IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⵋ Peer-reviewed & Refereed journal ⵋ Vol. 13, Issue 6, June 2024

DOI: 10.17148/IJARCCE.2024.13602

A Novel Approach using ASTAR algorithm for finding the shortest path to reach Rescue Supply Locations during the Natural and Military Disasters

Mr. Le Thanh Minh[1], Ms. Sharmila Mathivanan[2]

UG Scholar in Computing, FPT Greenwich University, Ho Chi Minh city, Viet Nam[1]

Lecturer in Computing, FPT Greenwich University, Ho Chi Minh city, Viet Nam[2]

Abstract: In recent years, there has been a frequent and intense occurrence of both Natural Calamities and Military disasters due to National Crisis. The matter of guiding and directing individuals to locations that offer necessary provisions is a significant one. However, map applications are unable to update information consistently and promptly regarding alterations in infrastructure, and roadways caused by natural catastrophes or modifications in humanitarian corridors during times of war. Various technological solutions have been implemented, including the utilization of communication channels. Hence, to find a more prominent solution for making an efficient route from the user's present location to the emergency relief Centre destination, a novel approach using ASTAR algorithm along with the satellite maps has been implemented to solve the problem in hand. Further in this paper, strategies for determining the shortest path, utilizing the ASTAR algorithm for pathfinding, outlining the necessary data preparation process, and evaluating the performance of algorithm in specific scenarios were experimented.

Keywords: Heuristic algorithms, A STAR algorithm, Premature Convergence, Computational intelligence.

I. INTRODUCTION

Over the past few decades, Maps are the most popular and indispensable applications for efficient routing and tracking the locations. Most of the smart technologies are embedded with Map apps for the benefits of users in all the perspectives. In certain circumstances, these Map applications are not highly useful to fulfill the emergency needs. For example, during the Ukrainian conflict, information regarding the humanitarian safety corridor or its alterations was communicated formation regarding the humanitarian safety corridor, or its alterations was communicated formation regarding the humanitarian safety corridor, or its alterations was communicated is the utilisation of signs, which is only feasible in densely populated regions or major thoroughfares. This approach necessitates the implementation of measures to ensure that routes stay unaltered in the face of natural catastrophes. This method necessitates the establishment of measures to ensure that routes remain intact in the face of natural catastrophes. Failure to do so may result in ineffective signage, making it difficult to retrieve information and leading to disagreements in directions. However, this approach may not be feasible when it comes to determining the most efficient method. This essay will mostly focus on utilizing the A* algorithm to discover the most efficient routes to relief spots, as it is a faster and more reliable method. The algorithm employed in constructing routes for robot systems.

In this work, the paper introduces the A* algorithm and then show how it can be applied to find the shortest path to reuse supply locations during natural disasters or war by transforming the map into a grid map and demo the A* algorithm on it.

II. LITERATURE REVIEW

The problem of finding the shortest path can be solved using many different algorithms.

A. Floyd-Warshall algorithm

The easy and simple to use is the Floyd-Warshall algorithm. The Floyd-Warshall algorithm exemplifies the utilisation of dynamic programming to address the challenge of finding the shortest route [1] [2] . This approach resolves the issue by examining solutions derived from prior solutions that are interconnected and permits the existence of several solutions [1] [3]. This approach is utilized to determine the shortest path between every pair of nodes in a graph. The $(n+1)$

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⪻ Peer-reviewed & Refereed journal ⪻ Vol. 13, Issue 6, June 2024
DOI: 10.17148/IJARCCE.2024.13602

matrices $S(m), m = 1,2, \dots, n+1$, each with size $n \times n$, are created by this procedure. If **TRUE** was returned, the shortest path from vertex X to vertex Y is $S'xy$. There is a negative cycle in the graph if **FALSE** is returned.
The algorithm's time complexity is $O(Y^3)$ in directed graph $G(Y, E)$.
Recursive solution to the all-Pair shortest-Paths problem is:

$$S_{ij}^{(m)} = \min \begin{cases} w_{ij} & m = 0, \\ (S_{ij}^{(m-1)}, S_{ik}^{(m-1)} + S_{kj}^{(m-1)}) & m \leq 1. \end{cases}$$

Fig .1 Recursive solution

B. Bellman-Ford algorithm

The Bellman-Ford algorithm is the second technique used to tackle the problem of finding the shortest path [4]. The Bellman-Ford algorithm is a versatile approach that may be employed to determine the shortest path in graphs, regardless of whether they are weighted or unweighted. The algorithm can solve the shortest path-finding issue if the network has negative edges and negative cycles.

This approach has an $O(Y * E)$ computing complexity, where Y represents the graph's vertices and E its edges [5]. The Bellman-Ford algorithm is shown bellow:

- ***Step 1***: *It is the pre-preparation step, the algorithm set the distance to the source vertex as 0, and the distance to all other vertices as infinity.*
- ***Step 2***: *Iterate through all edges in the graph $|E| - 1$ times, where $|E|$ is the number of edges. For each edge (u, v) with weight w, check if the distance from the source vertex to v can be reduced by going through u. If so, update the distance to v with the new, shorter distance.*
- ***Step 3***: *The algorithm checks for negative weight cycles. If there is a negative weight cycle in the graph, the algorithm will not converge and will continue to reduce the distance to some vertices with each iteration. To detect such cycles, Bellman-Ford algorithm repeat step 2 one more time. If any distance is updated in this extra iteration, there must be a negative weight cycle in the graph.*
- ***Step 4***: *If there is no negative weight cycle, now the algorithm determined the shortest distance from the source vertex to each vertex.*

C. Jonhson algorithm

The algorithm can find all pair shortest paths in $O(Y^2 logY + Y * E)$ time by using Johnson's technique [6] [7]. Both the Dijkstra and Bellman-Ford subroutines are used in Johnson's algorithm. Every vertex can be considered the source if Dijkstra's Single Source shortest path algorithm is applied, and this will allow us to identify all pair shortest pathways in $O(Y * YLogY)$ time.

The Johnson algorithm can be described step-by-step below with the given graph G.

- ***Step 1:*** *The algorithm adds a new vertex s to the graph, then creates an edge to connect vertex s with all the vertices in the graph, and the new edge weights are 0. Because the weight of all new edges is 0, so that it can't affect any existing shortest path between the original vertices of the graph. Let's denote the modified graph be G'.*
- ***Step 2:*** *Apply the Bellman-Ford algorithm on the graph G' using vertex s as the starting point. The distance determined by the Bellman-Ford algorithm is represented by the array h_0, h_1, h_2, h_3 ,..., h_{V-1}. If the algorithm find the cycle which has the negative weight, it will return. It is important to understand that a negative weight cycle cannot be formed by a new vertex s since there are no edges connected to s. All edges originate from the vertex s.*
- ***Step 3:*** *Recalculate the weights of the edges in the original graph. Assign the new weight for each edge (u, v) as the sum of the old weight and the difference between h_u and h_v.*
- ***Step 4:*** *Eliminate the newly inserted vertex s and execute Dijkstra's method for each vertex.*

D. Ant Colony Optimization algorithm (ACO)

ACO is a heuristic algorithm that draws inspiration from the foraging behaviour of ants to determine optimal paths in search of food. The use of parallelism not only enhances the stability of the algorithm, but also empowers it with robust global search capabilities. The algorithm is a meta-heuristic method that determines the ideal path for swarm agents in a partially understood environment. In addition to addressing the issue of finding the shortest path, the ACO algorithm was

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 13, Issue 6, June 2024
DOI: 10.17148/IJARCCE.2024.13602

employed for the first time to handle the nurse scheduling challenges. This was achieved by evaluating the dynamic regional problem at the Vienna Hospital compound [8]. The ACO algorithm has demonstrated advancements compared to the traditional method for solving graph coloring problems [9]. Kashef et al [10] have suggested a modified version of the ACO method called Advanced Binary ACO (ABACO) algorithm, which has shown certain enhancements in its outcomes. ACO possesses a high level of resilience, a reduced number of parameters, uncomplicated configurations, and may be seamlessly integrated with other optimization algorithms.

Here is the pseudocode of Ant Colony Optimization algorithm (ACO):

Algorithm 1 Pseudocode of Ant Colony Optimization algorithm

```
Initialize necessary parameters and pheromone trials;
while not termination do
    Produce a count of the ant population;
    Compute the fitness values corresponding to each ant;
    Discover the optimal solution using selection procedures;
    Revise the pheromone experiment;
end while
```

Fig .2 The pseudocode of Ant Colony Optimization algorithm (ACO)

E. Particle Swarm Optimization algorithm (PSO)

Particle swarm optimization (PSO) is an intelligence algorithm which is developed by Dr. Kennedy and Professor Eberhart, inspired by the collective hunting behaviour of animals [11]. The local optimization process achieves the global optimal solution by utilizing specific search algorithms and individuals' interactions. The Particle Swarm Optimization (PSO) algorithm is initially used to solve non-linear, continuous optimization problems. It offers several benefits, such as a reduced number of parameters to configure and a straightforward mathematical model that does not require complex calculations. One of its primary applications is in robot path planning. However, traditional Particle Swarm Optimization (PSO) encounters issues such as premature convergence and extended optimization time when dealing with intricate maps.
The step by step of Particle Swarm Optimization algorithm is shown bellow:

- ***Step 1:*** *Set the values of the algorithm's constants.*
- ***Step 2:*** *Set the solution's initial values for position and velocity using the solution space.*
- ***Step 3:*** *Assess the physical condition of every individual particle.*
- ***Step 4:*** *Revise the personal and overall highest achievements (pbest and gbest).*
- ***Step 5:*** *Revise the speed and location of every individual particle.*
- ***Step 6:*** *Proceed to step 3 and continue iterating until the termination condition is met.*

III. THE PROPOSED ALGORITHM FOR MAP APPLICATION

The A* algorithm was initially documented in 1968 by Peter Hart, Nils Nilsson, and Bertram Raphael from the Stanford Research Institute [12]. The A* method is an enhanced version of the Dijkstra algorithm used in computer science to efficiently determine the shortest path. The inclusion of the processing phase in converting the map to the grid, along with the heuristic function, makes it the most suitable method for determining the shortest path in video game and navigation maps.

The "best fit" approach is used to determine the most optimal nodes for reaching the target node from any given node. The A* algorithm is categorised as an acceptable heuristic algorithm [13]. The ASTAR method incorporates the evaluation function $f(n)$ for the current node n, and the definition of this evaluation function is presented in the following formula.

$$f(n) = g(n) + h(n)$$

Where :

- *The function $g(n)$ in (1) represents the quantification of the real expense incurred from the initial point to the current node n.*
- *The function $h(n)$ in (1) represents the estimated minimal cost from the current node n to the end point. $h(n)$ is sometimes referred to as the heuristic function.*

Impact Factor 8.102 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 13, Issue 6, June 2024

DOI: 10.17148/IJARCCE.2024.13602

It is worth mentioning that the intuitiveness of $f(n)$ stems from its reliance on an inherent function $h(n)$ that is grounded in prediction.

If the heuristic function $h(n)$ is equal to 0, meaning that no global information is used, the A* algorithm simplifies to the Dijkstra algorithm. Hence, the conventional Dijkstra algorithm can be considered as a specific instance of the Astar algorithm. The function $h(n)$ must satisfy the compatibility criterion, meaning it cannot be more than the real minimal cost from node n to the endpoint. If the heuristic function meets the compatibility criterion and the original problem has an optimal solution, it can be proven that the ASTAR algorithm is guaranteed to locate the optimal path [14]. The algorithm possesses a rather uncomplicated framework that employs the operation of addition in an upper position. In the method that employs a priority queue as its data structure, the node with the highest priority is determined by the node with the lowest value of $f(n)$.

The operation of the algorithm is as follows :

- ***Step 1:*** *The algorithm selects the node with the lowest value, which is considered the most significant, at each stage. It then removes this node from the queue.*
- ***Step 2:*** *The values of all neighbouring nodes are modified. This node now has a cost associated with reaching it, which is represented by the function* $f(n)$.
- ***Step 3:*** *The method iterates through the phases until the target is attained, which occurs when the target node is present in the priority queue or when there are no remaining nodes in the queue*.

IV. RESEARCH METHOD

Based on Figure 3, The initial step in the requirements stage is to ascertain the specific needs and requirements of potential users for the programme that is to be developed. At this point, it is evident that prospective consumers require the ability to easily locate the quickest route to the rescue supply location, and effortlessly modify the map to indicate regions that are subject to relocation or vice versa, to prevent users from accessing outdated information, which could potentially jeopardise their ability to reach rescue supplies points. Next, go on to the design phase. Based on the previously established requirements, it can be inferred that the task of finding the shortest path can be accomplished by implementing the ASTAR algorithm. The database will store data obtained from rescue supply stations, as well as maps that have been previously edited. During the deployment stage, the programme is subjected to testing by multiple potential users. During this stage of deployment, valuable inputs are acquired that can aid in the development of this programme. At a later point in time. During the maintenance stage, the programme undergoes maintenance procedures, the update of the maps. This is performed to rectify any errors that were not identified during the preceding phase.

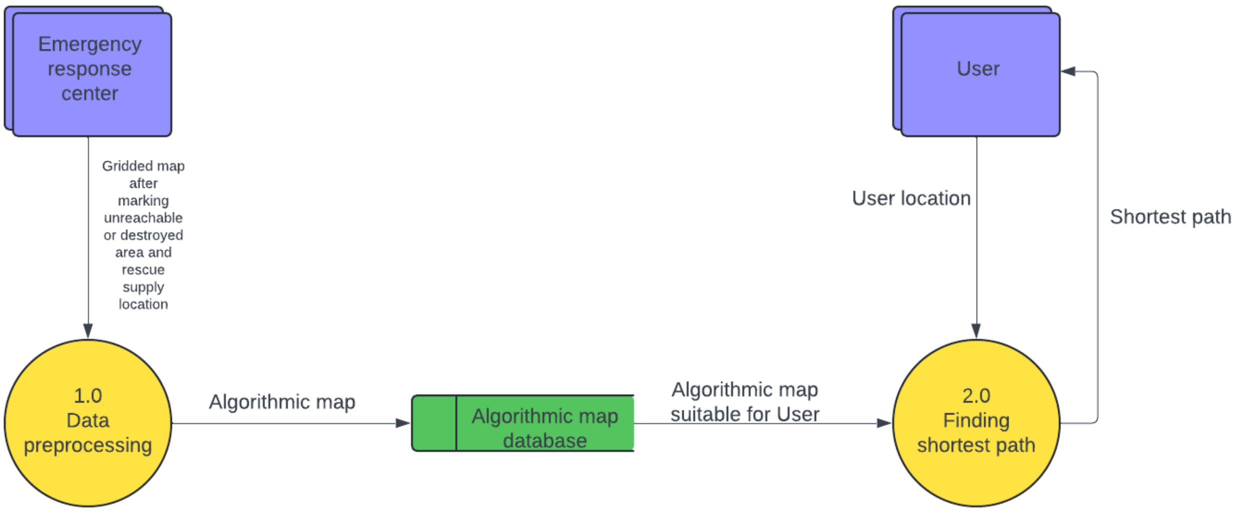

Fig .3 Data flow diagram for implementation

A. Data preprocessing stage for the algorithm

The map depicts the aftermath of an earthquake in Kahramanmaras, Turkey on February 8, 2023, as captured by a satellite. Most pathfinding algorithms rely on grid representation properties for navigation performance [15].

ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⨳ Peer-reviewed & Refereed journal ⨳ Vol. 13, Issue 6, June 2024
DOI: 10.17148/IJARCCE.2024.13602

The programme maps the jpeg file, which has the file extension ".jpeg", by dividing it into grids. Diagram. Figure 4 displays a satellite-derived map, whereas Figure 5 presents a $26*49$ gridded map. Figure 5 displays the user's present location by highlighting the corresponding grid cell in green. The entrance to the emergency shelters in a stadium is highlighted in red within its respective grid cell.

Fig .4 The map part taken from satellite

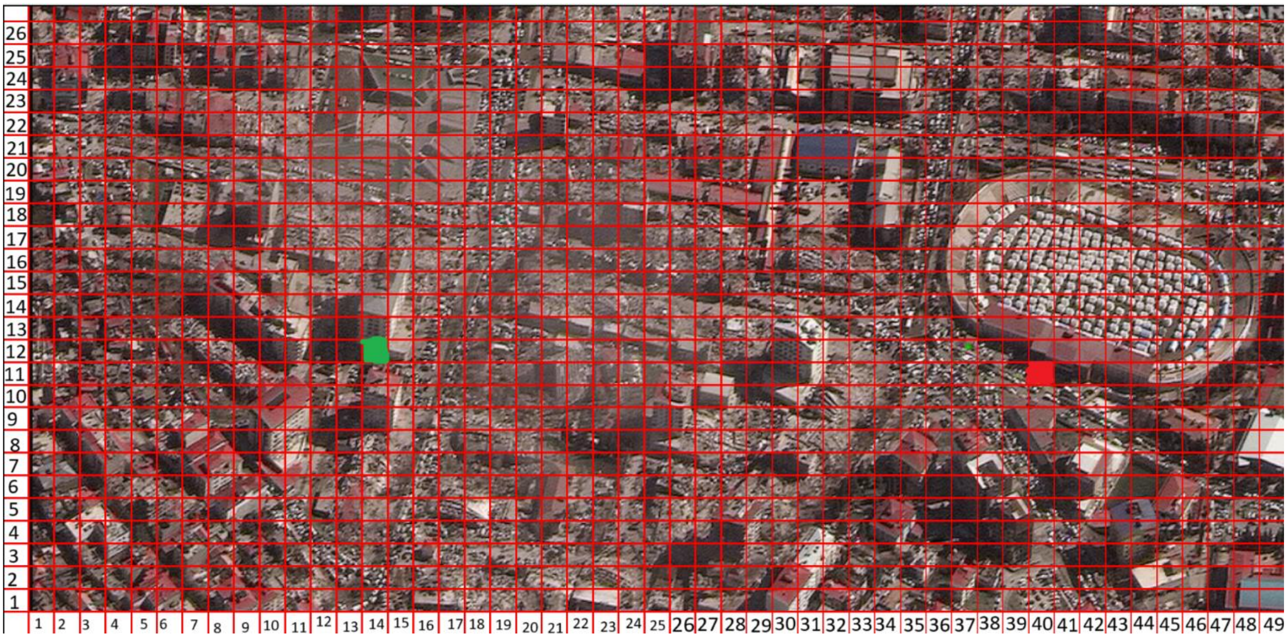

Fig .5 Part of gridded map

Unreachable or destroyed locations marked with white "X" on the $26 * 49$ grid map.
The grid map after updating the information is shown in Figure 6.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2024

DOI: 10.17148/IJARCCE.2024.13602

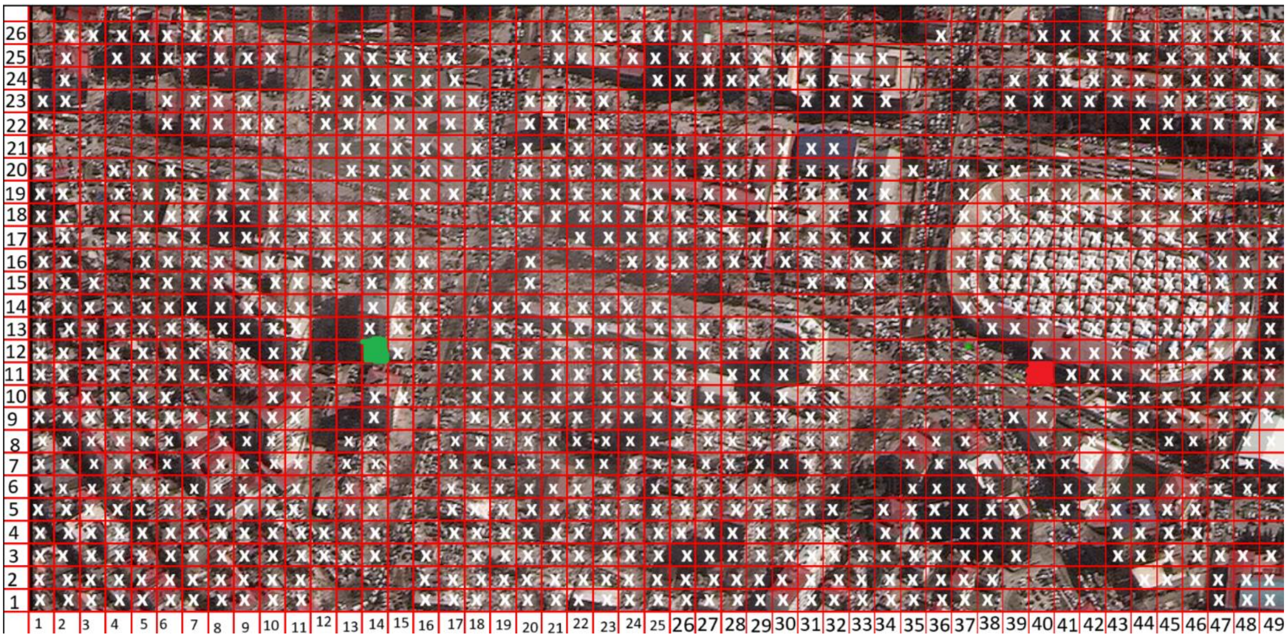

Fig .6 Part of gridded map after marking unreachable or destroyed area

Subsequently, the grid map containing designated places that are inaccessible or have collapsed was transformed into an algorithmic map. The map displays accessible areas highlighted in white, starting locations indicated in red, reach points denoted in blue, and inaccessible spots shown in grey. Figure 7 displays the algorithm map.

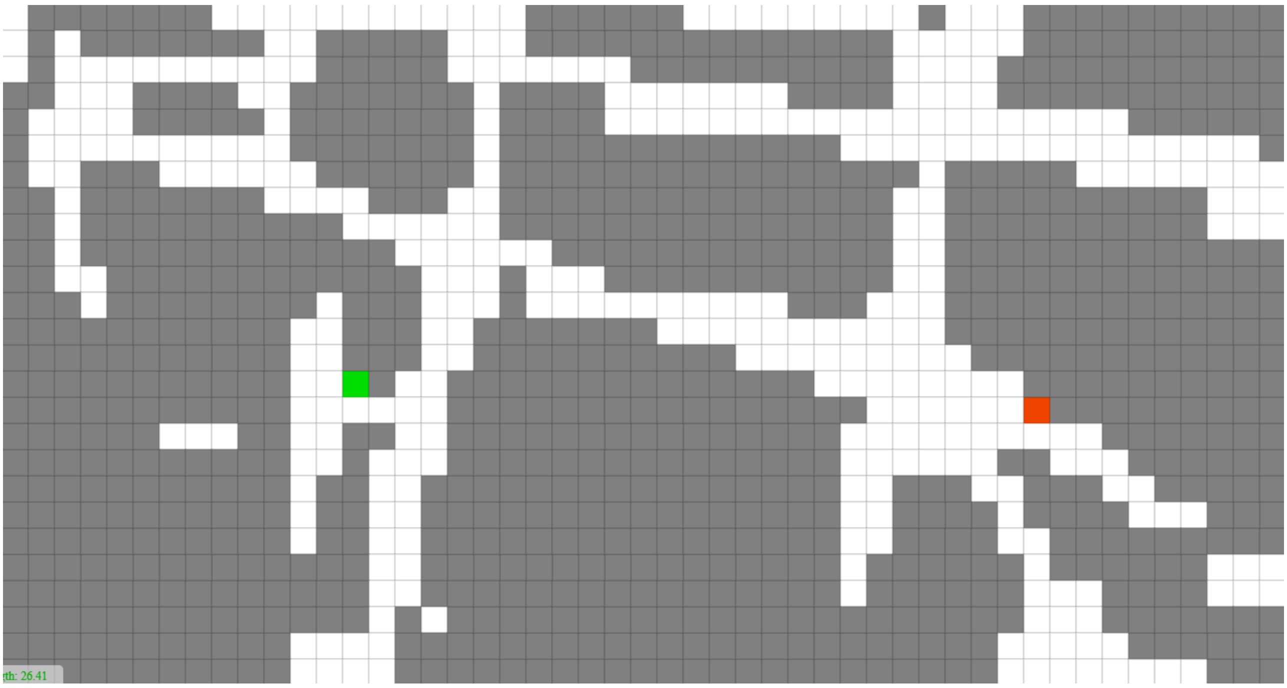

Fig .7 The algorithmic map after the transformation

B. Experiment and output

In order to determine the shortest path to the rescue supply location, the ASTAR algorithm is implemented. This is done by creating a program in the form of a website that can present the closest path to the rescue supply site based on the algorithmic map data. This software is implemented using HTML, JavaScript, and the A* method for solving and visualizing the shortest path problem that has been explored.

a. Output in algorithmic map and real map

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 13, Issue 6, June 2024

DOI: 10.17148/IJARCCE.2024.13602

Upon executing the ASTAR algorithm, we have obtained outcomes derived from the algorithmic map. The algorithm has identified the most efficient route from the user's current location to the stadium entrance where the emergency shelters are situated. The path has been indicated according to figure 8.

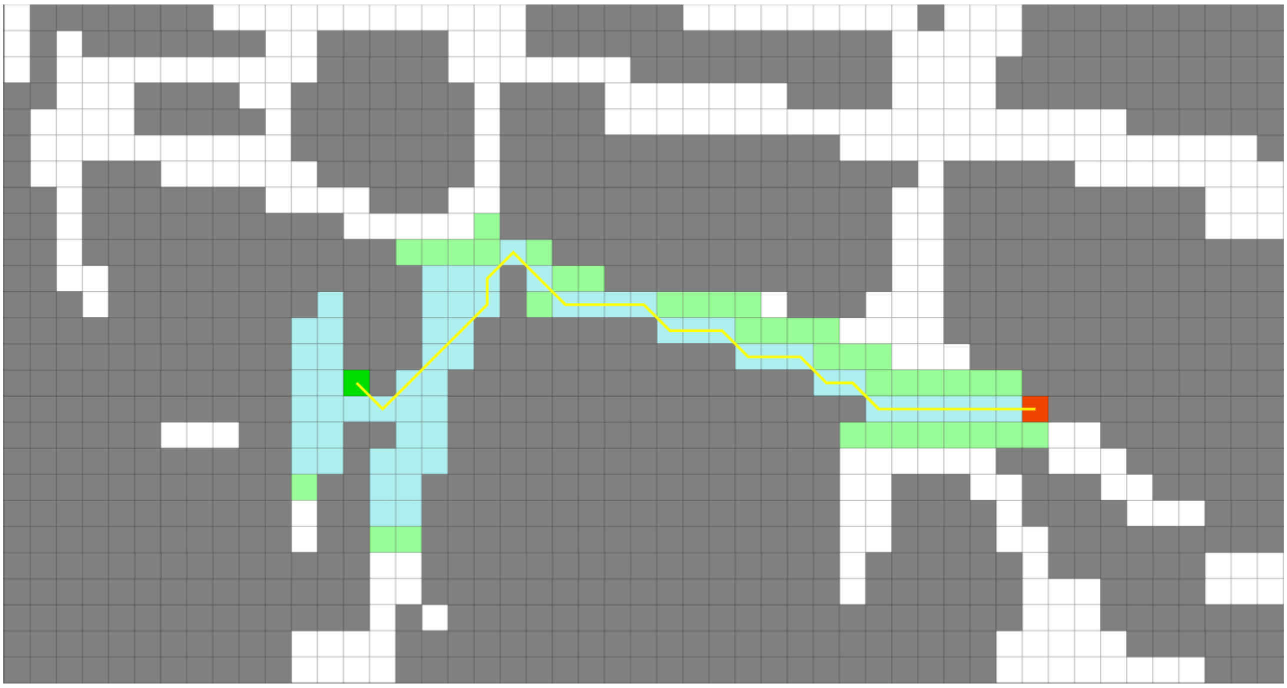

Fig 8. Output in algorithmic map after using ASTAR algorithm to find the shortest path

The algorithmic map depicting the shortest path will thereafter be transformed to a grid satellite map and ultimately to the original satellite map. Displayed in figure 9 and figure 10, correspondingly.

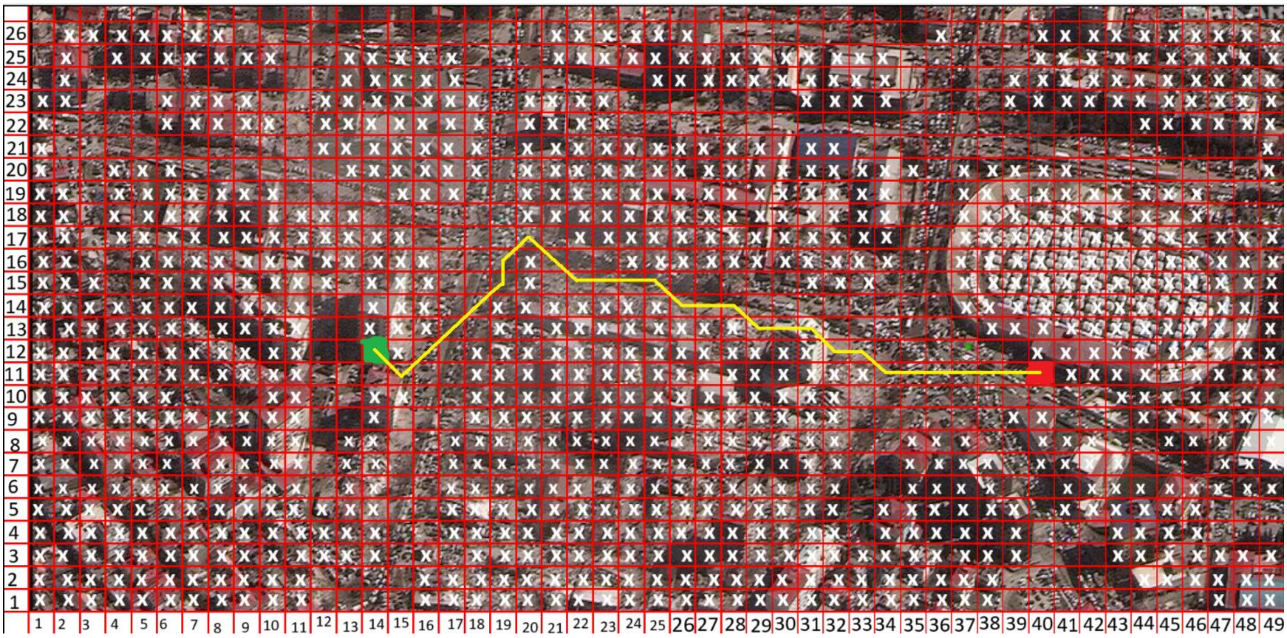

Fig 9. Output in grid satellite map after using ASTAR algorithm to find the shortest path

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 13, Issue 6, June 2024

DOI: 10.17148/IJARCCE.2024.13602

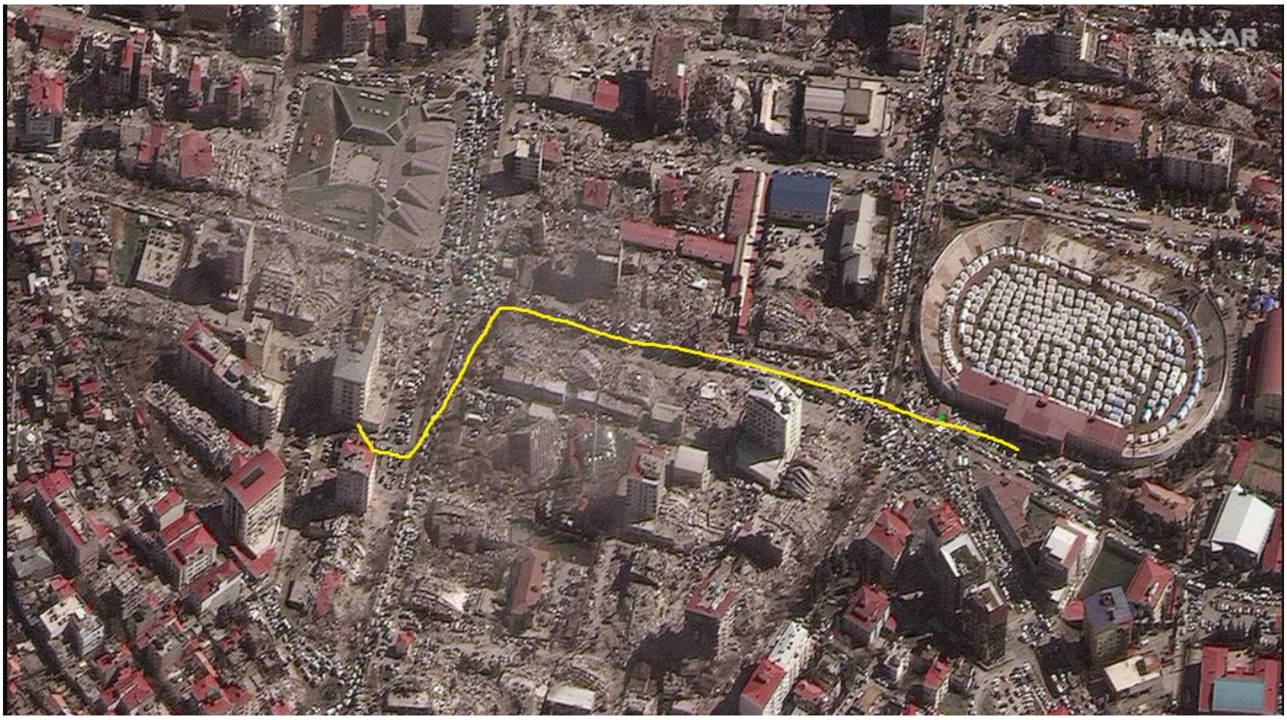

Fig 10. Output in the map part taken from satellite after using ASTAR algorithm to find the shortest path

C. Evaluation

a. Introducing research results

This article showcases the utilization of the ASTAR algorithm to assist in natural disaster and conflict relief operations. The specific scenario is illustrated in figure 10.

b. Discuss the meaning of the results.

The document provides a detailed description of the required steps to process data and execute the algorithm for determining the optimal route. This article's findings have revealed a technique for implementing the algorithm and have demonstrated the feasibility of using ASTAR.

This provides a means to utilize technology to address issues, adapt regions in the event of natural disasters, or modify humanitarian routes. By depicting that quantity of information on an algorithmic map. Additional data updates can be generated to aid the algorithm in identifying the most efficient route.

c. Limitations of the study

A weakness of the research is the absence of any identified applications for marking inaccessible locations or areas with insufficient infrastructure, which necessitates human marking. Individuals require reliable and thorough sources of information while verifying data or examining satellite maps.

d. Future research directions

A potential avenue for future research involves the utilization of machine learning techniques to detect and analyze inaccessible regions or to enhance data processing for algorithms. Alternatively, a comprehensive product application could allow rescuers to designate and assess the places that are reachable.

V. CONCLUSION

This research work involves the conversion of maps obtained from satellite pictures or other map package sources into algorithmic maps. This was achieved by transforming the maps into grid maps and identifying places that were inaccessible. Convert the marked grid map into the algorithm map, and then display the quickest path from the user's current location to the emergency shelters. The map processing component of the algorithm map is executed manually. The task can be performed digitally through the utilization of image processing methods and the creation of requisite software. There are numerous algorithms available to address the problem of determining the shortest path. However, the ASTAR method was selected for this example because to its ability to do fast searches with minimal complexity, through the support of heuristics algorithm. Additionally, it is deemed suitable for preparing the data required by the algorithm in this unique scenario. Through the experimentation with the shortest path search algorithm in this scenario, it is evident

DOI: 10.17148/IJARCCE.2024.13602

that employing the ASTAR algorithm enables the community to effectively navigate and surmount the devastation caused by natural disasters and warfare, thereby illustrating the utilization of information technology to resolve enduring issues.

REFERENCES

[1]. Y. S. Triana and I. Syahputri, "Implementation floyd-warshall algorithm for the shortest path of garage," *International journal of innovative science and research technology*, vol. 3, no. 2, pp. 871–878, 2018.
[2]. M. Głabowski, B. Musznicki, P. Nowak, and P. Zwierzykowski, "Efficiency evaluation of shortest path algorithms," in *The Ninth Advanced International Conference on Telecommunications (AICT) 2013 Proceedings*, 2013, pp. 154–160.
[3]. Ramadiani, D. Bukhori, Azainil, and N. Dengen, "Floyd-warshall algorithm to determine the shortest path based on android," *IOP Conference Series: Earth and Environmental Science*, vol. 144, no. 1, p. 012019, apr 2018. [Online]. Available: https://dx.doi.org/10.1088/1755-1315/144/1/012019
[4]. J. C. Dela Cruz, G. V. Magwili, J. P. E. Mundo, G. P. B. Gregorio, M. L. L. Lamoca, and J. A. Villaseñor, "Items-mapping and route optimization in a grocery store using dijkstra's, bellman-ford and floyd-warshall algorithms," in *2016 IEEE Region 10 Conference (TENCON)*, 2016, pp. 243–246.
[5]. L. Zhao and J. Zhao, "Comparison study of three shortest path algorithm," *in 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC)*, 2017, pp. 748–751
[6]. V. Pandey, S. Yadav, and P. Arora, "Retiming technique for clock period minimization using shortest path algorithm," in 2016 *International Conference on Computing, Communication and Automation (ICCCA)*, 2016, pp. 1418–1423.
[7]. S. Yigang, F. Jie, and Z. Hongying, "Research on the shortest path algorithm of vehicles dispatch in airport emergency rescue," in *2011 Third International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2, 2011, pp. 107–110.
[8]. W. J. Gutjahr and M. S. Rauner, "An aco algorithm for a dynamic regional nurse-scheduling problem in austria," *Computers and Operations Research*, vol. 34, no. 3, pp. 642–666, 2007, logistics of Health Care Management. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054805001139
[9]. E. Salari and K. Eshghi, "An aco algorithm for graph coloring problem," in *2005 ICSC Congress on Computational Intelligence Methods and Applications*, 2005, pp. 5 pp.–.
[10]. S. Kashef and H. Nezamabadi-pour, "An advanced aco algorithm for feature subset selection," *Neurocomputing*, vol. 147, pp. 271–279, 2015, advances in Self-Organizing Maps Subtitle of the special issue: Selected Papers from the Workshop on Self-Organizing Maps 2012 (WSOM 2012). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231214008601
[11]. L. Zhao and J. Zhao, "Comparison study of three shortest path algorithm," in *2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC)*, 2017, pp. 748–751.
[12]. P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic de termination of minimum cost paths," IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100–107, 1968.
[13]. R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality of a*," *J. ACM*, vol. 32, no. 3, p. 505–536, jul 1985. [Online]. Available: https://doi.org/10.1145/3828.3830
[14]. I. Chabini and S. Lan, "Adaptations of the a* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 60–74, 2002.
[15]. Z. A. Algfoor, M. S. Sunar, and H. Kolivand, "A comprehensive study on pathfinding techniques for robotics and video games," *Int. J. Comput. Games Technol.*, vol. 2015, pp. 736 138:1–736 138:11, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:21754736

