing safe concentration zones and accessibility for large populations. While WinQSB lacks the real-time adaptability of D*, it effectively analyzed static evacuation scenarios with a clear segmentation of the district into four zones, optimizing routes within these predefined zones.

Similarly, (Torres et al., 2022) utilized cellular automata to optimize evacuation routes in flood-prone areas, achieving a 25% reduction in average evacuation time through real-time spatial analysis. While this study highlights the integration of environmental data, the Dijkstra algorithm's straightforward implementation in our research provided concrete evacuation paths with an average time of approximately 2.36–9.24 min per route across various nodes. This offers actionable results for earthquake evacuation but lacks adaptability to environmental changes such as flooding.

Finally, (Wei et al., 2024) applied an enhanced Sparrow Search Algorithm (MSSA) with superior optimization capabilities, achieving a 99% improvement in accuracy and a 15% reduction in computation time. While the MSSA exhibits clear advantages in computational efficiency, our study emphasizes practicality and local implementation by using widely accessible software (WinQSB) and simple heuristics for zoning and route optimization. This ensures feasibility for urban planning in resource-constrained contexts like Barranco.

Conclusions

The meeting points were determined; however, it was noted that they were not sufficient to accommodate the entire population.

Another problem is that many of them are far from the most extreme areas of the district; therefore, it is necessary to consider additional meeting points outside the district, ensuring safe access for the inhabitants

To ensure the safety of the inhabitants, coordination between both municipalities (Barranco and Surco) would be necessary.

This study demonstrated that it is possible to determine optimal evacuation routes for the different areas of the Barranco district using Dijkstra's algorithm implemented in the WinQSB software. The algorithm allowed the calculation of the shortest routes to safe areas, facilitating decision-making by providing optimized paths that consider distances and intermediate nodes. This approach contributes to improving emergency preparedness and can be adapted to similar urban contexts.

Recommendations

To determine the sufficiency of safe areas, it is recommended to first know why an area is safe, this with the help of the corresponding professionals, such as INDECI personnel for example. Subsequently, we must determine the flow of people who will go to these areas and the distance they must travel from their homes.

Considering all the above, it will be possible to quantify and analyze the potential areas, discarding those that do not benefit the population.

Dijkstra optimizes the routes, taking into consideration previous research from the literature review where it is stated that traditional minimum path search algorithms do not show optimal performance when the graph in which the execution is performed is large. For example, Dijkstra's algorithm will always be subject to the size of the graph, which if excessively large will lead to a long response time (Henri et al., 2024).

To determine optimal routes to assembly areas using Dijkstra's algorithm and WinQSB, it is recommended to first understand the problem to be solved. This will help adapt the algorithm to specific local needs.

We then represent the problem as a graph where the nodes are locations and the edges have values that reflect the distance between them.

To use the WinQSB software, it is recommended to familiarize yourself with it to know what options to select when using it, so that when entering the corresponding data, it does not give us unwanted results.

Funding Information

The authors received no financial support for the research, authorship, and/or publication of this article.

Author's Contributions

SHG, HVH: Conceptualization, Formal analysis, Software, Resources, Writing – original draft.
GLEMN, PDC: Methodology, Project administration, Visualization, Writing – review & editing.
SMH, OBP: Investigation, Software, Methodology, Writing – review & editing.
JCD, JLG: Data curation, Validation, Supervision, Writing – review & editing.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all the other authors have read and approved of the manuscript and no ethical issues involved.

This study involved only field surveys of public spaces and non-sensitive interviews with municipal staff. Therefore, formal IRB approval was not required, but all procedures complied with institutional ethical standards.

References

Chen, Y., Shen, S., Chen, T., & Yang, R. (2014). Path Optimization Study for Vehicles Evacuation Based on Dijkstra algorithm. *Procedia Engineering*, *71*, 159–165. https://doi.org/10.1016/j.proeng.2014.04.023

Deng, Y., Chen, Y., Zhang, Y., & Mahadevan, S. (2012). Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Applied Soft Computing Journal*, *12*(3), 1231–1237. https://doi.org/10.1016/j.asoc.2011.11.011

Ebid, A. M., Abdel-Kader, M. Y., Mahdi, I. M., & Abdel-Rasheed, I. (2024). Ant Colony Optimization based algorithm to determine the optimum route for overhead power transmission lines. *Ain Shams Engineering Journal*, *15*(1). https://doi.org/10.1016/j.asej.2023.102344

Henri, M. R. B., Wahab, S. H. A., Fattah, S., Saad, N., Tambrin, N. A. M. B., Jamaluddin, N. F. I. B., Shien, S. O. S., & Ghazalie, Z. S. B. (2024). Utilisation of the Dijkstra Algorithm to Determine the Shortest Path on Universiti Malaysia Sabah's Main Campus in Kota Kinabalu. *6th IEEE International Conference on Artificial Intelligence in*

Engineering and Technology, IICAIET 2024, 500–505. https://doi.org/10.1109/IICAIET62352.2024.10730378

Liu, H., Luo, J., Zhang, L., Yu, H., Liu, X., & Wang, S. (2025). Research on Traversal Path Planning and Collaborative Scheduling for Corn Harvesting and Transportation in Hilly Areas Based on Dijkstra's Algorithm and Improved Harris Hawk Optimization. *Agriculture Switzerland*, *15*(3). https://doi.org/10.3390/agriculture15030233

Liu, T., & Meidani, H. (2025). Graph neural networks for travel distance estimation and route recommendation under probabilistic hazards. *International Journal of Transportation Science and Technology*. https://doi.org/10.1016/j.ijtst.2025.02.006

López-Córdova, F., Vega-Huerta, H., Maquen-Niño, G., Cáceres-Pizarro, J., Adrianzén-Olano, I., & Benito-Pacheco, O. (2025). Construction of a New Data Set of Pleural Fluid Cytological Images for Research. *International Journal of Online and Biomedical Engineering*, *21*(7), 138–151. https://doi.org/10.3991/ijoe.v21i07.54323

Makariye, N. (2017). Towards shortest path computation using Dijkstra algorithm. *2017 International Conference on IoT and Application (ICIOT)*, 1–3. https://doi.org/10.1109/ICIOTA.2017.8073641

Miyombo, M. E., Liu, Y.-K., Mulenga, C. M., Siamulonga, A., Kabanda, M. C., Shaba, P., Xi, C., & Ayodeji, A. (2024). Optimal path planning in a real-world radioactive environment: A comparative study of A-star and Dijkstra algorithms. *Nuclear Engineering and Design*, *420*. https://doi.org/10.1016/j.nucengdes.2024.113039

Pariona-Sánchez, W. S. (2019). Evaluación de un sistema de búsqueda de rutas de evacuación eficientes de un establecimiento usando el algoritmo D estrella (D *). *Innovando La Educación En Tecnología. Actas Del II Congreso Internacional de Ingeniería de Sistemas*, 77–90. https://doi.org/10.26439/ciis2019.5502

Rahman, S. A. F. S. A., Maulud, K. N. A., & Mazlan, M. F. M. (2024). Revolutionizing indoor emergency evacuation prediction with machine learning: A Hand-searching technique and content analysis on Random Forest algorithm. *Journal of Contingencies and Crisis Management*, *32*(1). https://doi.org/10.1111/1468-5973.12502

Rashid, M. M., Rahman, R., & Hasan, S. (2025). Network-Wide Evacuation Traffic Prediction in a Rapidly Intensifying Hurricane from Traffic Detectors and Facebook Movement Data: Deep-Learning Approach. *Journal of Transportation Engineering Part A Systems*, *151*(1). https://doi.org/10.1061/JTEPBS.TEENG-8416

Robledo-Giraldo, S., Duque-Méndez, N. D., & Zuluaga-Giraldo, J. I. (2013). Difusión de productos a través de redes sociales: una revisión bibliográfica utilizando la teoría de grafos. *Respuestas*, *18*(2), 28–42. https://doi.org/10.22463/0122820X.361

Sabri, N. A. M., Basari, A. S. H., Husin, B., & Samah, K. A. F. A. (2015). The utilisation of dijkstra's algorithm to assist evacuation route in higher and close building. *Journal of Computer Science*, *11*(2), 330–336. https://doi.org/10.3844/jcssp.2015.330.336

StalinMary, R., & Rajasingh, I. (2025). Minimum Vertex Cover in Extended Fibonacci Cubes. *Springer Proceedings in Mathematics and Statistics.*, *480*, 81–86. https://doi.org/10.1007/978-3-031-77764-6_10

Torres, M. A., Chávez-Cifuentes, J. F., & Reinoso, E. (2022). A conceptual flood model based on cellular automata for probabilistic risk applications. *Environmental Modelling & Software*, *157*, 105530. https://doi.org/10.1016/j.envsoft.2022.105530

Vega-Huerta, H., Gutierrez-Mejía, F., Calcina-Aguilar, B., Benito-Pacheco, O., De-La-Cruz-Vdv, P., Maquen-Niño, G. L. E., & Lázaro-Guillermo, J. C. (2025). ChatGPT-Based Conversational Artificial Intelligence System for Virtual University Admissions Office Client Attention. *Ingenierie Des Systemes d'Information*, *30*(4), 945–952. https://doi.org/10.18280/isi.300411

Vega-Huerta, H., Rivera-Obregón, M., Maquen-Niño, G. L. E., De-La-Cruz-VdV, P., Lázaro-Guillermo, J. C., Pantoja-Collantes, J., & Cámara-Figueroa, A. (2025). Classification Model of Skin Cancer Using Convolutional Neural Network. *Ingenierie Des Systemes d'Information*, *30*(2), 387–394. https://doi.org/10.18280/isi.300210

Vega-Huerta, H., Sanchez-Moreno, D., De-la-cruz, P., Maquen-nino, G., & Quispe Barra, M. (2024). Development and Usability Evaluation of Health Fit Mobile Application for Physical Training Management During the COVID-19 Pandemic. *BTSym 2023*, *1*, 329–339. https://doi.org/10.1007/978-3-031-66961-3_29

Wei, X., Zhang, Y., & Huaitao, S. (2024). Research on Evacuation Path Planning Based on Improved Sparrow Search Algorithm. *Computer Modeling in Engineering \& Sciences*, *139*(2), 1295–1316. https://doi.org/10.32604/cmes.2023.045096

Zhou, M., & Gao, N. (2019). Research on Optimal Path based on Dijkstra Algorithms. *Advances in Computer Science Research*, *87*(Icmeit), 884–892. https://doi.org/10.2991/icmeit-19.2019.141