Calculating the Total Distance for a Rider on a Task

Group 5

Muhammad Ahmed Ehtisham	24110167
Muhammad Faseeh Bilal	24110341
Khuzema Usman	24110016
Malaika Zaka	25110262

Swyft

Table of Contents

Abstract

This research project is aimed at route optimization of a last-mile delivery service provider, Swyft Logistics. Swyft is a renowned logistics provider in Pakistan that requires a more efficient method to create routes for their service. The project, after analysing datasets, delves into the different methods that can potentially be used to optimise these routes and create a robust and dynamic procedure that calculates distances and optimizes the routes for individual riders and their assigned deliveries. The report explores sequential routing, which is analysed for its simplicity and quick calculations, while the TSP method is observed for its more efficient results. The project aims to reduce time and cost for Swyft logistics routing needs in order to increase customer satisfaction, and overall efficiency and provide a dynamic solution to the long-standing issue of last-mile delivery.

Introduction

Swyft Logistics is one of the new uprising companies in Pakistan that is integrating technology into its daily operations to change the last mile delivery landscape in Pakistan. By building a new standard and bridging the gap between the consumers and the service providers, this company is able to provide a more dependable service, at a better cost, and at a faster rate. Operating in more than 300 cities within Pakistan, and serving key industries such as electronics, and fashion etc., Swyft Logistics has been able to create a strong presence in the major cities of this country.

The company has been able to create a diverse client portfolio which includes international brand names like Conatural and CAT which is a testimony of its dedication towards providing excellent service to its consumers. The firm is trying to streamline all of their logistical operations for every business at every level to make their portfolio even more diverse. Pick up, scanning, collection of parcels, same day deliveries, and robust digital systems are some of the key offerings and with the smooth integration of technology, ecommerce platforms, and their ERP systems, Swyft logistics is gaining market share. Another factor that puts this company apart from the others logistical service providers, is its transparent delivery package tracking system which allows seamless and thorough communication between the customer and the vendor in real time allowing for better overall customer experience.

Problem Overview

Over the years logistical services have changed and it was not until recently that they have tried to prioritize optimization and efficiency. An inevitable issue that has been faced by the other logistical companies as well as Swyft logistics is to optimize the complex and intricate route vile reducing the overall distance in their delivery operations. It is crucial that these routes are optimized as they are over 20,000 different deliveries happening each month only in the main cities of Pakistan. The company's current objective is to find the optimal routes for each livery along with the shortest distance for every order and integrate this twofold objective into a route planning strategy.

The primary objective would be to develop a system where the location data for every unique order can be automatically processed, which facilitates the calculation of the distances for the delivery routes. Minimizing the time and cost involved in the delivery processes are the two goals that can be achieved through this algorithm. One of the solutions to this problem could be done by using the principles of the Traveling Salesman Problem (TSP). TSP would allow the company to streamline its planning and cluster creation, reducing the burden of the operations on the riders and ensuring more timely deliveries.

By achieving these objectives, Swyft Logistics will be able to set a completely new standard in Pakistan's logistics industry, allowing the company to become one of the leaders in the technology-driven logistics market.

Literature Review

In the field of operational research, logistics, and computer science, the Traveling Salesman Problem (TSP) has been one of the biggest demic and practical interests for years. The main function of TSP is to find the shortest route that visits a set of predetermined locations only once and returns to its origin. The application of TSP and its relation to real-world scenarios and challenges, which range from logistics to microchip fabrication, has been comprehensively overviewed by Bellmore and Malone(Bellmore & Malone, 1974). The TSP problem is categorized as NP-hard by Karp, implying that the difficulty and time required to solve this problem would increase exponentially in the number of locations or cities, making achieving absolute solutions For larger datasets unachievable (Karp, 1972).

Several algorithms have been developed over the years to provide optimal or at least near-optimal solutions within a reasonable amount of time; one approach is the Lin-Kernighan heuristic, which is still one of the most used and effective methods for larger datasets of TSP, which was also demonstrated by Helsgaun through revisions and enhancements which allowed to improve its efficiency (Helsgaun, 2000). The concept of the TSP library was a recent one, providing huge storage for this problem, which has been significant in the enhancement and comparisons of different TSP algorithms (Reinelt, 1994).

Moreover, LaPorte explored the use of TSP in vehicle routing problems. This was a crucial step in logistical operations all over the world, and it showed the significance of efficient routing planning and minimizing operational costs (Laporte, 1992). As a result of the previous contributions the more recent innovations in the TSP algorithms were revised and a more modern perspective on the routing problems and its issues was provided by Applegate ET al. (Applegate, Bixby, Chvátal, & Cook, 2006).

Due to the technological enhancements happening daily such as the drone delivery systems set up by Amazon, advancements in the TSP algorithms, and the new optimization strategies mix the field of TSP relatively unexplored.

Methodology

Comparative Overview of Sequential and TSP Routing Methods

Two routing methods, sequential routing, and TSP routing, have been used to optimize the last-mile delivery for Swyft logistics. The reason that these two methods were selected was because of their characteristics and their ability to facilitate the specific challenges in logistics. Sequential routing offers a linear approach and is known for its simplicity. It takes a straightforward path from one delivery point to the next, which makes it less technologically intensive and more facilitative towards smaller datasets or deliveries that are already organized in a straightforward sequence. TSP routing, on the other hand, is known for its tech-intensive approach to solving complex routing issues. It seeks the most optimal route that will be able to visit all the locations specified and return to the origin. This would be particularly advantageous when reducing travel distance, allowing the optimization of time and operational costs for larger data sets. The two methods that have been used in this project will provide a better understanding of the limitations and their individual characteristics, allowing Swyft Logistics to make better decisions in their route optimizations.

Dataset:

The extensive dataset (LHE&ISB OCT.xlsx) provided by Swyft Logistics included several columns:

- Date Created at
- Pickup Date
- CN (Consignee Number)
- Vendor Order Id
- Vendor Name
- Consignee First Name
- Consignee Last name
- Address
- Origin City

- Destination City
- Destination Sub-City
- COD
- Weight
- Description of Product
- Current Status
- Current Status Verified
- Pickup Rider
- Zone
- Area Type
- Attempts
- City Date Code
- Expense

Data Preparation for Routing Analysis

To achieve the goal of route optimization, the first step would be to filter out the huge data set that has been provided by the company into manageable sizes. The primary focus was on data related to successfully delivered orders, and the targeted sample size was 1000 rows from a specific region, Islamabad, in this case, to ensure relevance and manageability. This filtering was done through the application of setting the conditions to the current status column and destination city column while isolating only the delivered orders in Islamabad.

The next step would be to geocode the data or, in other words, translate the textual addresses into geographic coordinates in the form of longitudes and latitudes. This step is crucial for any routing method, as the data in the text would be converted into a format that could be analyzed. Hence, the Google Maps API was used to analyze the addresses to produce their respective coordinates. This allows us to connect the gap between the raw data and usable geographic information, which would be required in future routing calculations.

During this process, the code would continuously check and handle any NA values or anomalies in the geocoded data, which would allow the routing algorithms to operate on accurate and complete spatial data.

Sequential Method:

The sequential approach that is used in the code allows for a clear an organized structure which would make it easier to understand the flow and the different parts of the code, the method is straightforward and follows a linear motion from one point to another. It is less computationally intensive, which makes it easier to apply and manage. However, it is more suitable for smaller datasets or when the deliveries are naturally organized in a linear way.

The code would be further used to geocode the filtered data, which would allow us to obtain the latitude and the longitude for the destinations. The missing values in the new geocoded data would be checked, and the data frame, which would be given as the output, would be printed. The warehouse coordinates would be defined using the coordinates that would be previously generated, allowing the creation of a distance matrix for a group of deliveries. Using this, the code can calculate the distance matrices along with the routes for each individual rider, assuming that the visits are all sequential.

The output, which would be the total distances for each rider, would be stored in a data frame, which would allow the code to calculate the collective distances for all the riders while discarding the NA values.

The data frame is then able to store all the routes of the rider sequentially in the code, providing the calculated routes for each rider, but it's right in the name and the total distance that it has travelled. The unique rider code, name, and total distances would be added to each row, and this output would be combined into a single data frame.

TSP Application:

To calculate the shortest distance of each individual rider, the TSP package in R studio would be used. This package allows the creation and manipulation of TSP instances, calculating the distance between two points and employing various algorithms such as the Haversine distance formula.

With the continuation of data processing, the TSP package is introduced using an inbuilt function, the calc_tsp_route, which would Take a group of deliveries and warehouse coordinates as the input and further filter the rules with any missing values in the latitude or longitudes. The warehouse and the delivery coordinates would be combined into a distance matrix using the system function. Finally, the TSP is solved using the solve_TSP function and the order of the location from which the TSP is extracted. The ordered coordinates that would be provided as the output would represent the optimal route.

Rider_Code, Rider_Name, and Total_Distance are the new columns that would be added to the data frame “all_routes” to store information about every unique writer's delivery route. A new data frame called total_distances Is made, which would store the total distances travelled by each rider. These new columns were in hands the data frame structured, which would allow for more comprehensive storage and a deeper analysis of the optimized routes.

Results

In our analysis of Swyft Logistics' data, we compared two different methods of planning delivery routes: Sequential Routing and the Traveling Salesman Problem (TSP) Routing. Each method yielded different results and had different pros and cons associated with them.

With Sequential Routing, we set up the routes in a straight line, going from one delivery point to the next in the order they appear in the data. For example, a delivery route starting from the Rawalpindi warehouse would go to each address one after the other. This method is straightforward and does not require as much computing power, which is beneficial for areas with less dense delivery networks. However, this can lead to longer total overall distances. For instance, in Islamabad, a delivery rider had to cover around 400 kilometres using this method.

For TSP Routing, we used the built-in TSP package in R-Script that arranges the delivery points to prioritize minimization of the total distance travelled. Using TSP for the same Islamabad route reduced the distance to about 350 kilometres – which is a 12.5% reduction. This method works well in highly dense areas with many delivery points, as it makes the route much more efficient, even though it takes longer to yield results because it requires more computing power.

In less dense clusters with fewer deliveries, the benefits of TSP aren't as noticeable. So, the choice between Sequential and TSP Routing should depend on how many delivery points there are and the layout of the delivery network.

Conclusively, the results from the application of both methods highlighted the trade-offs between computational ease and route efficiency. While Sequential Routing offered ease of use and lower computational requirements, TSP Routing delivered significant improvements in reducing travel distances which ultimately brought us closer to our goal of time and cost minimization.

Limitations

Inaccurate Coordinates:

In developing our algorithm, we encountered certain unavoidable limitations. One significant challenge was the unreliability of coordinates obtained via the Google Location API in Pakistan. The issue stems mainly from the inconsistent address formats prevalent in the region, which the Google API struggles to interpret accurately. For instance:

- Addresses like "House number 1543dhoke gangal opposite iesco sub division office rwp" don't provide clear and accurate location information.

- Some addresses mix house numbers, flat numbers, and street numbers in a way that confuses the API, such as "Area: E11/2 house number: Flat number 404, Second Hills Plaza street number: 88 Mashoor jagah:Islamabad Hospital , Islamabad."
- Special characters and distorted text strings in addresses further complicate the problem.
- Incomplete address formats like "askari 14 Sector B Street 8 Hno306" also contribute to unreliable coordinates from Google.

When used as inputs in our model, these inaccurate coordinates compromise the results' reliability. Given that our project's primary objective is to calculate the total distance covered by riders based on location coordinates, this posed a significant challenge. To address this, we utilised coordinates determined by another team through their algorithm. However, upon running our program and calculating distances using these coordinates, we discovered inaccuracies in the coordinates themselves, resulting in miscalculated distances. We received distances in meters which when converted to Kilometers was showing an average of 200-400 kilometers per rider which is not possible given the size of the city and the distance of different city hubs from the warehouse location.

Nearest Neighbor (KNN) Algorithm Failure:

An unfortunate extension of this problem was its effect on our ability to code an algorithm that would calculate distances based on the Nearest Neighbor (KNN) method. The Nearest Neighbor method calculates distances based on the nearest destination depending on where the rider is currently. For example, if the rider starts at the warehouse, the algorithm must look at the location(a) closest to the warehouse and then take location(a) as a reference point and do the same. This continues until all data points are organised in such a manner. This is one of the methods of solving the Travelling Salesman Problem. However, due to the inconsistent format of the addresses, our attempt at coding the KNN failed as there were constant errors when the algorithm tried to calculate the distance from one coordinate to another. Hence, the general TSP package was used to execute the command.

Computation Time:

Another limitation we faced in our algorithm concerns the computation time of the TSP package in R script. Although TSP offers more efficient routing with shorter distances compared to sequential routing, it demands significant computational resources. This results in an extended processing time, which is both time-consuming and labour-intensive. For example, running our algorithm on a sample size of 1,000 addresses took about 20 minutes to complete. This would increase exponentially with larger datasets, such as over 20,000 addresses. If we were to apply this process on an annual scale, the computation time would grow even further. This is a crucial consideration in our model, particularly in scenarios where time efficiency is essential. It highlights the need to balance optimal route calculation and practical processing time in our algorithmic design.

Real-World Variables and Euclidean Distances:

Another limitation of using this model is that relying solely on computed distances based on coordinates is inefficient as it doesn't account for real-world variables. This model assumes Euclidean distance between two points and calculates distances using the Euclidean formula, resulting in the shortest distance between two points. This does not apply to the real-world infrastructure and geography of the delivery networks. The shortest distance between two points will be calculated in straight lines despite the presence of a water body, mountain, or other immovable obstacle in the route, and create pathways based on those distances, which is highly impractical. This model also does not factor in the unpredictability of traffic situations, diversion routes and construction closures. A recent practical example of this was when Cavalry Road in Lahore underwent construction. This event caused considerable disruption on the roads for several days. The construction led to heavy traffic, diverted routes, and, consequently, delays in deliveries. Our algorithm, in its current form, doesn't factor in such dynamic conditions as traffic congestion or road closures, which can drastically alter the time taken to travel from one location to another and change route planning.

Recommendations:

Improving Data Collection:

To address these challenges, we propose adopting a standardized address input system, similar to the approach used by many companies. This system utilises drop-down menus for towns, cities, countries, states, and sectors, offering several advantages:

First, it ensures accuracy by enforcing consistent address formats minimising errors and typos. Second, it enhances geocoding ease as standardized addresses are readily interpreted by Google Maps API, facilitating accurate geocoding and route optimisation. Third, it enables targeted delivery as knowing the specific town and sector will allow Swyft Logistics to create delivery clusters and assign riders to specific areas, improving efficiency. To implement this solution, a phased approach is recommended in a few major cities for experimental purposes before adopting the strategy as a whole:

- Develop a comprehensive list of towns, cities, countries, states, and sectors in collaboration with local authorities and mapping services to ensure accuracy and completeness.
- Integrate user-friendly drop-down menus into the customer interface, designed for easy navigation and search.

- Offer auto-fill suggestions based on user input to streamline the process further.
- Educate and train customers on the benefits of standardised addresses and guide them through the new system.

Integration of Real-Time data:

Being constrained by time and resources, we were not able to use the “Shiny-R” package which directly links routes with live Google Map updates. Hence, using similar packages or techniques, an amalgamation of real-time data, into the exciting model, can include other factors like current traffic conditions, road closures which the model can instantly detect and then dynamically update routes based upon that to achieve higher efficiency. This can help Swyft Logistics to reduce the number of delivery attempts and cutdown on delivery expenditure.

Machine-Learning Predictions:

By using current routes, Swyft Logistics can implement machine learning models based on historical data to predict delivery demand patterns which can lead to further optimized routes, reduce overall distances, and increase operational efficiency.

Integration with Vendor ERP Systems:

By creating a seamless connection with the Vendor ERP systems, Swyft Logistics can improve its overall routing through well-informed inventory levels, restocking levels, and stockout data. This will improve the routing algorithm to inculcate inventory avalailablitiy and demand patterns, rather than only being based on delivery destinations.

Dynamic Clusering:

For further optimization, the routes and the delivery routes would have to take account of varying delivery destinations and geographic landscapes. Hence, dynamic-clustering algorithm can adjust cluster sizes based upon the spread of the delivery points which can lead to more efficient routes.

Conclusion

To conclude, our efforts to improve rider distance calculations for Swyft Logistics revealed limitations in using sequential methods alone versus incorporating TSP. We also identified issues with data input and distance calculation. These findings have led to practical recommendations that can help Swyft Logistics address last-mile delivery challenges and strengthen their position in Pakistan's logistics sector.

Works Cited

Applegate, D., Bixby, R., Chvátal, V., & Cook, W. (2006). The Traveling Salesman Problem: A Computational Study. Princeton University Press.

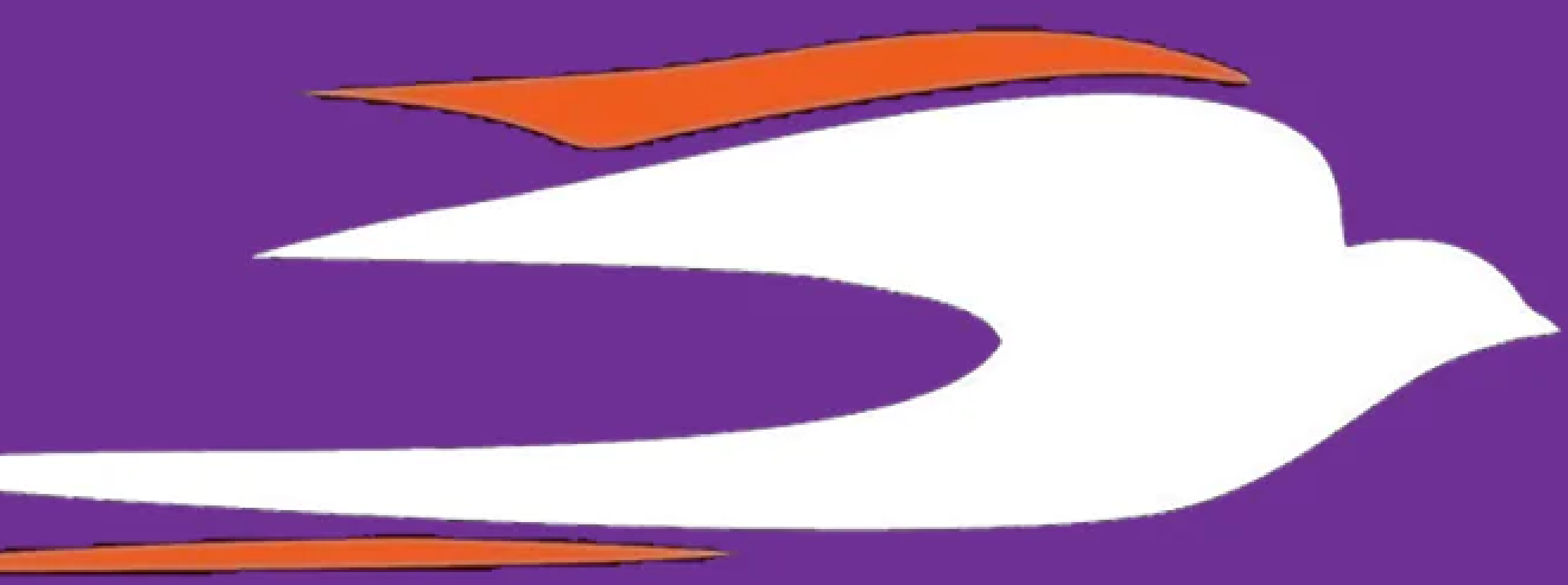
Bellmore, M., & Malone, J. C. (1974). Pathology of Traveling-Salesman Problem-Generating Algorithms. Operations Research, 22(2), 224-234.

Helsgaun, K. (2000). An Effective Implementation of the Lin–Kernighan Traveling Salesman Heuristic. European Journal of Operational Research, 126(1), 106-130.

Karp, R. M. (1972). Reducibility Among Combinatorial Problems. In R. E. Miller, J. W. Thatcher (Eds.), Complexity of Computer Computations (pp. 85-103). Plenum Press.

Laporte, G. (1992). The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms. European Journal of Operational Research, 59(3), 345-358.

Reinelt, G. (1994). The Traveling Salesman: Computational Solutions for TSP Applications. Springer-Verlag.



Appendix

First thirty rows of the dataset

		Sequential Distancing	TSP calculations
Lat	**Lon**	**Total_Distance**	**Total_Distance**
33.6232856	73.0117406	397353.5465	4444000.542
33.6021521	72.8422721	397353.5465	4444000.542
33.6206294	73.1006553	397353.5465	4444000.542
33.6242054	73.0944849	397353.5465	4444000.542
33.6274752	73.1054872	397353.5465	4444000.542
33.6174773	73.106937	397353.5465	4444000.542
33.6860876	73.0452013	8333.760214	4385603.375
33.5599421	73.1302996	48762.03313	4417625.32
33.5598254	73.1338259	48762.03313	4417625.32
33.6398237	72.9575331	234945.0461	4390639.015
33.6507704	72.9681325	234945.0461	4390639.015
33.6522357	72.9584862	234945.0461	4390639.015
33.6427005	72.960714	234945.0461	4390639.015
33.6875051	72.83727	234945.0461	4390639.015
33.7018396	72.9563912	167128.7149	4413643.956
33.7005072	72.9418242	167128.7149	4413643.956
33.7130156	73.0845099	167128.7149	4413643.956
33.6978809	72.9518863	167128.7149	4413643.956
33.7036146	72.9785353	198277.7685	4393030.334
33.7005772	72.9724431	198277.7685	4393030.334
33.7055838	72.981029	198277.7685	4393030.334
33.7036146	72.9785353	198277.7685	4393030.334
33.7012081	72.9685126	198277.7685	4393030.334
33.7015043	72.970797	198277.7685	4393030.334
33.7032546	72.9796642	198277.7685	4393030.334
33.6982291	72.9705775	198277.7685	4393030.334
33.6452045	72.9566621	66182.97223	4392171.158
33.6674797	72.9912442	66182.97223	4392171.158
33.6637144	72.9922129	66182.97223	4392171.158