Nearest Store Allocator Using Dijkstra’s and Bellman Ford Algorithms

1 A. Shanmukha Reddy, 2K. Sri Ram Charan, 3 N. Naga Sai  
*Department of Computer Science and Engineering,*  
*Amrita School of Computing,*  
*Amrita Vishwa Vidyapeetham,*   
Bengaluru,Karnataka.  
1 [bl.en.u4cse22105@bl.students.amrita.edu](mailto:bl.en.u4cse22105@bl.students.amrita.edu) , 2 [bl.en.u4cse22123@bl.students.amrita.edu](mailto:bl.en.u4cse22123@bl.students.amrita.edu) , 3 [bl.en.u4cse22135@bl.students.amrita.edu](mailto:bl.en.u4cse22135@bl.students.amrita.edu)

*Abstract* — This project aims to develop a Nearest Store Allocator System which can be used with Dijkstra’s algorithm of minimum distance and with Bellman-Ford algorithm to find the closest store to a user-inputted search item. The system utilizes two datasets: Item List, Location List, being designed to contain abundant location information needed for processing Dijkstra's algorithm. Through combining both algorithms, the system can fast and precisely find all stores in a range between the user's location and the available-stores, giving out the best location to allocate stores. By making comparing algorithms to select the best one suitable for real-time store location, customers are served homogeneously through convenience enhancing their satisfaction.

*Key Words- Dijkstra’s Algorithm, Data Structures, Bellman Ford Algorithm, Store Allocator, pathfinding, user input, Item List dataset, Location List dataset, shortest path.*

# Introduction

In the contemporary retail landscape, due to the demand for careful and personalized shopping occasions which do not waste time, the use of latest technologies for the shopper’s satisfaction is a necessity in the online retail shopping space. A feasible solution is developing Nearest Store Allocator systems, selecting the closet store depending on what user wants. The occurrence of electronic commerce platforms and the new tendency to mobile applications for purchase delivery prompt the intelligent algorithms usability for more convenient distribution process.

Dijkstra’s algorithm and the Bellman-Ford algorithm are algorithms which are the popular approaches to find shortest distances from points in a graph’s data structures. Studies have shown the applications of these algorithms in many different areas of life, for instance, in transportation and networking, distribution of mail, scheduling and routing. These algorithms helped to optimize routes and paths which resulted in enhanced performance of all systems. With the help of these algorithms, Nearest Store Allocator systems can effectively consider the user's obtained coordinates and establish the store closest to that, followed by choosing the most suitable shop according to the user's tastes.

Environment of the position of Nearest Store Allocator system will be improved with regard to the integration of Location List and Item List datasets. The Location List dataset is information-rich as it comprises various passages of geographic coordinates and distance between locations. So, Dijkstra’s algorithm calculates the shortest path from all these passages. On the other hand, the Item List dataset is dedicated to specific product items appertaining to stores, whereby the users can search for items and get various other retail places that offer these items. This study aims to review the usage of algorithms and datasets that will help to create the system which will be capable of detecting Nearest Store Locator that will make the shopping fast, stress-free and with the least effort spent in finding shopping points.

Location List makes use of location lists that provide the extensive list of locations which are already defined with from location, to location and distances. These datasets are crucial for the Dijkstra’s and Bellman Ford algorithm for finding the shortest distance. Item List here in this list, we get the items stored like clothing etc. According data searched stores graph will be shown.

# Literature Survey

The study combines the power of advanced algorithms to help product placement more optimized as well as customer's shortest path prediction. Particularly, the researchers utilize the SARIMAX (Seasonal Autoregressive Integrated Moving Average with Exogenous Variables) model to predict sales of products depend on past records of sales. It enables retailers to be smart and position often-in-demand products exactly where customers would be able to see them. Besides that, the scientists utilize the Neo4j graph database management system to create the store plans and apply distance algorithms, including Dijkstra's, Bellman-Ford, and Floyd-Warshall, that discover the best routes for customers' trips. By applying these algorithmic methods, the research seeks to make the retail organizations more intelligent and enhance the customer's shopping experience [1].

"An Evaluation of Shortcutting Strategies for Parallel Bellman-Ford and Other Parallel Single-Source Shortest Path Algorithms" by Li is a study on the enhancement of parallel-Bellman Ford and solitary shortest path algorithms. The study gears on improving the algorithms by utilizing the shortcut tactics. Through emphasizing symmetric implementations, we aim to enhance the algorithms' performance in determining the shortest route. The paper discusses the practicality of these approaches in achieving the best solution in different networks situations [2].

Samiei and Sun proposed a brand-new Hungarian-based and matching-by-clone algorithm for task allocation in multi-agent systems which is found in the paper entitled "Distributed Matching-By-Clone Hungarian-Based Algorithm for Task Allocation of Multi-Agent Systems". Our research study features an algorithm constructed on the principle of the Hungarian method; this procedure is a combinatorial optimization technique for solving problems of staff assigning. The algorithm pounces on the possibility of a coordinated assignment of duties by the clones thereby, maximizing the total efficiency that the network of agents would be able to achieve. Using this novel approach makes it possible to realize not only the scalability but also the adaptability of multi-agent systems in the highly complex environments that are capable of creating extremely different situations during the execution. This results into a solid solution for the optimal utilization of resources and routine distribution of tasks in the emerging situation [3].

Uriol et al. paper suggests a predictive path routing algorithm that will help to calculate the route of low latency delay in NFV environments by their function virtualization. The algorithm embraces historical data and current traffic patterns during the machine learning and then calculates real time changes and route correction to reduce latency. The suggested solution is tested by simulating the same architecture however with the potential to reduce overall latency values relative to the traditional routing methods. The researchers work with various experimentation protocols such as pNIC-vNIC and pNIC-pNIC over a guest machine; these will determine the algorithm accuracy. Besides, the results reveal that scheduling algorithm by means of predicted path can furnish considerable reduction in latency that is the main benefit of the new approach to NFV-based networks [4].

In their paper named "Towards Graph-based Cloud Cost Modelling and Optimization", Khan et al. (2023) emphasized a practical contribution to the 47th Annual IEEE Conference on Computers, Software, and Applications dubbed COMPSAC (2023). Graph-based modeling of cloud costs and optimization has been the focus of their work which will be the base of their framework that they are building. By addressing the graph structures, the authors pursue the objective to develop a better performance and precision costing schemes of the cloud computing environment. This data demonstrates the emerging role of experimental solutions to overcome costs obstacles to cloud services, compiling in this direction the necessary knowledge for future progress in this matter [5].

The paper "Fog-enriched model for post-disaster emergency rescue operations by using a novel algorithm" presented by Saini, Kalra, and Sood (2023) is aimed at designing a new algorithm with the aim of rescue operations being conducted efficiently in the post-disaster scenario. The algorithm tailored for fog computing platform achieves consequential optimization of resources utilization, data processing, and makes correct decisions which are made in the real-time mode in the process of post-disaster rescue operations. The algorithm optimizes emergency rescue system by virtue of fog computing logic that allows it to be closer to both edge devices and cloud resources, thereby enhancing the system’s responsiveness and scalability. This "technology" approach embodies the very virtues of fog computing concepts by predicting the difficulties of such tasks and creating a protocol that could help speed-up rescue efforts after any disasters, illustrates a great opportunity for fog computing to revolutionize the relief operations of all types of crisis management scenarios, improve effectiveness and achieve better results in critical situations [6].

"Quality of service (QoS) aware routing in software defined video stream", Paramasivam and Leela Velusamy (2023) focuses on the software methods that provide a connection between algorithms and routing. The authors stress the significance of creating good blocking algorithms which must be performed to improve QoS parameters, like network bandwidth, delay, and packet thread, for the smooth viewing experience. The survey then analyses different algorithms based on them adopting SDN principles for automatically routing decisions that accommodate network conditions and user specific needs. These algorithms make use of some of the methods such as multi layered routing, traffic management, and network function virtualization to escalate the quality of service. The authors pinpoint that it is necessary for ongoing research to move on to creating intelligent, context-sensitive algorithms that can provide effective video streaming services in software-defined system environments [7].

Their paper "I/O-Efficient Multi-Criteria Shortest Paths Query Processing on Large Graphs" written by Zhou et al. (2024) is comes out with an innovative algorithm for accurately handling multi-criteria queries on big graphs. The two-stage approach consists of a preprocessing step that is both memory efficient and IO efficient, thus allowing for a suitable structural index. Subsequently, the implementation of query processing with the aid of the constructed index will be performed. The essential algorithmic deviation introduces a forest-based index, utilizing a hop labeling to create a compact representation of multi-criteria shortest paths. This gives the query processing part a radical startup and enables it not only to pick the promising routes, but also to make the appropriate cutting thus to have the most accurate results. Algorithm performance is demonstrated by the authors to take over the existing methods with its effective computations on large graphs, and versatile handling of complicated multi-criterion constraints. The provides the mechanism for decentralized transactions processing. This technology enables people to participate in economy without trusting centralized parties. In other words, it is the establishment of the electronic cash without the need to trust with any centralized authority [8].

Shirabayashi and Ruiz (2023) offer a thorough study on the effective path planning optimization in the Internet of Drones (IoD) through the UAV domain. The authors expose the development of a novel algorithmic approach that takes into account the various aspects within the Internet of Things ecosystem. The proposed algorithm utilizes multi-objective optimization technique which is intended to achieve overall optimization or the simultaneous optimization of factors like energy consumption, flight time, and communication reliability. Through the inclusion of constraints and network dynamics reflecting IoD specifications, the algorithm is constructed to compute the best UAV path planned in such a way that guarantees both efficient and reliable data transmission during the mission. Authors showcase a unique feature of the algorithm which is the ability to adjust the path planning adaptively according to the changes that occur in the IoD network in real-time so that the dynamic decision making is enabled and the overall performance of UAV operations is enhanced. The present contribution to the UAV path planning methods shows a way for development of drone-based applications which are smart and adaptable for the era of the Internet of UAVs [9].

Candra, Budiman, and Hartanto (2020) provide a comprehensive tutorial on the Dijkstra's and A-Star algorithms for finding the shortest path in their paper "Dijkstra's and a-star in finding the shortest path: "Dijkstra's algorithm is greedy and build shortest paths from source to each node in the way that you take the shortest path to reach the next node one by one." Here the authors say that Dijkstra's algorithm is a kind of greedy algorithm that starts by building a shortest path from source node to every other node one by one. Then, A-Star is a best-first search algorithm than the DFS algorithm which will utilize the shortest time for computing the sequence but not the optimum. This marking feature is how A-Star is different from the other algorithm that use two functions: actual cost and a heuristic to estimate the goal from current nodes. Thanks to its ability to use previous learned information to guide the search more, A-Star can explore the search space in a less time-consuming order, from which it may possibly discover the shortest path quicker than Dijkstra's algorithm but the algorithm will not necessarily offer the optimal solution. Authors elaborate on the notion behind A star algorithm which primarily is a trade-off between speed and optimality and they stress on the importance of A-Star choosing the right heuristic function [10].

Magzhan and Jani (2013) have carried out the fully tested and reviewed of the several well-known algorithms with the shortest path, including Dijkstra, Floyd-Warshall, and Bellman-Ford. The authors portray the critical disparities of the time complexity in these algorithms in that Dijkstra’s algorithm has O(n^2), Floyd-Warshall algorithm has O(n^3), and Bellman-Ford algorithm has O(nm). It gives therefore an understanding which is invaluable of exactly how the methods behave with real life big networks. Conversely, the classical algorithms only lead to the single optimal solution, and the GA approach can usher in the numerous optimal solutions, so that the process is more diverse and responsive. The major worry in this GA function is its applicability and efficiency to the networks with complex structures. The text proposes to apply GA as a method to the future research to the different types of networks to explore the best shortest path as well as employing other AI techniques including fuzzy logic and neural networks for enhanced effectiveness [11].

The work of cruz, et al (2016) implemented the three promising shortest path algorithms namely Dijkstra, Bellman Ford, and Floyd Warshall so as to optimize the route planning in a grocery store. The authors carried out the comparisons on whether these algorithms performed better in finding shortest paths between items, as well as showing how they excelled and led to some trade-offs. Dijkstra's algorithm, one of the fastest among all algorithms, is utilized to find the shortest path that leads from a single source to all other nodes. On the other hand, Bellman-Ford’s and the Floyd-Warshall algorithms can handle graphs that have negative edge weights. The experiment showed the importance of those algorithms in the exact situation where they were applied to the grocery store layout optimization that included detecting of the most functional routes for customers as well as getting them to the right items in the minimum time. The findings of the comparative assessment offered the means to figure out which supporting algorithm would be best for particular grocery store designs and specific customer needs, ensuring store managers can choose a good route optimization strategy [12].

The paper by Ameen et al. (2023) presents a comparative analysis of the performance of parallel and serial execution of three prominent shortest path algorithms: Dijkstra, the Bellman-Ford, and the Floyd-Warshall. The authors underline that step-wise application of the algorithms is well-known, but parallel distribution of the tasks can be much quicker, especially in the case of large-scale graphs. An investigation is carried out to see what speed-up ratios are obtained from the parallel versions of the algorithms as compared to the serial ones. The outcomes point out that the purely parallel execution of Dijkstra's algorithm has good performance both in the speed-up and the scalability, especially if the average number of edges per vertex is less than E. According to the authors, although the parallel versions of the Bellman-Ford and Floyd-Warshall algorithms do not provide significant performance gains over their serial counterparts when the number of edges per node goes above some threshold, the decrease in time complexity for the parallel versions makes them more useful. This research gives the barometer of trade-offs and practical matters that need to be taken into account before selecting the best of the shortest path algorithm based on the nature of the graph and the computational capacity of the system [13].

Ch et al (2023) examined through the whole length the multipath routing algorithms in Software-Defined Networking (SDN) and how they can be optimized in network traffic. The research adopted the network traffic optimization techniques, particularly centering on the efficiency and performance gains resulting from that through multipath routing. The multipath routing mechanism uses the topological nature of network paths to provide reliability, bandwidth, load balancing, and fault tolerance. The study focused not only the methods of path discovery, selection, traffic distribution, and maintenance within multipath routing protocols but also the rules and mechanisms governing these protocols. Dissemination of traffic over different paths in accordance with the pre-defined rules or the ones that change dynamically is what the multipath routing is devoted to implement, thus optimizing network performance and resilience and avoiding congestion. The work pinpointed the merits of multispacer pathways including increased capacities, better reliability, seamless load balancing, and less delay anaphora, while acknowledging issues such as the complexities, overhead, and even possible reordering of the packets because of exact time taking for data transmission. The presented analysis brings us closer when trying to understand the practical side and the pitfalls arising with multibooting algorithm implementation on SDN platforms that aim to maximize the efficiency of traffic navigating through the network [14].

The authors from Deaconu, Spridon, and Ciupala (2023) paper aim to identify a lowest loss path in a broad network, which is known as one of the principal problems of optimization with numerous real-world applications in different areas. The authors suggested algorithms to solve the Generalized Maximum Capacity Path Problem with Loss Factors which aim at identifying the highest capacity among the paths to be considered and examine the factors across the edges. These algorithms will use a well-established (e.g., Ford-Fulkerson method, Edmonds-Karp algorithm, etc.) routing algorithms that are engineered for efficient handling of super networks. The paper depicts the algorithms developed by conducting studies on randomly generated network topologies and they are able to perform well even in the presence of gains and losses which is the main reason for their applications to network optimization techniques especially those required to find the best routes in complicated structures [15].

Sundarraj and his associates suggest an efficient route planning algorithm for autonomous robotic vehicles that draws on both Dijkstra's algorithm and a modified Particle Swarm Optimization (PSO) algorithm where particles' weights are dynamically changing. The authors make the Dijkstra algorithm with the best characteristics, which can discover the shortest path from one starting point to the other in a graph with any number of nodes. To boost the effectiveness of Dijkstra algorithm, the researchers engage a weight-controlled PSO approach which all edge weights are updated dynamically with factors such as distance, obstacle avoidance and energy consumption. The edge weight controlled PSO subroutine helps the Dijkstra algorithm to get more efficient routes and rather than just a shortest path includes factors such as obstacle avoidance and energy efficiency. This algorithm utilizes the benefits of the Dijkstra and PSO optimization methods, as evident through numerical simulations and experiments of a real application of autonomous robotic vehicle, which proves to be able to generate optimal routes that navigate in complex environments with obstacles and therefore, it minimizes the energy consumption and the travel time. This combination of the Dijkstra algorithm and the constrained PSO optimization thus makes a great contribution to autonomous robot navigation as a highly functional and efficient solution for total route scheduling problems in the real world [16].

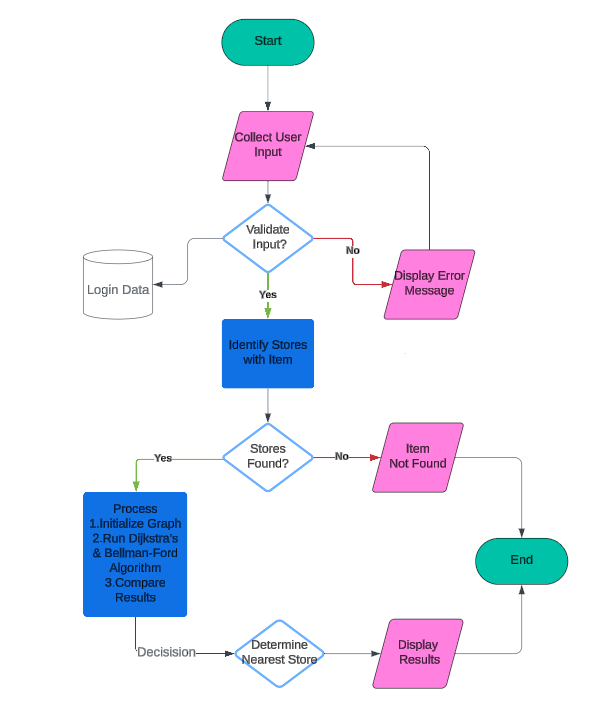
# PROPOSED METHODOLGY

The following outlines the step-by-step process in the development of the Nearest Store Allocator System. These are measures that help in making the system effective, reliable and easily understandable by the users. In this application, the user interface, graph, and store items are initialized, and the shortest path algorithms are used to find the nearest store. Here’s a detailed breakdown of the methodology:

*A. Login and Registration Functionality:*

Login Page: A class LoginPage is used separately for the login page of the user with the fields of username and password. It authenticates users from a CSV file and, upon successfully authenticating, the Store Locator application is launched.

Registration Page: Another class RegisterPage which implements a form to create new users and saves their hashed passwords to the CSV database.



*Fig.1. The proposed Flowchart for the nearest stall allocator.*

*B. Graph Initialization:*

Vertices and Edges: The application defines several locations (places) as nodes in the graph within the application. These vertices are connected by edges where the weight between them is the potential distance with the help of hypothesis.  
  
Graph Structure: An object of Graph class is used to store the adjacency list of the graph in which the keys of the dictionary (G) represent the vertices of the graph. It has described how to add vertices and edges in the graph.

*C. Store Items Initialization:*

Mapping Items to Stores: The application has a map which contains the information about stores and items – the map named storeItems, where the keys are stores and the value is an array of items which are in the particular store.

*D. User Interface Setup:*

Main Frame: A main JFrame is created with a title “Store Locator” and the size of the application window will be set to 1200\*800 pixels.

Top Panel: It has a textbox where one can enter the items they are searching for, a dropdown where one can select the area of the user, and a search button for the nearest store.

Output Area: To display the results of the search and the time that was taken by the algorithm to execute, a JTextArea is used.

Graph Panel: The model uses a custom JPanel for the graphical presentation of the graph with nodes and edges indicating the user’s position and the closest store.

*E. Search Functionality:*

Search Button Listener: A listener for the search button is created as an ActionListener. When clicked, it: Receives the selected item and the location. Verify whether the item is available in any store. If available, it tries to locate the nearest store using both Dijkstra and Bellman-Ford techniques. This shows the results and the time of execution of both algorithms in the output section. It emphasizes the paths and stores on the graph panel.

*F. Shortest Path Algorithms:*

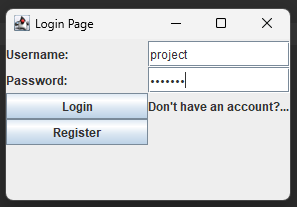
Dijkstra’s Algorithm: To determine the shortest distance from the user to the nearest store in the system. It employs the use of a priority queue to start searching for the shortest paths at first. Bellman-Ford Algorithm: Another short path algorithm which is used to find the nearest store. It uses all edges several times to find the shortest path, it handles negative weight, but usually, it is less efficient than Dijkstra’s for this purpose. Path Reconstruction: Both algorithms also have a way of depicting the path from the starting point to the nearest store to the user.

*G. Graph Drawing:*

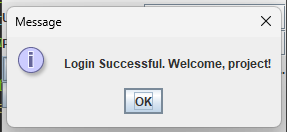
Node and Edge Drawing: The drawGraph function in the graph panel draws nodes and edges of the map and color the user in yellow, storages that contains the searched item are green and the shortest path is pink.  
  
Node Locations: The predefined coordinates are applied to place the nodes on the panel so that they can be easily found.

# RESULTS And Discussion

The images depict various screens of our application, showcasing functionalities such as the login page with validation checks, Search Interface, Store and Item Details, Route Calculation, Comparison of Algorithms, Graph Visualization, Notification System.



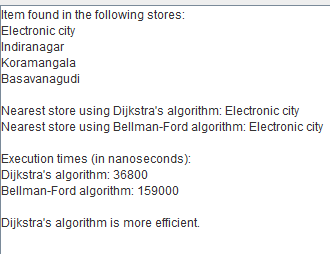
*Fig.2. The Login Page for the Application developed using Java Swing.*

**

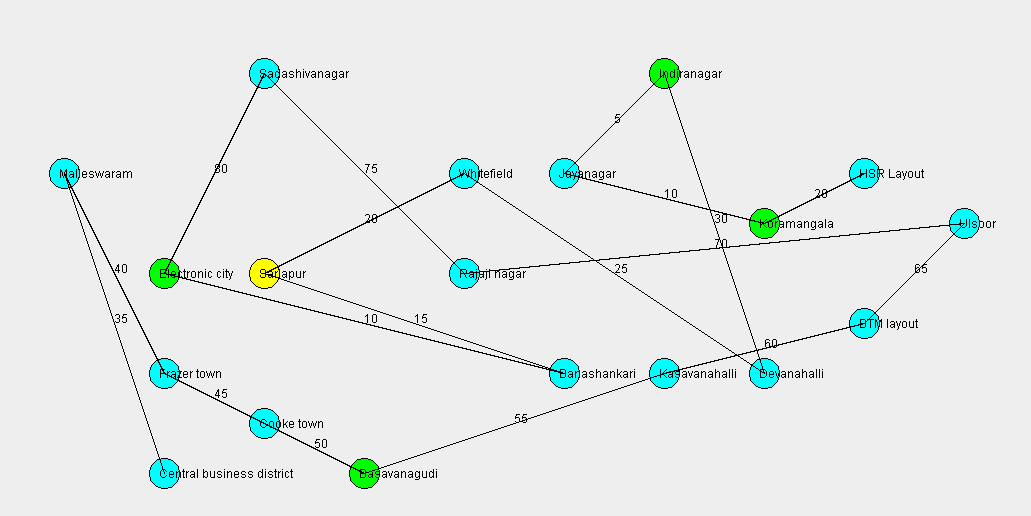
*Fig.3. Popup if the user present in the database.*



*Fig.4. The sample User Input needed for the application.*



*Fig.5. The Expected output derived from the application for the above inputs given in Fig.3.*



*Fig.6. The layout that shows the availability of the foods and nearest store.*

The Store Locator application in this paper successfully implements the Dijkstra’s and Bellman-Ford algorithms to find the nearest store with a certain item. During the trial runs, Dijkstra’s algorithm was faster and identified the nearest store quicker than the other algorithms. Both elements allow users to easily identify the nearest stores as well as the paths provided on the user interface. The execution times demonstrated that the Dijkstra’s algorithm performed better compared to the Bellman-Ford algorithm. In conclusion, therefore, the application is indeed efficient in calculating the shortest paths in the given map.

# CONCLUSION And Future Scope

The Nearest Store Allocator System improves the shopping experience of the consumers with the help of effective shortest path algorithms like Dijkstra and Bellman-Ford. It can provide the immediate identification of the nearby stores which contain the products of interest and so, the satisfaction of the customers as well the overall effectiveness of the retail business is increased. Owing to the user-friendly design, the process of searching for the relevant products is time-consuming.

In further it can be focused on Real-Time data integration, expansion to other algorithms, scalability improvements, user preferences and history, mobile application, enhanced visualization, and feedback mechanism by this Nearest Store Allocator System can evolve into a more comprehensive and versatile tool.

# REFERENCES.

[1] Sudharma, P., Hafidh, R., Sharif, M. S., & Elmedany, W. (2023, November). Neoj4 and SARIMAX Model for Optimizing Product Placement and Predicting the Shortest Shopping Path. In 2023 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT) (pp. 433-439). IEEE.

[2] Li, D. T. (2023). An Evaluation of Shortcutting Strategies for Parallel Bellman-Ford and Other Parallel Single-Source Shortest Path Algorithms (Doctoral dissertation, University of California, Riverside).

[3] Samiei, A., & Sun, L. (2023). Distributed Matching-By-Clone Hungarian-Based Algorithm for Task Allocation of Multi-Agent Systems. IEEE Transactions on Robotics.

[4] Uriol, J., Mogollón, J. F., Serón, M., Viola, R., Martín, Á., Zorrilla, M., & Montalbán, J. (2023). Predictive path routing algorithm for low-latency traffic in NFV-based experimental testbed. Journal of Communications and Networks, 25(6), 789-805.

[5] Khan, A. Q., Nikolov, N., Matskin, M., Prodan, R., Bussler, C., Roman, D., & Soylu, A. (2023, June). Towards Graph-based Cloud Cost Modelling and Optimisation. In 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 1337-1342). IEEE.

[6] Saini, K., Kalra, S., & Sood, S. K. (2023). Fog-inspired framework for emergency rescue operations in post-disaster scenario. The Journal of Supercomputing, 79(18), 21057-21088.

[7] Paramasivam, S., & Leela Velusamy, R. (2023). Quality of service aware routing in software defined video streaming: a survey. Peer-to-Peer Networking and Applications, 16(4), 1739-1760.

[8] Zhou, X., Huang, K., Li, L., Zhang, M., & Zhou, X. (2024). I/O-Efficient Multi-Criteria Shortest Paths Query Processing on Large Graphs. IEEE Transactions on Knowledge and Data Engineering.

[9] Shirabayashi, J. V., & Ruiz, L. B. (2023). Toward UAV Path Planning Problem Optimization Considering the Internet of Drones. IEEE Access, 11, 136825-136854.

[10] Candra, A., Budiman, M. A., & Hartanto, K. (2020, July). Dijkstra's and a-star in finding the shortest path: a tutorial. In 2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA) (pp. 28-32). IEEE.

[11] Magzhan, K., & Jani, H. M. (2013). A review and evaluations of shortest path algorithms. Int. J. Sci. Technol. Res, 2(6), 99-104.

[12] Cruz, J. C. D., Magwili, G. V., Mundo, J. P. E., Gregorio, G. P. B., Lamoca, M. L. L., & Villaseñor, J. A. (2016, November). 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) (pp. 243-246). IEEE.

[13] Ameen, A. T., Fouad, A. M., Matta, E. M., Elkhishen,

March). Comparison of Parallel and Serial Execution of Shortest Path Algorithms. In 2023 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE) (pp. 232-237). IEEE.

[14] Ch, S., Panda, S. K., & Geetha, V. (2023, December). Strategies for Network Traffic Optimization: An In-Depth Analysis of Multipath Routing Algorithms in SDN. In 2023 3rd International Conference on Mobile Networks and Wireless Communications (ICMNWC) (pp. 1-8). IEEE.

[15] Deaconu, A. M., Spridon, D. E., & Ciupala, L. (2023, July). Finding minimum loss path in big networks. In 2023 22nd International Symposium on Parallel and Distributed Computing (ISPDC) (pp. 39-44). IEEE.

[16] Sundarraj, S., Reddy, R. V. K., Babu, B. M., Lokesh, G. H., Flammini, F., & Natarajan, R. (2023). Route Planning for an Autonomous Robotic Vehicle Employing a Weight-Controlled Particle Swarm-Optimized Dijkstra Algorithm. IEEE Access.