An Intelligent Optimazitation Method for Evacuation Route Planning in the Occurrence of Natural Disasters

Hafsah Nirwana
Department of Electrical Engineering, Politeknik Negeri Ujung Pandang, Sulawesi Selatan, Indonesia
hanir@poliupg.ac.id (corresponding author)

Wawan Firgiawan
Department of Engineering, Universitas Sulawesi Barat, Sulawesi Barat, Indonesia
wawanfirgiawan@unsulbar.ac.id (corresponding author)

Sugiarto Cokrowibowo
Department of Engineering, Universitas Sulawesi Barat, Sulawesi Barat, Indonesia
sugiartocokrowibowo@unsulbar.ac.id

Zahir Zainuddin
Department of Engineering, Universitas Hasanuddin, Sulawesi Selatan, Indonesia
zahir.zainuddin@unhas.ac.id

Received: 28 July 2024 | Revised: 27 August 2024 | Accepted: 3 September 2024

 DOI: https://doi.org/10.48084/etasr.8538

ABSTRACT

This research aims to design and apply intelligent optimization methods using various algorithms to find disaster evacuation routes. The efficiency and effectiveness of evacuation routes are essential in disaster situations to ensure the safety of the affected residents. This research focuses on developing an intelligent optimization method utilizing the Multi Vertex Multi Goals (MVMG) scheme to find optimal evacuation routes. In this scheme, multiple starting points and evacuation destinations reflect the actual conditions on the ground. The Ant Colony Optimization (ACO) algorithm was chosen because of its superiority in finding optimal solutions in dynamic and complex conditions. This research also compares the performance of ACO with traditional algorithms, such as Dijkstra and Breadth-First Search (BFS). The test results show that ACO consistently achieves the lowest evacuation time and the highest efficiency compared to the other two algorithms. In addition, this research opens opportunities for further research by considering complex factors, including traffic congestion and disaster-prone areas, to improve the robustness and application of optimization algorithms in more realistic and dynamic scenarios.

Keywords-disaster; evacuation route; Multi-Vertex Multi-Goals (MVMG)

I. INTRODUCTION

Natural disasters, such as earthquakes and tsunamis, often occur suddenly and can cause significant losses in terms of material and human lives. In such an emergency, the efficiency and effectiveness of evacuation routes are critical factors that determine the safety of the residents affected by the disaster. Well-planned evacuation routes can minimize the number of fatalities and ensure rapid evacuation towards safer places. Speed and accuracy in the evacuation process are crucial factors that determine the number of victims, which can be minimized. Poorly planned evacuations can cause confusion, backlogs, and even additional accidents that worsen disaster conditions. Therefore, the importance of fast and safe evacuation underscores the need for optimization methods that can adapt to dynamic and complex field conditions. Many researchers have studied the selection of evacuation routes by conducting disaster case studies. Some of them involve case studies on finding evacuation routes during earthquake and tsunami disasters, which are an attractive discussion topic nowadays. Various methods have been designed to solve the problem of finding evacuation routes. These include searching for evacuation routes with several exits in a building using cellular automata [1]. Other research studies have also utilized route optimization addressing to the route search process, applying various algorithms [2-8]. However, in some cases only one starting point and one destination point are considered. Thus, greater flexibility in handling scenarios like this is required, since the aforementioned cases cannot accommodate the diversity of the evacuation points and changing conditions.

Evacuation routes are designed to be used by people in emergencies so that they will be able to reach a safe place quickly and efficiently. The former are essential for in-depth study. Based on several previous studies, the optimal evacuation route considers the shortest distance and various other factors, such as the traffic density during evacuation, changes in environmental conditions, and the time needed to reach a safe place. However, this research focuses on developing an intelligent optimization method that can be deployed to plan disaster evacuation routes on land by assuming that there are various starting points and several safe areas, which are called goals. An illustration of this problem is outlined in Figure 1. The image depicted contains some information related to the vertex start (illustrated in red) and vertex goals as destination points or safe places (illustrated in blue).

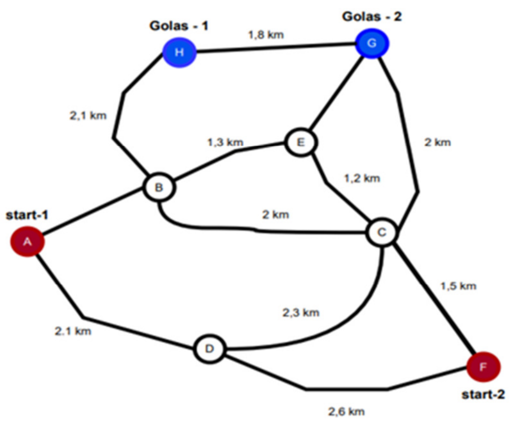

Fig. 1. Illustration of the problem of finding evacuation routes with multi-vertex-goals.

This problem acts as a source of reference in real situations. It refers to the case in which a disaster occurs, with people being in different places and being asked to find a route to the nearest safe place. Traditional algorithms, such as Dijkstra and DFS, along with more modern optimization algorithms, help to optimize the evacuation time to a safer point. The implementation of ACO algorithm in optimizing evacuation routes offers several advantages, making it highly suitable for evacuation planning. These advantages include computational efficiency, adaptability to more complex problems, and the ability to identify evacuation routes that are not only optimal overall, but also efficient at every stage of the journey. Previous research has demonstrated that traditional approaches in designing evacuation routes often need more flexibility and adaptability to dynamic conditions that arise during the evacuation process, since the recommended routes are not optimal [9-11]. Therefore, this research aims to develop an evacuation route optimization method by putting into service an MVMG scheme that is more suitable for dynamic disaster conditions, as well as by comparing the effectiveness of traditional route-finding algorithms, such as Dijkstra and DFS, with modern algorithms, such as ACO, in the context of disaster evacuation routes. The MVMG constitutes the main contribution to this research. Modern algorithms are expected to offer more efficient and adaptive solutions, since they use more sophisticated and heuristic approaches to the evacuation problem, as the ability to learn from the environment in which the scenarios are tested develops. The results of this research can significantly contribute to the field of emergency management and public safety and provide guidance for policymakers and practitioners in designing more effective and efficient evacuation routes.

II. RELATED WORK

Research related to finding evacuation routes has been carried out by various researchers employing numerous approaches and methods [12-14]. Authors in [15] explored the search for evacuation routes during dynamically changing dangerous conditions. Authors in [16] introduced a model by optimizing bus schedules and route planning for evacuation, intending to reduce the total evacuation time through simultaneous optimization of the bus capacity, refugee demand, time window, and flow balance. Authors in [17] developed a method for allocating evacuation routes and shelters to reduce the number of fatalities by considering various tsunami scenarios. This model allocates optimal routes and shelters based on a combination that provides a safe evacuation route. Authors in [18] developed the Time-Varying Equivalent Weight Dijkstra Algorithm (TVEWDA), which combines dynamic fire simulation to find optimal evacuation routes in fire scenarios taking place on cruise ships. Authors in [1] developed an extended cellular automata model for pedestrian emergency evacuation dynamics. This model introduces group route and terrain change probabilities, allowing pedestrians to change direction as needed and stay together in groups. Authors in [19] introduced the Cellular Automata-based Dynamic Route Optimization (CADRO) algorithm, which identifies dynamic flood evacuation routes by considering hydrodynamics, topography, and human response time, showing significant improvements in evacuation route efficiency compared with the traditional A* algorithm. In 2023, Authors in [20] used the Dijkstra algorithm to determine the shortest evacuation route in Benoa Village, a tsunami-prone area. This research produces a route from the evacuation starting point to the safe zone, where the Dijkstra algorithm is appluied to process the adjacency matrix, which describes the road network and intersections.

Authors in [21] proposed an improved ACO algorithm version to determine the optimal evacuation route in a fire scenario in a supermarket building by considering temperature conditions and burning products. ACO is a widely utilized algorithm in various route search cases, while according to several studies, ACO is a good algorithm to use in dynamic conditions [22-24]. In this research, the use of ACO is the main method for finding evacuation routes by considering vertex-start positions, which are spread across many points and have many vertex-goals as destinations. This case resembles the concept of Multi Depot Multi Traveling Salesman Problem (MDMTSP), where there are many depots (departure points) carried out by a salesman who is going to make sales in several planned cities [25-27]. However, the difference in the concept provided in the present study is that determining the evacuation route refers to the condition in which people will go to the vertex-goals closest to them when a disaster occurs.

III. METHOD

A. *Mutlti Vertex Multi Goals (MVMG)*

The Multi Vertex Multi Goals (MVMG) scheme is an approach used to overcome evacuation problems where there are many starting points (vertex-start) and many safe destinations (vertex-goals). This scheme reflects a situation where residents are spread across various locations and must be evacuated to several predetermined safe points. In the context of disaster evacuation, MVMG allows for modeling that is more realistic and adaptive to dynamic conditions in the field. MVMG provides greater flexibility in planning evacuation routes than traditional methods, which often only consider one starting point and one destination point. By considering multiple starting points and destination points, this scheme can create evacuation plans that are more responsive and in line with the real needs of the field. For example, in a disaster, such as an earthquake or a tsunami, residents may be scattered across various locations and must be transferred to the available safe places.

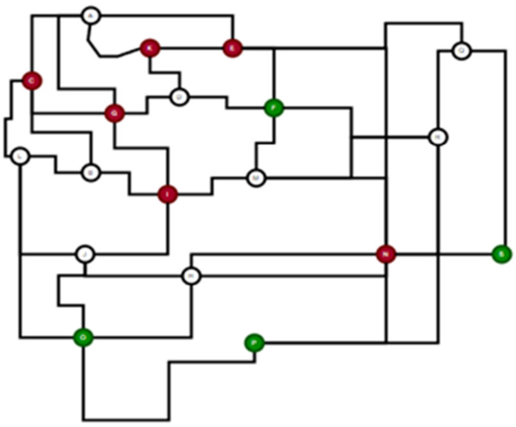

Fig. 2. An MVMG schematic that depicts several starting points (vertex-start) and several safe goals (vertex-goals) connected by various paths. This scheme shows the complexity and adaptability of the MVMG approach in disaster situations.

MVMG enables more efficient organization of the evacuation routes by directing residents from multiple starting points to multiple safe destinations simultaneously. The MVMG approach also considers essential aspects, such as the population density at the starting point, the capacity at the destination point, and the condition of evacuation routes. Thus, the resulting evacuation plan is effective in terms of travel distance and considers evacuees' safety and comfort. For example, if a destination point is too wide, the MVMG scheme can direct some evacuees to alternative points with sufficient capacity. Implementing MVMG in evacuation route planning also involves using intelligent algorithms such as ACO. This algorithm helps find optimal evacuation routes by considering various dynamic variables in the field, involving changes in environmental conditions, obstacles, and population distribution. With this approach, evacuation plans can be updated in real time based on the latest information, ensuring that the chosen route is always safe and efficient. Overall, the MVMG scheme offers a more comprehensive and adaptive disaster evacuation route planning solution. It combines flexibility in determining the starting and destination points, while using smart technology to optimize the route. MVMG can significantly increase the effectiveness and efficiency of the evacuation process, resulting in more lives being saved while reducing risks during disasters.

B. *Ant Colony Optimization (ACO) Implementation*

ACO is a technique for finding optimal evacuation routes in the MVMG scheme. ACO is inspired by the foraging behavior of ants [28, 29], which leave pheromone trails that guide other ants to find the best route.

1) *Problem Formulation*

This problem is formulated to find optimal evacuation routes from several starting points (vertex-start) to several safe points (vertex-goals). The main goal is to minimize the total evacuation time and ensure the safety of evacuees by avoiding dangerous areas. A graph represents an evacuation area, where nodes represent the start, midpoint, and destination, while edges represent possible paths between the nodes.

2) *Dataset and Matrix Adjacency*

The dataset for this graph is created using an adjacency matrix, which describes the connectivity between nodes. An adjacency matrix is a square matrix, where the entry aij is 1 if there is an edge from node i to node j, and 0 if there is no edge. Meanwhile, the weight or distance between the nodes is the distance that will be filled in or be left in the same position. Figure 3 displays an adjacency matrix filled with distance data between the nodes, which represents a graph for the MVMG model. Columns and rows in the matrix are labelled with the nodes in the graph (A, B, C, ..., S). Each cell in the matrix shows the distance between two nodes corresponding to that column and row. For example, a cell in row B and column J, having a value of 50, indicates a distance of 50 between nodes B and J. The green nodes (G-C, I-G, J-M and P-H) are the starting vertices, namely the points from which evacuation begins. The red nodes (C-B, D-F, B-J, F-O, L-C, and N-P) are the vertex goals, namely the safe destination points that must be reached during evacuation. Examples of the distances between the nodes are from C to B with a distance of 15, P to H with a distance of 20, and F to O with a distance of 5. In the context of evacuation using ACO, ants will start from the initial vertex (green) and look for the optimal path to the vertex goals (red). The distance between the nodes will influence the path chosen to minimize the total distance or travel time while considering safety and efficiency.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
A	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0
B	0	0	15	20	0	0	0	0	25	30	0	0	35	0	0	0	0	0	0
C	0	15	0	10	0	0	25	0	0	0	0	20	0	0	0	0	0	0	0
D	0	20	10	0	5	30	0	0	20	0	10	0	40	0	0	0	15	0	0
E	0	0	0	5	0	10	0	0	0	0	0	0	0	0	0	0	0	25	0
F	0	0	0	30	10	0	20	0	0	0	0	0	0	0	5	0	0	0	15
G	0	0	25	0	0	20	0	0	15	0	25	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	10	20	0	0	30	0	0	0	0	0	0
I	0	25	0	20	0	0	15	10	0	0	0	0	0	5	0	0	0	0	0
J	0	30	0	0	0	0	0	20	0	0	0	0	35	0	10	15	0	0	0
K	10	0	0	10	0	0	25	0	0	0	0	0	0	0	0	0	20	0	0
L	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	35	0	40	0	0	0	30	0	35	0	0	0	10	0	0	0	0	0
N	0	0	0	0	0	0	0	0	5	0	0	0	10	0	20	15	0	0	30
O	0	0	0	0	0	5	0	0	0	10	0	0	0	20	0	25	0	0	0
P	0	0	0	0	0	0	0	20	0	15	0	0	0	15	25	0	0	0	0
Q	0	0	0	15	0	0	20	0	0	0	20	0	0	0	0	0	0	10	0
R	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	10	0	20
S	0	0	0	0	0	15	0	0	0	0	0	0	0	30	0	0	0	20	0

Fig. 3. Adjacency matrix.

3) *Ant Movement*

- Ant Placement**:** Ants are placed at the starting point (vertex-start).
- Path Selection**:** Ants probabilistically select paths, considering pheromone levels and heuristic information. The probability Pij of an ant moving from node i to node j is formulated as:

$$P_{i,j}(t) = \frac{[T_{i,j}(t)]a\,.[n_{i,j}(t)]b}{\sum_{k\,\epsilon\,allow}[T_{i,k}(t)]a\,.[n_{i,k}(t)]b} \quad (1)$$

where $T_{i,j}$ is the pheromone level at the edge (i, j), $n_{i,j}$ is the heuristic information (e.g., the inverse of the distance), a and b are the parameters that control the influence of the pheromone and heuristic information, and Ni is the set of neighboring nodes of i.

4) *Pheromon Update*

- Local Update: When ants traverse the graph, they leave pheromones on the edges they traverse.
- Global Update: Once all ants have completed their route, the pheromone level at the edge is updated. Edges that are part of the best route receive more pheromones, while others experience pheromone evaporation. The pheromone renewal rules are:

$$T_{ij} = (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij} \quad (2)$$

where ρ is the evaporation rate, and $\Delta\tau_{ij}$ is the amount of the pheromone deposited.

5) *Termination*

Termination in the ACO algorithm is the stage where the algorithm execution is stopped based on specific criteria. There are two main termination methods commonly used in ACO [30]:

- Iteration Set: In this method, the algorithm runs for a predefined number of iterations. For example, an algorithm might be set to run for one thousand iterations. Once the last iteration is reached, the algorithm will stop, and the best route found during that iteration will be considered the optimal evacuation route. The advantages of this method are simplicity and ease of implementation, but the weakness is that it only sometimes guarantees that the optimal solution has been found.
- Convergence Criteria: In this method, the algorithm stops when the resulting solution has reached stability or convergence. The convergence criterion can be a minimal change in the quality of the best solution over several consecutive iterations. For example, if there is no significant improvement in the quality of the best route for fifty consecutive iterations, the algorithm is considered to have reached convergence and is terminated. The advantage of this method is that it ensures that the algorithm stops when there is no significant further improvement. However, the disadvantage is that it may require more iterations and computing time.

By evaluating these two termination methods, the best approach can be found, providing a balance between computational efficiency and solution quality in the context of evacuation route optimization utilizing ACO.

6) *Simulation and Results*

The ACO algorithm was implemented and tested in an evacuation simulation scenario. The results are compared with traditional algorithms, such as Dijkstra and DFS, to evaluate their performance. The key performance metrics deployed include the evacuation distance and efficiency. The simulation settings are:

- Graph Construction: The graph represents the evacuation area that is created with nodes and edges based on the layout. These nodes include start points (vertex-start), goal points (vertex-goals), and possible paths between them. The image has been converted into an adjacency matrix containing the distance between nodes.
- Total Evacuation Time: It measures the time required for evacuation from the start to the destination point.
- Parameter Settings: The ACO algorithm parameters, such as the number of ants, pheromone evaporation rate, and influence factor (α for pheromone and β for heuristic information), are adjusted to achieve optimal performance. Proper parameter settings ensure that the algorithm can explore multiple paths effectively and converge towards an optimal solution.
- Performance Evaluation: The performance of the ACO algorithm is measured and compared with the Dijkstra and DFS algorithms. Evaluation includes: (a) the evacuation distance that measures the total distance travelled from the starting point to the destination point, and (b) the efficiency that measures the computing time and resources used to find a solution.

7) *Pseudocode MVMG-ACO*

Pseudocode for MVMG utilizing ACO.

```
INITIALIZATION:
- S = {set of starting vertices}
- G = {set of goal vertices}
```

```
- τ = pheromone matrix (initial pheromone
values)
- D = distance matrix (distances between
vertices)
- n_ants = number of ants
- n_iterations = number of iterations
- ρ = evaporation rate
- α = influence of pheromone
- β = influence of distance
For each iteration from 1 to n_iterations:
1. For each ant from 1 to n_ants:
      a     Randomly select a starting
vertex (v) from S
      b     Initialize path (P) with the
starting vertex (v)
      c     Calculate the likelihood of
moving to each neighboring point (u) that
hasn't been visited yet
            i     Calculate the
probability of moving to each unvisited
neighboring vertex (u)
            ii    Select the next vertex
(u) based on the probabilities norm_p(u)
            iii   Move to the next vertex
(u), add u to the path (P)
2. Update pheromone:
      Evaporate pheromone on all edges:
      τ(v, u) = (1 - ρ) * τ(v, u)
      Add new pheromone based on the paths
found by the ant. For each edge (v, u) in
the path (P):
      τ(v, u) = τ(v, u) + Q / length(P) (Q
is a constant)
RETURN THE BEST PATH FOUND AND ITS LENGTH
```

IV. EXPERIMENTS AND RESULTS

A. Testing Result

For the testing needs, 20 datasets were considered along with three algorithms: ACO, Dijkstra, and BFS. Each dataset was tested using these three algorithms. The test results were compared, as shown in Figure 4. Figure 4 portrays the test results of three algorithms (ACO, Dijkstra, and BFS) against 20 different datasets. This graph displays each algorithm's total evacuation time for each dataset tested. It demonstrates that ACO is the most effective and efficient algorithm in optimizing evacuation time compared to BFS and Dijkstra. BFS consistently has the highest evacuation time, while Dijkstra performs better but falls short of ACO. Therefore, ACO should be the selected algorithm for scenarios requiring fast and efficient evacuation.

B. Efficiency

1) Identification of the Worst Time and Optimal Time

- Worst Time: Highest evacuation time among all algorithms for each dataset.
- Optimal Time: Lowest evacuation time among all algorithms for each dataset.
- Time of Algorithm: Evacuation time required for each algorithm.

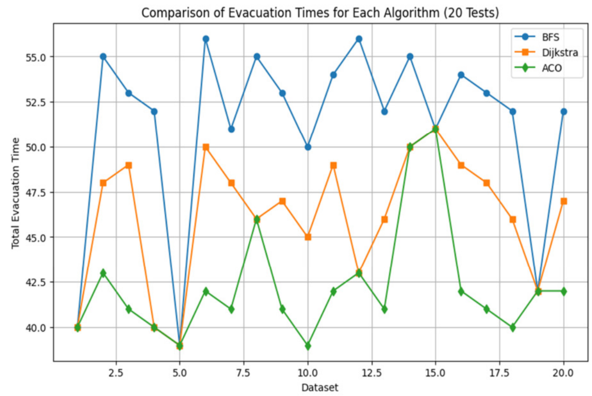

Fig. 4. Comparison of evacuation times for each algorithm.

2) Calculation of the Efficiency for Each Algorithm in the Dataset

To calculate the efficiency of each algorithm, the efficiency formula previously explained will be used. The steps followed are:

$$efficiency = \left(\frac{\text{Worst Time} - \text{Time Algorithm}}{\text{Time Algorithm}}\right) \times 100\% \quad (3)$$

Table I compares the performance of the three algorithms, BFS, Dijkstra, and ACO, on three different datasets. The total evacuation time and efficiency calculation for each algorithm on each dataset are also displayed.

TABLE I. COMPARISON OF TESTING RESULTS

Datasets	Algorithm	Total evacuation time	Efficiency (%)
1	BFS	50	0.00
1	Dijkstra	45	11.11
1	ACO	40	25.00
2	BFS	55	0.00
2	Dijkstra	48	21.21
2	ACO	43	66.67
3	BFS	53	0.00
3	Dijkstra	49	9.76
3	ACO	41	29.27

As evidenced in Figure 5, and particularly in dataset 1, it can be concluded that ACO consistently shows higher efficiency than BFS and Dijkstra. In each dataset, BFS shows the lowest efficiency with a value of 0%, indicating that this algorithm requires the longest evacuation time. Dijkstra performs better than BFS with a higher average efficiency but still lags behind ACO. ACO recorded the highest efficiency in each dataset, with an achievement of 25% in Dataset 1, 66.67% in Dataset 2, and 29.27% in Dataset 3, indicating that ACO can optimize evacuation routes more effectively. These results show that ACO is the most efficient algorithm for evacuation route optimization, followed by Dijkstra, while BFS shows the lowest performance. In Dataset 1 (Figure 5), BFS shows the highest evacuation time with a total efficiency time of 0%, confirming its inefficiency in finding the optimal evacuation path. Dijkstra maintains moderate performance with a total efficiency evacuation time of 11.11%, which is better than BFS but still lower than ACO. ACO is again the most efficient, with a total efficiency evacuation time of 25%, indicating its ability to optimize evacuation paths. In dataset 2 (Figure 6), BFS still shows the same inefficiency with a total evacuation time efficiency of 0%, making it the algorithm with the lowest performance. Dijkstra records an increase in efficiency with a total evacuation time of 21.21%, better than its performance on Dataset 1, but still inferior to ACO. ACO once again shows the highest efficiency with a total evacuation time of 66.67%, confirming its dominance in finding the most optimal evacuation path. In dataset 3 (Figure 7), BFS again shows the worst performance, communicating it to produce efficient distribution paths. Dijkstra's efficiency slightly decreases compared to Dataset 2, but is still better than BFS's and still far from ACO. Even though ACO has lower efficiency than that in Dataset 2, it is still the most efficient algorithm.

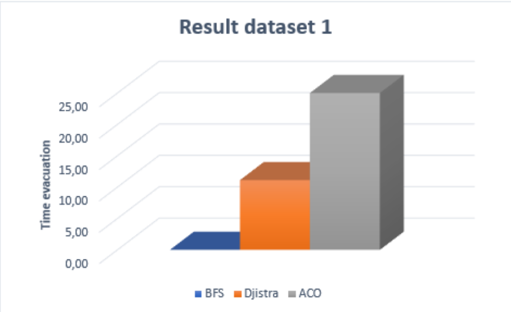

Fig. 5. Comparison of BFS, Dijkstra, and ACO evacuation times on dataset 1.

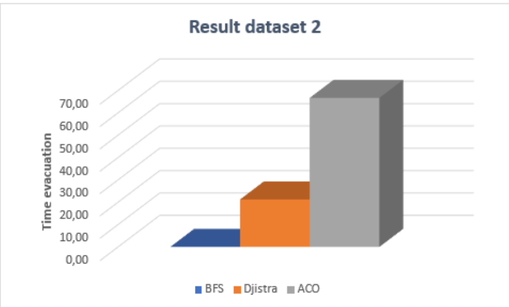

Fig. 6. Comparison of BFS, Dijkstra, and ACO evacuation times on dataset 2.

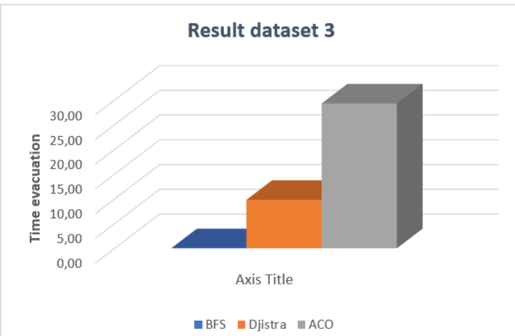

Fig. 7. Comparison of BFS, Dijkstra, and ACO evacuation times on dataset 3.

As shown in Table II, the BFS has consistently the highest evacuation times among all datasets, resulting in 0% efficiency each time, indicating that it is not suitable for scenarios requiring fast evacuation. Dijkstra shows moderate performance with 14.03% efficiency on all datasets. Although better than BFS, this algorithm still falls short compared to ACO, indicating that there are more optimal solutions for this scenario despite the observed improvements. In contrast, ACO consistently achieved the highest efficiency and the lowest evacuation time across all datasets. This suggests that ACO is very effective and efficient for optimizing evacuation routes in dynamic and complex scenarios.

TABLE II. AVERAGE EFFICIENCY RESULTS

Algorithm	Average efficiency (%)
BFS	0.00
Dijkstra	14.03
ACO	40.31

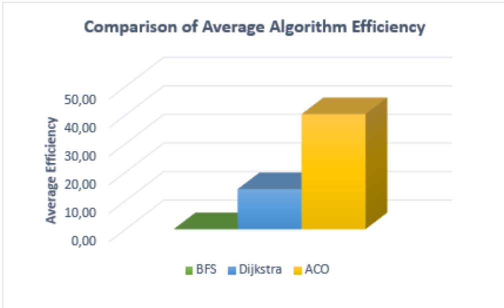

Fig. 8. Average efficiency BFS, Dijkstra, and ACO.

From this analysis, it is noted that ACO is the best algorithm for evacuation route optimization. This algorithm significantly outperforms BFS and Dijkstra, providing the lowest evacuation time and highest efficiency in all tested datasets. Therefore, ACO should be the algorithm chosen for scenarios requiring fast and efficient evacuation.

V. CONCLUSIONS

This research is driven by the urgent need to develop an efficient and effective evacuation route planning method in disaster scenarios, where fast and accurate decision-making is critical to save lives. Recognizing the complexity of real-world evacuation scenarios, which often involve multiple origins and destinations, an intelligent optimization method is developed using the MVMG scheme. This approach closely reflects the complex reality of evacuation planning. The ACO algorithm, which is chosen for its strength in navigating dynamic and complex conditions, is deployed. The test results of the ACO against traditional algorithms, such as Dijkstra and BFS, on 20 datasets reveal its clear superiority. ACO consistently achieves the lowest evacuation time and highest efficiency, significantly outperforming BFS and Dijkstra. These findings confirm that ACO is more effective in optimizing evacuation routes and better suited to real-world disaster scenarios' dynamic and complex conditions. The novelty of this study lies in the application of the MVMG scheme combined with the ACO, which offers a more accurate and practical approach to disaster management compared to the traditional single-source, singledestination methods. Future research should build on these findings by incorporating factors such as traffic congestion and disaster-prone areas, further improving the robustness of the optimization algorithm under multi-node and multi-objective conditions.

ACKNOWLEDGMENT

This research was supported by Politeknik Negeri Ujung Pandang, Sulawesi Barat University, and Hasanuddin University. The authors extend their gratitude to these institutions for their invaluable support and contributions to the current study.

REFERENCES

[1] L. A. Pereira, D. Burgarelli, L. H. Duczmal, and F. R. B. Cruz, "Emergency evacuation models based on cellular automata with route changes and group fields," *Physica A: Statistical Mechanics and its Applications*, vol. 473, pp. 97–110, May 2017, https://doi.org/10.1016/j.physa.2017.01.048.

[2] P. C. Iloh, "A Comprehensive and Comparative Study of DFS, BFS, and A* Search Algorithms in a Solving the Maze Transversal Problem," *International Journal of Social Sciences and Scientific Studies*, vol. 2, no. 2, pp. 482–490, Apr. 2022.

[3] A. Alyasin, E. I. Abbas, and S. D. Hasan, "An Efficient Optimal Path Finding for Mobile Robot Based on Dijkstra Method," in *2019 4th Scientific International Conference Najaf (SICN)*, Apr. 2019, pp. 11–14, https://doi.org/10.1109/SICN47020.2019.9019345.

[4] M. Abdul-Niby, M. Alameen, A. Salhieh, and A. Radhi, "Improved Genetic and Simulating Annealing Algorithms to Solve the Traveling Salesman Problem Using Constraint Programming," *Engineering, Technology & Applied Science Research*, vol. 6, no. 2, pp. 927–930, Apr. 2016, https://doi.org/10.48084/etasr.627.

[5] H. Jafarzadeh, N. Moradinasab, and M. Elyasi, "An Enhanced Genetic Algorithm for the Generalized Traveling Salesman Problem," *Engineering, Technology & Applied Science Research*, vol. 7, no. 6, pp. 2260–2265, Dec. 2017, https://doi.org/10.48084/etasr.1570.

[6] M. Kammoun, H. Derbel, M. Ratli, and B. Jarboui, "A variable neighborhood search for solving the multi-vehicle covering tour problem," *Electronic Notes in Discrete Mathematics*, vol. 47, pp. 285–292, Feb. 2015, https://doi.org/10.1016/j.endm.2014.11.037.

[7] R. D. Prayogo and S. A. Karimah, "Numerical Simulation of Tsunami Evacuation Route Planning in Padang Using Linear Programming," in *2020 8th International Conference on Information and Communication Technology (ICoICT)*, Jun. 2020, pp. 1–6, https://doi.org/10.1109/ICoICT49345.2020.9166266.

[8] W. wirgiawan, A. A. A. Cirua, Akbar, and S. Cokrowibowo, "Perbandingan Waktu Komputasi Algoritma Greedy-Backtracking, BFS, DFS, dan Genetika pada Masalah Penukaran Koin," *Konferensi Nasional Ilmu Komputer (KONIK)*, vol. 5, no. 1, pp. 5–8, 2021.

[9] K. Dong, D. Yang, J. Sheng, W. Zhang, and P. Jing, "Dynamic planning method of evacuation route in dam-break flood scenario based on the ACO-GA hybrid algorithm," *International Journal of Disaster Risk Reduction*, vol. 100, Jan. 2024, Art. no. 104219, https://doi.org/10.1016/j.ijdrr.2023.104219.

[10] M. Hai-bo and S. Yu-bo, "Study on evacuation routes under unexpected event based on Petri net," in *2017 4th International Conference on Transportation Information and Safety (ICTIS)*, Dec. 2017, pp. 773–778, https://doi.org/10.1109/ICTIS.2017.8047855.

[11] N. Wang, T. Wang, Z. Zhang, S. Shao, Y. Jiang, and Y. Li, "Mobile Robot Path Planning Method Based on an Improved A* Algorithm," in *2022 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, Jul. 2022, pp. 140–143, https://doi.org/10.1109/CYBER55403.2022.9907135.

[12] N. A. Hasanah, L. Atikah, D. Herumurti, and A. A. Yunanto, "A Comparative Study: Ant Colony Optimization Algorithm and Backtracking Algorithm for Sudoku Game," in *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Sep. 2020, pp. 548–553, https://doi.org/10.1109/iSemantic50169.2020.9234267.

[13] M. Aleksandrov, C. Cheng, A. Rajabifard, and M. Kalantari, "Modelling and finding optimal evacuation strategy for tall buildings," *Safety Science*, vol. 115, pp. 247–255, Jun. 2019, https://doi.org/10.1016/j.ssci.2019.02.017.

[14] Y. Akizuki, "Evacuation route design based on visibility for reducing evacuation delays," *Fire Safety Journal*, vol. 144, Mar. 2024, Art. no. 104099, https://doi.org/10.1016/j.firesaf.2024.104099.

[15] S. Arias, S. La Mendola, J. Wahlqvist, O. Rios, D. Nilsson, and E. Ronchi, "Virtual Reality Evacuation Experiments on Way-Finding Systems for the Future Circular Collider," *Fire Technology*, vol. 55, no. 6, pp. 2319–2340, Nov. 2019, https://doi.org/10.1007/s10694-019-00868-y.

[16] M. Lujak and S. Giordani, "Centrality measures for evacuation: Finding agile evacuation routes," *Future Generation Computer Systems*, vol. 83, pp. 401–412, Jun. 2018, https://doi.org/10.1016/j.future.2017.05.014.

[17] X. Zhao, K. Ji, P. Xu, W. Qian, G. Ren, and X. Shan, "A round-trip bus evacuation model with scheduling and routing planning," *Transportation Research Part A: Policy and Practice*, vol. 137, pp. 285–300, Jul. 2020, https://doi.org/10.1016/j.tra.2020.05.008.

[18] F. Kitamura, D. Inazu, T. Ikeya, and A. Okayasu, "An allocating method of tsunami evacuation routes and refuges for minimizing expected casualties," *International Journal of Disaster Risk Reduction*, vol. 45, May 2020, Art. no. 101519, https://doi.org/10.1016/j.ijdrr.2020.101519.

[19] L. Liu, H. Zhang, J. Shi, and J. Geng, "Real-time evacuation route optimization in the fire scenarios of cruise ships," *Simulation Modelling Practice and Theory*, vol. 129, Dec. 2023, Art. no. 102843, https://doi.org/10.1016/j.simpat.2023.102843.

[20] M. He *et al.*, "An efficient dynamic route optimization for urban flooding evacuation based on Cellular Automata," *Computers, Environment and Urban Systems*, vol. 87, May 2021, Art. no. 101622, https://doi.org/10.1016/j.compenvurbsys.2021.101622.

[21] I. B. K. P. A. K, W. Sukartiasih, and A. Sedayu, "Optimization of the Shortest Tsunami Evacuation Route Using Djikstra's Algorithm in Benoa Village," *Brilliance: Research of Artificial Intelligence*, vol. 3, no. 2, pp. 217–224, Jun. 2023, https://doi.org/10.47709/brilliance.v3i2.3089.

[22] L. Xu, K. Huang, J. Liu, D. Li, and Y. F. Chen, "Intelligent planning of fire evacuation routes using an improved ant colony optimization algorithm," *Journal of Building Engineering*, vol. 61, Dec. 2022, Art. no. 105208, https://doi.org/10.1016/j.jobe.2022.105208.

[23] S. T. Ng and Y. Zhang, "Optimizing Construction Time and Cost Using Ant Colony Optimization Approach," *Journal of Construction Engineering and Management*, vol. 134, no. 9, pp. 721–728, Sep. 2008, https://doi.org/10.1061/(ASCE)0733-9364(2008)134:9(721).

[24] M. Guntsch, M. Middendorf, and H. Schmeck, "An Ant Colony Optimization approach to dynamic TSP," in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, San Francisco, CA, USA, Apr. 2001, pp. 860–867.

[25] L. Bianchi, L. M. Gambardella, and M. Dorigo, "An Ant Colony Optimization Approach to the Probabilistic Traveling Salesman Problem," in *Parallel Problem Solving from Nature — PPSN VII*, Berlin, Heidelberg, 2002, pp. 883–892, https://doi.org/10.1007/3-540-45712-7_85.

[26] L. Zhen, C. Ma, K. Wang, L. Xiao, and W. Zhang, "Multi-depot multi-trip vehicle routing problem with time windows and release dates," *Transportation Research Part E: Logistics and Transportation Review*, vol. 135, Mar. 2020, Art. no. 101866, https://doi.org/10.1016/j.tre.2020.101866.

[27] A. Soriano, M. Gansterer, and R. F. Hartl, "The multi-depot vehicle routing problem with profit fairness," *International Journal of Production Economics*, vol. 255, Jan. 2023, Art. no. 108669, https://doi.org/10.1016/j.ijpe.2022.108669.

[28] H. Bae and I. Moon, "Multi-depot vehicle routing problem with time windows considering delivery and installation vehicles," *Applied*

Mathematical Modelling, vol. 40, no. 13, pp. 6536–6549, Jul. 2016, https://doi.org/10.1016/j.apm.2016.01.059.

[29] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *Biosystems*, vol. 43, no. 2, pp. 73–81, Jul. 1997, https://doi.org/10.1016/S0303-2647(97)01708-5.

[30] Y. Wang and Z. Han, "Ant colony optimization for traveling salesman problem based on parameters optimization," *Applied Soft Computing*, vol. 107, Aug. 2021, Art. no. 107439, https://doi.org/10.1016/j.asoc.2021.107439.

[31] Q. Lv and X. Xia, "Towards Termination Criteria of Ant Colony Optimization," in *Third International Conference on Natural Computation (ICNC 2007)*, Dec. 2007, vol. 5, pp. 276–282, https://doi.org/10.1109/ICNC.2007.782.