Priority-based routing: A shortest path algorithm for e-commerce deliveries


∗Institute of Information Technology, Noakhali Science and Technology University, Bangladesh.
∗∗Cyber Center, Noakhali Science and Technology University, Bangladesh.

Abstract Since the COVID-19 pandemic, the popularity of online shopping through social media and e-commerce sites has increased in Bangladesh. Due to different delivery policies of the courier service companies and other unavoidable reasons, products often do not reach the customer on time. This paper proposes an optimized shortest-path algorithm for e-commerce companies to efficiently and quickly deliver products to multiple destinations by considering their priorities. The use of graph theory and the Google Map API provides flexibility and real-time optimization of the shortest-path calculation. The proposed approach introduces two types of demand destinations: normal and priority demanded destinations. The superiority of the proposed algorithm over the conventional delivery system is a 12%–15% increase in the delivery time and a 15%–20% decrease in overall costs. Our proposed algorithm will benefit companies dealing with product delivery systems such as courier product delivery, e-commerce product delivery, and online food delivery.

Keywords: e-commerce, product delivery system, shortest path algorithm

1. Introduction

During the last fifteen years, the rapid expansion of the internet and associated smart devices has increased e-commerce business exponentially in Bangladesh. According to a statistic, approximately 2,000 e-commerce sites and 5,000 social media (Facebook, Whatsapp, Viber, etc.) based online shops in Bangladesh deliver more than 30,000 products every day (Parvin et al., 2021). Other statistics show that in 2016, USD 50 million was invested in e-commerce businesses, 10 million of which came from foreign direct investment (Rahman, 2020).

According to the e-Commerce Association of Bangladesh (e-Cab), there are currently 7000 e-commerce sites in Bangladesh (Yasin et al., ). As the e-commerce business increases, the corresponding demand for home delivery increases rapidly. Business-to-Business (B2B), Business-to-Consumer (B2C), Consumer-to-Consumer (C2C), and Consumer-to-Business (C2B) are the four popular dimensions of e-commerce business throughout the world (Karim & Qi, 2018). Among these, Business-to-Consumer (B2C) is the most popular dimension in Bangladeshi cities (Rahman, 2023). Nowadays, the popularity of B2C is increasing in urban, semi-urban, and rural areas. Since the COVID-19 pandemic, e-commerce product delivery and related businesses have become very popular in Bangladesh, and people in remote areas are used to purchasing products through e-commerce. AJR, Janani, USB, SA paribahan, Express Noakhali, Metro Express, Paperfly, Steadfast, Sundarban paribahan, REDEX, Daraz, Shodagor express, e-courier, e-ghar, and Pathao are among the top participants in Bangladesh’s B2C distribution sector. The service provided by the courier service varies based on distance, service fee, and delivery type (emergency/ordinary). The country’s weak infrastructure, inadequate logistics support, low-speed internet, weak mobile network connections, and transportation are the notable challenges faced by Bangladesh’s e-commerce business. These challenges affect on-time product delivery. In addition, consumers in urban, semi-urban, and rural areas face longer delivery delays and higher expenses. Minimizing delivery time and cost, delivery riders want to use the most efficient and shortest path to deliver products to the consumers and return to the office shortly. However, choosing the best route for delivery is a challenge for delivery personnel because there are so many paths that can reach the consumers. Therefore, it is necessary to find the most effective way to reduce the overall cost and reach customers quickly.

Most research articles on the internet discuss e-commerce business types, dimensions, methodologies, prospects, drawbacks, and threats of product delivery in Bangladesh theoretically. Moreover, some suggestions to overcome the drawbacks are also discussed. In our observation, we found that the introduction of information technology to resolve existing e-commerce product delivery-related issues is minimal. Among these, some research works focused on solutions between two locations, but none of them proposed or designed a completely optimized route for multiple destinations with
normal and priority-based destinations. Furthermore, the shortest efficient pathfinding and cost assessment for domain-specific fields such as e-commerce product delivery is not available.

To combat these problems, we have proposed an algorithm/model that introduces information technology to minimizes on-time product delivery challenges. This algorithm/model will handle multiple delivery requests at a time, identify and prioritize emergency deliveries over regular deliveries, and suggest a single path to all delivery points. In addition, the algorithm/model guides the delivery rider to choose the shortest path that requires the least amount of time to deliver all the products. The algorithm considers factors such as traffic jams, road conditions, and distance to suggest the best possible route. With the help of information technology, the delivery rider can use a route optimization algorithm to find the most efficient path to deliver products. By doing so, customer satisfaction will increase and decrease delivery costs. If the delivery rider can reach the customer’s door directly, they don’t have to contact the customer multiple times to locate their address. By reducing the number of calls, the proposed algorithm/model can reduce call costs and increase efficiency.

For solving shortest path-related problems, some well-known algorithms include the Bellman-Ford algorithm, Dijkstra algorithm, Floyd-Warshall algorithm, graph growth algorithm, Johnson’s algorithm, and topological sort algorithm. We can use these algorithms according to our needs. Review and summarization of three algorithms: the graph growth algorithm, the Dijkstra algorithm (approximate bucket), and the Dijkstra algorithm (double bucket) (Zahan, 1997). To include the shortest path algorithm, the authors used 21 real road networks. In this article, we focus on the opportunities and threats, history and evolution, objectives, and methodologies of e-commerce business in Bangladesh (Bappy, 2018). In addition, current market conditions and opportunities, and technical, legal, and regulatory aspects are also illustrated by the authors. They also provide some recommendations for this sector.

The rules, regulations, and other associated issues of e-commerce business in Bangladesh are highlighted by the author (Hossain, 2000). The authors focused on the three dimensions of e-commerce namely Business-to-Consumer (B to C), Business-to-Government (B to G), and Business-to-Business (B to B). The author Das (2017) proposed an algorithm that can be used to find efficient paths within a short period. They used Dijkstra’s algorithm for this purpose. The authors show that the proposed algorithm can find an efficient path within 20–150 s, thereby allowing people and organizations to save time and money.

Product delivery time, service quality, price, and condition of the foods delivered the factors beyond the success of online food delivery, identified by the author (Saad, 2021). In addition, they explored the behavior of consumers toward the online food delivery business. The authors Arnold et al. (2018) have attempted to identify B2C delivery challenges and ways to overcome them. They calculate the number of deliveries performed by the delivery man, per delivery cost, and time spent per delivery on the basis of some parameters. They found a diesel van to be probably the most effective vehicle for delivery. In addition, the authors also highlighted that the absence of a consumer, and route distance are the parameters for which delivery may delay.

The authors Jain et al. (2022) have proposed an evacuation model that will suggest an optimized evacuation path during an emergency. For this purpose, authors use Dijkstra’s algorithm to find the shortest and least-consuming path. Authors Almeida et al. (2004) introduced the time-dependent travel-time data sets and discussed the time-dependent vehicle routing, data aggregation, travel time collection, and integration of time-dependent travel times in time-dependent vehicle routing models. To identify the shortest optimal route within the warehouse, a dynamic programing model can be used (Shariff et al., 2022). The authors identified the inner transportation problem in the warehouse as one of the notable reasons for late delivery. Synchronization in storage assignment, task allocation, etc. can reduce the transportation problem.

The introduction of a medical transportation system can solve the routing problems of the Medical Information Collectors (MIC) between medical facilities (Al-Turjman, 2017). In addition, using the system, the delivery of urgent medical information using the navigated routes and traveling distances minimize the total costs. The development and testing of a conceptual research model for e-retailers suggested a guideline about the customer's satisfaction, intention to purchase products repeatedly, on-time delivery, and so forth (Saha et al. 2020). The authors collected the data through an online survey in Bangladesh, and most importantly, their study showed that the sales of e-commerce products increased based on timely delivery. A proposed machine learning model/framework associated with delivery success rates and reducing delivery costs is illustrated in (Kandula et al., 2021). The proposed model predicts delivery schedules generated from two real-world datasets of a large e-commerce platform. In addition, the proposed model results indicate the effectiveness of the decision support framework by saving up to 10.2% in delivery costs compared with current industry practice. The authors of (Bergmann et al., 2020) analyzed the route efficiency trade-offs arising from combining first-mile pickup and last-mile delivery operations in an urban distribution system. They proposed a set of closed-form adjustment factors using comprehensive numerical experiments and regression analysis. It improved the existing continuous estimation-based route length estimation function by reducing the impact of urban traffic and emissions by up to 16% and reducing the cost of operating its vehicle fleet. A third-party distribution model was adopted as the logistic method as the main research subject of fruit and vegetable agricultural products in Zhuzhou City (Feng, 2020). They applied the ant colony algorithm to find the shortest path and calculated the cost using MATLAB software. The findings of the proposed algorithm improved the distribution efficiency and reduced logistic costs, thereby advancing rural e-commerce.
The authors Chen et al. (2019) presented a pricing strategy for reducing congestion in e-commerce logistics. The outcome of this study demonstrated how additional congestion fees affect stakeholders in e-commerce in terms of both early and delayed delivery. Using these pricing models, e-commerce managers encouraged customers to choose off-peak delivery times and change the volume of planned shipments to affect the interaction between home delivery businesses and e-commerce customers. This proposed pricing model contains no routing solution.

The authors Tiwari and Sharma (2023) provided several vehicle routing problems with a maximum capacity limitation but no time constraint. They put into practice and evaluated several optimization methods, including Intra-Route Local Search, Inter-Route Local Search, and Tabu Search, which offer a less-than-ideal alternative to the greedy solution to this NP-hard problem. A comparison of all algorithms reveals that while one (TABU search) may offer a somewhat more optimized solution than the others, it also consumes a disproportionately higher amount of CPU time. Effective path planning ensures delivery speed by which we can save money and bring more benefits.

The authors of Zhong et al. (2023) described the Vehicle Routing Problem (VRP). It uses Simulated Annealing (SA) for pathfinding, and distribution and obtains the optimal solution for path planning.

The authors Lin (2022) established an e-commerce logistics distribution optimization mathematical model based on the Dijkstra algorithm. In addition, this research paper combined theory and practice to study the logistics distribution of e-commerce logistics in terminal distribution. This method reduces the cost of the company. This study showed that the path optimization distance can be reduced by 10 km compared with the empirical path.

2. Materials and Methods

In this section, we describe our proposed shortest-path algorithm for efficient e-commerce delivery. For this purpose, we focused on the description of a simplified workflow of the algorithm, working procedure of the algorithm, flow chart, description of the proposed algorithm (pseudo code), time complexity analysis, data collection, and evolution of the proposed algorithm.

2.1. Workflow of the algorithm

The workflow of product delivery from the local hub to the consumer is shown in Figure 1. The e-commerce product delivery company can choose the vehicle (Pickup van, Motorcycle/Bicycle, or others) according to the policy (distance from the local hub to the customer's doorstep, cost policy, kinds of road infrastructure, availability of the vehicles, etc.). In Figure 1, visual representations of product delivery using two different vehicles are represented. If the company chooses a pickup van, then the route to successful delivery is denoted as a>c>d>f, and if missed then, a>c>d>e to return to the local hub/office/deposit. Similarly, if the company chooses motorcycle/bicycle then the route to successful delivery is denoted as b>g>h>i, and if missed then, b>g>h>j to return to the local hub/office/deposit. In case of an unsuccessful/missed delivery, the procedure is repeated until successful completion of the delivery process.

![Figure 1 Simplified workflow of the algorithm.](image)
2.2. Working procedure of the proposed algorithm

The proposed algorithm according to the principle of graph theory, where each destination is denoted as a node, and the distances between them are denoted as edges. The algorithm collects the prerequisite information like classifying the starting point, destination points, priority, and non-priority destination points, etc., and finds the most efficient path to visit all destinations.

The algorithm completes the following steps during execution:
1. First, the algorithm builds the graph structure using the customers’ location, which represent the networks of destinations and the distances between them. Here, the Google Map API calculates the travel time and distance between each pair of nodes. In addition, it finds the most efficient path to visit all destinations.
2. The next step is to classify the destinations based on priority. If a customer requests emergency delivery of a product, the request is stored in the priority destination list, whereas the rest of the requests are in the non-priority destination list. The algorithm shows the paths to priority destinations first, followed by the non-priority destinations.
3. The third step involves calculating the of travel time from the source point to each destination. This information is crucial for the algorithm to make decisions about which destination to cover first and which to follow. The Google Map API is used to collect this data and calculate the most efficient path.
4. The fourth step involves finding the nearest destination from the source point. If there are both priority and non-priority destinations, the algorithm first covers the priority destinations. The algorithm passes the source and priority destination addresses to the Google Map API in pairs, and the most efficient path is selected on the basis of travel time. The nearest destination is denoted as Destination1, which is then marked as visited to avoid considering it as a destination again.
5. The fifth step involves finding the next nearest destination. The algorithm uses Destination1 as the source point and again calls the Google Map API to provide travel time from Destination1 to the remaining priority destinations. The algorithm then compares all path costs between Destination1 and priority destinations and stores the most efficient path, travel time, and distance. This path is appended to the previously stored path in the Path variable. The next nearest destination is denoted as Destination2 and marked as visited.

The algorithm repeats the fifth step until all destinations have been visited. Priority destinations are visited first, followed by non-priority destinations. After visiting all destinations, the final path that covers all priority destinations first and then all non-priority destinations are found. The total time and distance of the path are calculated and shown on Google Maps for the delivery person to follow.

![Flow Chart of the proposed algorithm](https://www.malque.pub/ojs/index.php/msj)
2.3. The Proposed algorithm to shortest-path calculation

Algorithm steps:

Step 1: Input:
1.1: Input initialSource node.
1.2: Input priority destinations node and insert into priority list.
1.3: Input normal destinations node and insert into normal list.

Step 2: Take a variable nearestDestination: = source.

Step 3: for (i=1 to all nodes) cost[i]: = infinity, parentOf[i]: = 0

Step 4: Take a min heap PQ can contain pair of data (Cost, Node) and push nearestDestination with cost 0.

Step 5: Extract a node from PQ, parentNode:= Extracted node

Step 6: Select a neighbor of parentNode.
6.1: If (cost[parentNode]+Cost of parentNode to neighbor) < cost[neighbor])
6.1.2: parentOf[neighbor] = parentNode
6.1.3: Insert updated Neighbour cost and Neighbour node as a pair into PQ

Step 7: Continue from Step 5 until PQ is not empty.

Step 8: Search unvisited priority destination with minimum cost and assign it to nearestDestination.

Step 9: Find path from nearestDestination to Source through ParentOf array.
9.1: u< nearestDestination
9.2: Path <- u // Here Path is a set contains all nodes constructing path from source to destination
9.3: u = parentOf[u]
9.4: continue from 9.2 until parentOf[u]! =0.

Step 10: Continue from Step 3 until all priority destinations are not visited.

Step 11: Search unvisited normal destination with minimum cost and assign it to nearestDestination.

Step 12: Do Step: 3-10 for normal destination list.

Step 13: Collect path from recently calculated nearestDestination to initialSource.

Step 14: Merge all segmented paths generated in step: 9 into a single path.

Step 15: Find the minimum edge from this single path and increase its value by one.

Step 16: Continue from step: 2 certain times to generate some efficient paths.

Step 17: End.

2.4. Time complexity analysis

Here, we consider that there can be multiple destinations and maybe two kinds of situations. (1) If the number of destinations is 1 (A), then Dijkstra needs to run two times to run to generate a path from Source to A, and A to Source. (2) If the number of destinations is 2 (A, B) and A is comparatively nearer than B, then we have to travel to A first. Then we have to travel from A to B and then from B to the source. Here, we need three segments of the path Source to A, A to B, and B to Source. Therefore, the process requires three times Dijkstra. Therefore, if the number of destinations is n combining priority and normal destinations, then we have to run the Dijkstra algorithm for n+1 times. Therefore, Complexity comes to O(n*ELogV) when generating a single shortest path. However, we have made additional modifications to find more efficient paths. We run the algorithm for certain times to generate some efficient paths. Moreover, we have determined the shortest path and increased the minimum edge of the path by one so that in the next turn, the algorithm will try to avoid the minimum edge. As a result, the second shortest path is generated. These specific times can be 100/500/1000, and it is constant. So, it will not change the time complexity. So, we can say, the overall time complexity is O(N*ELogV).

For this same scenario, the time complexity for the DFS algorithm will be O(N*VV), which is too bad. For example, we consider a map of a large city with nine destinations and 10 possible paths between each pair of nodes. Then, total paths will be generated 9^10. This is not a good process. In the case of Floyd-Warshall, which requires the same criteria is O(N*V3). It is still greater than our proposed algorithm complexity O(N*ELogV).

2.5. Data collection

This is a crucial part of this research. For this purpose, we prepared a questionnaire, as shown in Table 1. The questionnaire consists of nine questions. We visited the local hub offices of Daraz, Steadfast, e-Courier, Express Noakhali, Fox Parcel, Shodagor Express, REDEX, Paper Fly, Metro Express, and Pathao in Noakhali, Bangladesh. We asked the office manager to provide the data according to the questionnaire. Apart from the questionnaire list, we collected some more related data. As most of these data are business-related data and are directly associated with the business, we have to omit that. The questionnaire is as follows:
Table 1 Questionnaire.

<table>
<thead>
<tr>
<th>No</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>What is the name of your company?</td>
</tr>
<tr>
<td>2</td>
<td>What kind of company is it?</td>
</tr>
<tr>
<td>3</td>
<td>How many deliveries/landings (on average) do you have in a day from the central hub?</td>
</tr>
<tr>
<td>4</td>
<td>What kind of delivery vehicles are used by your company?</td>
</tr>
<tr>
<td>5</td>
<td>Do you use any cluster method for product delivery?</td>
</tr>
<tr>
<td>6</td>
<td>How delivery riders communicate with the customers?</td>
</tr>
<tr>
<td>7</td>
<td>How many products (on average) are delivered in a day?</td>
</tr>
<tr>
<td>8</td>
<td>How many products (on average) are not delivered/unsuccessful/missed in a day?</td>
</tr>
<tr>
<td>9</td>
<td>Do you have any software for delivery riders to find the shortest path from local hub to Customer locations?</td>
</tr>
</tbody>
</table>

Based on the provided data, we prepared the dataset shown in Table 2. The dataset contains the data of the month December-January, 2022. From this dataset, we can easily calculate the cost of delivery of successful and unsuccessful/missed products.

2.6. Evolution of our approach

Starting address: From this point, a delivery rider starts his trip to deliver products to all destinations, and then returns to this starting address. The starting addresses may be the warehouse or food store from where a rider collects products to deliver.

Different destination addresses: Here, the application requires all destination addresses. When a customer requests a product using a particular e-commerce app, the customer’s address is considered as a destination.

Priority destination selection: If a delivery rider receives an emergency delivery request, then this address is considered a priority destination. First, the application shows the path to the priority destinations and then shows the path to reach the non-priority destinations.

Travel time: Our application operates based on the travel time amongst the destination points. As we are using Google Map API, the application will get travel time information from Google Map API.

Table 2 Details of dataset (December–January, 2022)

<table>
<thead>
<tr>
<th>Company Name</th>
<th>Total Product Landing (Central hub to local hub)</th>
<th>Successful Delivery</th>
<th>Missed delivery</th>
<th>No of delivery riders</th>
<th>Clustering</th>
<th>Communication (Mobile and SMS)</th>
<th>Delivery Vehicle (Motorcycle/Pickup Van)</th>
<th>Availability of software for delivery riders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daraz</td>
<td>737</td>
<td>447</td>
<td>290</td>
<td>32</td>
<td>N/A</td>
<td>1400+</td>
<td>Motorcycle</td>
<td>No</td>
</tr>
<tr>
<td>Steadfast</td>
<td>580</td>
<td>402</td>
<td>178</td>
<td>20</td>
<td>N/A</td>
<td>1000+</td>
<td>Motorcycle</td>
<td>No</td>
</tr>
<tr>
<td>Pathao</td>
<td>436</td>
<td>287</td>
<td>149</td>
<td>16</td>
<td>N/A</td>
<td>800+</td>
<td>Motorcycle</td>
<td>No</td>
</tr>
<tr>
<td>e-courier</td>
<td>423</td>
<td>257</td>
<td>166</td>
<td>17</td>
<td>N/A</td>
<td>800+</td>
<td>Both</td>
<td>No</td>
</tr>
<tr>
<td>Express-Noakhali</td>
<td>289</td>
<td>167</td>
<td>122</td>
<td>10</td>
<td>N/A</td>
<td>550+</td>
<td>Motorcycle</td>
<td>No</td>
</tr>
<tr>
<td>Shodagor Express</td>
<td>186</td>
<td>97</td>
<td>89</td>
<td>6</td>
<td>N/A</td>
<td>300+</td>
<td>Motorcycle</td>
<td>No</td>
</tr>
<tr>
<td>Fox Parcel</td>
<td>135</td>
<td>89</td>
<td>46</td>
<td>5</td>
<td>N/A</td>
<td>250+</td>
<td>Motorcycle</td>
<td>No</td>
</tr>
<tr>
<td>Metro Express</td>
<td>203</td>
<td>119</td>
<td>84</td>
<td>9</td>
<td>N/A</td>
<td>400+</td>
<td>Motorcycle</td>
<td>No</td>
</tr>
<tr>
<td>Paperfly</td>
<td>226</td>
<td>137</td>
<td>89</td>
<td>8</td>
<td>N/A</td>
<td>400+</td>
<td>Motorcycle</td>
<td>No</td>
</tr>
<tr>
<td>Redex</td>
<td>420</td>
<td>280</td>
<td>140</td>
<td>16</td>
<td>N/A</td>
<td>800+</td>
<td>Motorcycle</td>
<td>No</td>
</tr>
</tbody>
</table>

After collecting this information, the application starts to run and generates an efficient path to cover all destinations within the shortest time. The final output contains the following information:

The efficient path: The path connects all the destinations in such a way so that a delivery man can follow the route and can return in a short time. This part starts from the source point. It then connects to the nearest priority destination. Then the path connects to the nearest priority destination. After all the priority destinations are connected then the path...
connects to the next nearest non-priority destination. After finishing connecting all priority and non-priority destinations, the path connects to the starting point. Using this path, a delivery man can start his trip to deliver all the products and can easily return to the source point in a short time.

For a better understanding of our proposed algorithm/model, we consider two different scenarios to find the shortest-path. We denoted the two scenarios as Scenario 1 and 2.

Scenario 1: Here, the map for the Sonapur region is shown in Figure 3. The red color is used for priority-based destinations, the green color is used for normal destinations, the blue color is used for the source place, the red arrows are used for the path of the source to priority-based destinations, and the green arrows are used for the path of normal destinations to the source.

Suppose, there is an e-commerce company XYZ in Sonapur. We considered 21 nodes for different locations and assumed that Sonapur is the central node (starting address) from which the e-commerce company serves its products to different destinations.

We have drawn a case where the company needs to deliver its products to 4 customers located at Motipur, Kazirtek, Thakkar, and Dotterhat. Let us say that the customers from Kazirtek and Motipur want their products delivered as soon as possible, although it costs an extra fee. The other two destinations, Thakkar and Dotterhat have normal delivery, which means that the customers from these places have no conditions for fast delivery. Now, a delivery rider wants to deliver their products efficiently so that he can return to his office soon. Here in Sonapur, he has many choices for delivering products to customers. But which will be the best? Let us take three different paths and calculate their distances.

**Figure 3** Conceptual map of Sonapur region.

Path One: Sonapur-Mohabbatpur-Motiput-Mohabatpur-Kalitara Bazar-Doctor bazar-Kazirtek-Subarna Agro-Thakkar-Mohabatpur-Kalitara-Lalpur-Binodpur-Dotterhat-Uttar Sonapur-Sonapur. Total distance for Path One: $1 + 1 + 1 + 1.5 + 1 + 2 + 3.5 + 1.5 + 1.5 + 1.5 + 3 + 1 + 1.5 + 1 + 1 = 23$ km.

Path Two: Sonapur-Mohabbatpur-Motiput-Chor elahi-Thakkar-Subarna Agro-Kazirtek-Doctor bazar-Kalitara-Lalpur-Binodpur-Dotterhat-Uttar Sonapur-Sonapur. Total distance for Path Two: $1 + 1 + 2 + 2 + 1.5 + 3.5 + 2 + 1 + 3 + 1 + 1.5 + 1 + 1 = 21.5$ km.

Path Three: Sonapur-Mohabbatpur-Motiput-Mohabatpur-Kalitara Bazar-Doctor bazar-Kazirtek-Subarna Agro-Thakkar-Mohabatpur-Sonapur-Uttar Sonapur-Dotter Hat-Uttar Sonapur-Sonapur. Total distance for Path 3: $1 + 1 + 1 + 1.5 + 1 + 2 + 3.5 + 1.5 + 1.5 + 1 + 1 + 1 + 1 = 19$ km.

Here Path-3 is optimal which is shown in figure 3. Priority-based destinations Motipur and Kazirtek are delivered first and then Thakkar and Dotterhat are delivered.

Scenario 2: Suppose there is an e-commerce company XYZ in Bosur Haat. In Figure 4, we have taken 16 nodes for different locations. We assumed that Bosur Haat is the central node from which the e-commerce company serves its...
products to different destinations. We have drawn a case in which the company should deliver its products to three different destinations: Shantir Haat, Nobipur, and Kabir Haat. Suppose, Shantir Haat and Nobipur are priority-based destinations and products should be delivered there immediately. The other destination, Kabir Haat, is normal delivery, and there is no condition for quick delivery. Now, a delivery rider wants to deliver the products efficiently so that he can return to his office soon. Here from Bosur Haat, he has many choices to deliver products to customers through many paths. But which will be the best? Let us take three paths and calculate their distances.

Path One: Bosur Haat-Shantir Haat-Bosur Haat-Bhuiyan Bazar-Nobipur-Bhuiyan Bazar-Kabir Haat-Chaptrashir Haat-Bosur Haat. Total distance for Path One: $2.5 + 2.5 + 5.5 + 4.5 + 4.5 + 4 + 5.5 + 5 = 39$ km.

Path Two: Bosur Haat-Shantir Haat-Bosur Haat-Bhuiyan Bazar-Nobipur-Bhuiyan Bazar-Kabir Haat-Bhuiyan Bazar-Bosur Haat. Total distance for Path Two: $2.5 + 2.5 + 5.5 + 4.5 + 4.5 + 4 + 4 + 5.5 = 33$ km.

Path Three: Bosur Haat-Shantir Haat-Bosur Haat-Yakubpur-Nobipur-Bhuiyan Bazar-Kabir Haat-Bhuiyan Bazar-Bosur Haat. Total distance for Path Three: $2.5 + 2.5 + 4 + 4.5 + 4 + 4 + 5.5 = 31$ km.

Here Path three is optimal which is shown in Figure 4. Priority-based destinations Shantir Haat and Nobipur are delivered first and then Kabir Haat is delivered.

![Conceptual map of Bosurhat region.](https://www.malque.pub/qjs/index.php/msj)

Visual display of the complete path and direction: The application uses Google Maps to show the path. The map shows from where the path starts and how the destinations are covered and return to the source place. A rider can follow the route on Google Maps and travel to all destinations in a short time.

Total distance covered by the path: The Application shows the distance covered by the efficient path in kilometers. Since we are using Google Maps, it helps us calculate the distance of the path.

Total time needs to travel: The Application shows the amount of time required to travel the path at that moment. The Google Map API gives the travel time information.

3. Results and Discussion

The total cost of the delivery may differ from courier to courier based on the service policy. In our proposed algorithm/model we suggest some guidelines to follow. Before starting daily product delivery activities, some prerequisite tasks need to be completed. For each delivery, the customer is contacted and the rider leaves with the product after ensuring the customer's presence. After confirming the presence of the customer, the delivery location is locked and prioritized according to the instructions. Moreover, if the customer is unable to receive the product, the product goes to the pending list and waits for next-day delivery. In this way, we can reduce fuel costs.
For a more specific illustration, we consider two cases, the ideal case, and the unusual case. In an ideal case, the total cost of delivery (per unit) is the combination of communication cost (Manpower cost per hour + Fuel cost per KM), SMS charge (we consider 0.22 BDT per SMS), and phone call fee (we consider 0.80 BDT per minute). Here, we consider the distance from the local hub to the customer. The equation used to calculate the total cost in an ideal situation is,

\[ C_{\text{Total}} = (C+S+P) + k \]  

(1)

where,
- \( C_{\text{Total}} \) = the total cost,
- \( C \) = the communication cost,
- \( S \) = the SMS fee,
- \( P \) = the phone call fee,
- \( k \) = constant for disorganized routes.

In an unusual case, the process of the ideal case iterates several times which increases the total delivery cost. Here we consider some cases as follows: 1. Choose an unorganized path, 2. Unpicked phone calls by the recipient, 3. Unavailability of the recipient, 4. Not receiving the message. So, the equation to calculate the total cost in unusual situation is,

\[ C_{\text{m-Total}} = 2(C+S+P) + k \]  

(2)

where,
- \( C_{\text{m-Total}} \) = the total cost for missed delivery,
- \( C \) = the communication cost,
- \( S \) = the SMS fee,
- \( P \) = the phone call fee,
- \( k \) = constant for disorganized routes.

Here, we calculate the total cost of a product delivery from the local hub to the customer's doorstep. Suppose, the delivery rider must work for 8 hours a day and the salary is 9,000 tk per month. The average mileage of the motorcycle is 45 km/liter. The price of the petrol/liter is 130tk. We assume that for a successful delivery from the local hub to the customer's doorstep, it requires at least 3km and is 130tk. Therefore, in an ideal case, the total cost for a successful delivery is (18.75 tk/ per half an hour + (2.9 * 3 km) tk + 0.22 tk + (0.80* 2 calls) tk) = 29.27 tk. If the number of deliveries increases, the reduced total cost also increases. Again, for Scenario 2, we calculate the total cost. Here for three different deliveries, the rider must travel three different paths. For path one, the rider traveled 39 km, for path two 33 km, and for three 31 km. So, the total cost for paths one, two and three will be 39Km*29.27 tk=767.31 tk, 33 km *29.27 tk=965.91 tk, and 31*29.27 tk=907.37 tk. Our proposed algorithm suggests the three different paths and by using it we can reduce the total distance (39-31) =8 km and reduce the total cost of (767.31-907.37) =234.16 tk. If the number of deliveries increases, the reduced total cost also increases.

Thus, we can formulate the equation of Total Cost Reduction \( R_s \) = Delivery Cost before using algorithm (\( T_s \)) - Delivery Cost after using algorithm (\( T_a \)). We can see that, before using the proposed algorithm in Scenario 1, the product delivery cost is, \( T_s =673.21 \) tk and after using the proposed algorithm the delivery cost is \( T_a = 556.13 \) tk. Total cost reduction is, \( R_s = (673.21-556.13) \) tk = 117.08 tk. For scenario 2, the product delivery cost is, \( T_s =1141.53 \) tk and after using the proposed algorithm the delivery cost is \( T_a =907.37 \) tk. Total cost reduction is, \( R_s = (1141.53-907.37) \) tk = 234.16 tk. We calculate the percentage of reduced cost amount \( TR_p \) by dividing the reduced cost amount \( R_s \) by 100 by the total cost of the products before using the proposed algorithm \( T_c \).

\[ TR_p = (R_s*100)/T_c \]  

(3)

where,
- \( TR_p \) = the percentage of reduced cost amount,
- \( R_s \) = the reduced amount of cost,
- \( T_c \) = the total cost of the products before using the proposed algorithm.

For scenario 1, by using equation 1 we get the \( TR_p \) 17.40%, and for scenario 2 the \( TR_p \) is 20.52%. In both cases we get the percentage of reduced cost amount in between 15 to 20% (approximately).
In addition, conventional shortest path-finding algorithms do not consider the priority of the destination paths. In our proposed algorithm, we suggest categorizing the priority and non-priority conditions based on user’s choice.

5. Conclusions

The main goal of the current study was to propose an algorithm that will be helpful in determining an efficient path whenever we need to deliver products to multiple destinations based on priority. The evidences of the study indicate that our proposed algorithm can suggest the priority based destinations and shortest distances among those to reach the destinations quickly. The main findings can be summarized as follows: (1) The proposed algorithm can suggest multiple efficient paths, and the delivery rider can choose any path among those to deliver the products. (2) By using the efficient path, the delivery rider can deliver products quickly and minimize the overall cost over conventional delivery systems. As an extension to this work, we will design a location-based clustering system for third-party courier companies so that they can automatically assign their delivery riders for specific clusters. In addition, we will provide apps to the delivery riders for finding the shortest path to deliver products that will help to reduce the number of delivery riders to reach the customer to their previously mentioned location. Finally, we also try to propose a third-party outsourcing delivery riders’ model by which it will be possible to deliver the products of multiple courier service companies through a single delivery rider within a single cluster.

Acknowledgment

We acknowledge the managers/local hub in charge of the e-commerce product delivery organizations namely Redex, Daraz, Steadfast, Pathao, e-courier, Express Noakhali, Shodagor Express, Fox Parcel, Metro Express, and Paper fly for providing some valuable data and answers of the questionnaire accordingly. We would also like to thank the honorable reviewers for their guidance, editing, proofreading, and valuable suggestions for making the paper enriched.

Ethical considerations

Not Applicable.

Conflict of Interest

The authors declare no conflicts of interest.

Funding

This research did not receive any financial support.

References


