Route Optimization with a Traveling Salesperson Tool in Python Language: A Case Study in a Vegetable Oil Collection Company

Eliomar de Almeida Santana[1]*, Alexandre do Nascimento Silva[2]
[1]UNEB, PPGMSB, BR 110, Km 03; Alagoinhas; Bahia; [2]UESC, DEC, Campus Soane Nazaré de Andrade; Ilhéus, Bahia, Brazil

Transportation constitutes a significant portion of organizational expenses, amounting to 60% of total costs, thereby underscoring its pivotal role in company operations. Effective transportation management is paramount for enhancing efficiency and fostering organizational growth. The primary objective of this article is to showcase the viability of utilizing free software solutions to address the traveling salesperson problem, mainly targeting companies with limited financial resources. Our method entails researching to identify and evaluate free software tools capable of addressing the traveling salesperson problem in the context of route optimization for a vegetable oil collection company. Through case studies and performance analyses, we aim to assess the effectiveness of these tools in comparison to commercial solutions. Moreover, we will consider factors such as ease of implementation and accessibility to determine the practicality of open-source tools. This study anticipates demonstrating the feasibility and efficacy of leveraging free software to solve the traveling salesperson problem, thereby providing smaller companies with a cost-effective way to optimize transportation costs. By facilitating the reduction of transport expenses and enhancing operational efficiency, these solutions are poised to bolster the growth and competitiveness of companies in the market landscape.
Keywords: Traveling Salesperson. Organization. Route.

Introduction

Logistics plays a crucial role when it comes to facilitating the growth and standing out of a company. Through efficient logistics, products can be delivered to customers optimally, resulting in reduced delivery times, lower costs, and accurate deliveries. Customer satisfaction remains a significant factor, as evidenced by reviews of companies offering such services. By carefully planning routes for collections or deliveries, companies can avoid high expenses, especially considering that transportation represents a significant portion of logistics costs, accounting for 60% of overall expenses [1]. Currently, several approaches to optimizing these routes drive the advancement of this process.

With the increasing availability of technology and information, organizations continually seek continuous improvements in their processes. It is crucial for companies with the potential to recognize irregularities in their daily operations, especially with a clear action plan. To ensure alignment with desired standards, adequate knowledge, tools to assist the team, and effective management are essential.

Given the significance of technological tools in supporting logistical tasks, this article proposes the application of the "traveling salesman" algorithm, programmed in Python, to simulate a more efficient route for deliveries. This optimized route can subsequently be applied to other routes. As an optimization method, achieving a route that minimizes distance traveled, time spent, and resources used is possible, thereby contributing to more effective and economical logistics.

According to Barreto and Ribeiro [2], the primary function of logistics is to provide the correct service in the right place, at the right time, and under the desired conditions. This logistics philosophy holds significant importance within an organization, as understanding this concept is fundamental to developing efficient work that satisfies customer needs.

Received on 20 October 2023; revised 15 December 2023.
Address for correspondence: Eliomar de Almeida Santana. BR 110, Km 03, Alagoinhas. Zipcode: 48.000.000. Alagoinhas Bahia, Brazil. E-mail: eng.eliomarsantana@gmail.com.

J Bioeng. Tech. Health 2023;6(Suppl 2):71-77

Numerous obstacles can impede the development of logistics activities, particularly in Brazil, where significant bureaucracy and deficiencies in transportation modes are prevalent. For instance, the railway system remains stagnant, and more road investment is needed, which plays a crucial role in the circulation of marketable products [3].

The logistics manager must be prepared to make various decisions, including selecting the location for installing a distribution center and determining the optimal circulation times. However, defining the most efficient routes is one of the most critical decisions. These routes significantly reduce costs, as transportation accounts for a substantial portion of expenses [4].

Understanding these aspects is essential to avoid unexpected challenges and make timely, effective strategic decisions [5].

Among logistical activities, three are the most significant within an organization: transportation, stock maintenance, and order processing. Transportation, as previously mentioned, incurs the highest expenses, making it a critical aspect of logistics. Stock maintenance is essential to prevent inventory from becoming a depreciating investment in the warehouse or resulting in stockouts, which can lead to customer dissatisfaction and lost sales to competitors. Although relatively low-cost, order processing plays a crucial role in customer service [4].

Logistics faces numerous challenges, including cost reduction and streamlining processes to ensure prompt customer delivery. Finding solutions to optimize the flow of materials and information is imperative to enhance the efficiency and effectiveness of logistics operations [5].

Good planning, technology utilization, and efficient strategies are essential for overcoming these challenges and achieving satisfactory results. Focusing on continuous improvement in logistics processes contributes significantly to the success and competitiveness of companies in the market. Creating strategic itineraries that benefit the organization without compromising customer satisfaction is a significant challenge for logistics managers. According to de Melo [6], a company's success is closely linked to providing the best service to the customer. To achieve this point, it is crucial to reduce costs, with a significant portion of these costs being attributed to transportation. Therefore, optimizing routing to minimize unnecessary expenses and shorten delivery times is imperative.

A tool that can aid in the creation of these optimal paths is the application of graph theory. As stated by de Melo [6], the concept of graphs allows for the modeling of various real-world scenarios, including computer networks, communication systems, the internet, highways, and family trees. These representations enable the attainment of desired outcomes.

Graph theory originated as a solution to a problem proposed by the Swiss mathematician Leonhard Euler (1707-1783), known as the "Königsberg bridge problem." Euler aimed to find a route around the city that would allow him to cross each bridge exactly once. Through extensive studies, it was determined that this challenge could not be completed. However, graph theory evolved, and these techniques began to find applications in engineering and chemistry [7].

As standed by Dias [7], a graph is an ordered pair (V, A), where V is any set and $A \subset u,v \subset V$. Graph theory is based on representing problems through graphical structures called graphs. These graphs consist of vertices, also known as nodes, and edges, which represent the connections between these vertices. This visual representation allows anyone to analyze and solve complex problems related to relationships and interconnections between elements.

Many methods have been developed to address specific problems in the study of graphs. One of the prominent tools in this domain is the Traveling Salesman Problem (TSP). Introduced in 1954, the TSP involves determining the optimal route for a salesperson who needs to visit a predefined set of cities and return to the starting city to minimize the total distance traveled [8].

The Traveling Salesman Problem (TSP) is a classic combinatorial optimization problem that

seeks to determine the most efficient sequence for a salesperson to visit multiple cities without repetition. This involves finding the optimal Hamiltonian path in a graph, denoted as G = (N, A), where N represents the set of cities (nodes) and A denotes the set of arcs between the cities, representing distances, time, or travel costs. The TSP finds applications in various fields, such as logistics, transportation, and vehicle routing, where activities involve traveling salespeople [5].

Python is a programming language extensively utilized in data analysis activities. Unlike languages like R, Python's codes serve a broader purpose and are not exclusive to data research activities [1]. As noted in Ballou [1], Python boasts several noteworthy characteristics, including its capability to seamlessly integrate with other languages and its highly mature library system. This enables developers to harness the capabilities of low-level languages when necessary while also taking advantage of an extensive collection of high-level libraries to expedite development and enhance productivity.

Python offers libraries that are essential for facilitating the development of PCV (Problema do Caixeiro Viajante or Traveling Salesman Problem). One such library is the Python-MIP package, recognized as a combinatorial optimization language [9]. This underscores the increasing popularity of Python usage today. Additionally, Python boasts numerous features, including powerful classes that support object-oriented programming [10]. These characteristics are crucial for optimization studies as they simplify finding solutions to problems by providing tools for problem modeling and demonstrating expected results. Consequently, Python's versatility makes it one of the primary languages for working with TSP and other optimization tasks.

Materials and Methods

A study was undertaken to apply the traveling salesman algorithm in response to the logistical challenges associated with the company's oil collection operations. The goal was to determine the most efficient route to be followed during collections at various points. The primary objective of this study is to identify the shortest route for a given itinerary. This optimized route can subsequently be replicated for other routes, aiming to minimize costs and optimize execution time.

The research focused on a route in the Bahian city of Lauro de Freitas, which is part of the Salvador metropolitan region. This study was conducted within a company specializing in vegetable oil collection. A specific route was selected to achieve the proposed objective, and a comprehensive list of collection points, along with the corresponding distances between them, was compiled. As previously mentioned, the data was adapted to suit the traveling salesman method using the Python programming language, leveraging the or-tools library provided by Google. Given that the code exclusively handles integers, the distances were multiplied by 100 to ensure the absence of decimal values. Figures 1 to 4 present the fragments of code 1 used to apply this method.

Results and Discussion

After applying the traveling salesperson algorithm to the database, which represents the distances between points multiplied by 100 (normalization), a simulation of an optimized route was obtained in Python (Figures 5 and 6).

In Figure 6, the optimized route was plotted on Google Maps based on the simulation results. A significant disparity between Figures 5 and 6 indicates substantial savings achieved through route optimization.

For instance, while the original route without any optimization technique spanned 47.50km, the simulated route using the traveling salesman method covered only 24.1km, resulting in savings of 23.4km in distance. This outcome underscores the importance of leveraging technology to drive innovation within companies, enabling them to enhance competitiveness, mainly through free software.

Figure 1. Code fragment in PYTHON - 1.

```
from ortools.constraint_solver import routing_enums_pb2
from ortools.constraint_solver import pywrapcp


def create_data_model():

    data = {}
    data["distance_matrix"] = [
        # fmt: off
       [0, 200, 420, 480, 410, 460, 460, 910, 690, 200, 200, 570, 180, 720],
    [240, 0, 210, 280, 200, 340, 340, 710, 490, 200, 200, 360, 180, 520],
    [400, 470, 0, 230, 150, 140, 140, 650, 430, 240, 190, 140, 330, 470],
    [740, 820, 450, 0, 490, 410, 410, 430, 210, 580, 580, 410, 670, 240],
    [260, 330, 200, 360, 0, 230, 230, 790, 570, 100, 100, 230, 190, 600],
    [390, 500, 110, 220, 140, 0, 0, 650, 430, 230, 170, 0, 330, 460],
    [390, 500, 110, 220, 140, 0, 0, 650, 430, 230, 170, 0, 330, 460],
    [850, 920, 560, 670, 600, 510, 510, 0, 490, 690, 690, 510, 780, 650],
    [790, 870, 510, 230, 540, 460, 460, 480, 0, 630, 630, 460, 730, 100],
    [220, 300, 260, 330, 250, 310, 310, 750, 530, 0, 70, 310, 80, 570],
    [200, 290, 210, 380, 120, 240, 240, 810, 590, 70, 0, 240, 110, 620],
    [390, 470, 160, 220, 140, 0, 0, 650, 430, 230, 170, 0, 330, 460],
    [180, 260, 340, 410, 230, 350, 350, 840, 610, 80, 110, 350, 0, 650],
    [750, 830, 650, 240, 500, 610, 610, 420, 100, 590, 590, 610, 690, 0],
        # fmt: on
    ]
    data["num_vehicles"] = 1
    data["depot"] = 0
    return data
```

Figure 2. Code fragment in PYTHON - 2.

```
def print_solution(data, manager, routing, solution):
    """Prints solution on console."""
    print(f"Objective: {solution.ObjectiveValue()}")
    max_route_distance = 0
    for vehicle_id in range(data["num_vehicles"]):
        index = routing.Start(vehicle_id)
        plan_output = f"Route for vehicle {vehicle_id}:\n"
        route_distance = 0
        while not routing.IsEnd(index):
            plan_output += f" {manager.IndexToNode(index)} -> "
            previous_index = index
            index = solution.Value(routing.NextVar(index))
            route_distance += routing.GetArcCostForVehicle(
                previous_index, index, vehicle_id
            )
        plan_output += f"{manager.IndexToNode(index)}\n"
        plan_output += f"Distance of the route: {route_distance}m\n"
        print(plan_output)
        max_route_distance = max(route_distance, max_route_distance)
    print(f"Maximum of the route distances: {max_route_distance}m")
```

Figure 3. Code fragment in PYTHON - 3.

```
def main():
    """Entry point of the program."""
    # Instantiate the data problem.
    data = create_data_model()

    # Create the routing index manager.
    manager = pywrapcp.RoutingIndexManager(
        len(data["distance_matrix"]), data["num_vehicles"], data["depot"]
    )

    # Create Routing Model.
    routing = pywrapcp.RoutingModel(manager)

    # Create and register a transit callback.
    def distance_callback(from_index, to_index):
        """Returns the distance between the two nodes."""
        # Convert from routing variable Index to distance matrix NodeIndex.
        from_node = manager.IndexToNode(from_index)
        to_node = manager.IndexToNode(to_index)
        return data["distance_matrix"][from_node][to_node]

    transit_callback_index = routing.RegisterTransitCallback(distance_callback)

    # Define cost of each arc.
    routing.SetArcCostEvaluatorOfAllVehicles(transit_callback_index)
```

Figure 4. Code fragment in PYTHON - 4.

```
    # Define cost of each arc.
    routing.SetArcCostEvaluatorOfAllVehicles(transit_callback_index)
 # Add Distance constraint.
    dimension_name = "Distance"
    routing.AddDimension(
        transit_callback_index,
        0,  # no slack
        3000,  # vehicle maximum travel distance
        True,  # start cumul to zero
        dimension_name,
    )
    distance_dimension = routing.GetDimensionOrDie(dimension_name)
    distance_dimension.SetGlobalSpanCostCoefficient(100)

    # Setting first solution heuristic.
    search_parameters = pywrapcp.DefaultRoutingSearchParameters()
    search_parameters.first_solution_strategy = (
        routing_enums_pb2.FirstSolutionStrategy.PATH_CHEAPEST_ARC
    )
 # Solve the problem.
    solution = routing.SolveWithParameters(search_parameters)

    # Print solution on console.
    if solution:
        print_solution(data, manager, routing, solution)
    else:
        print("No solution found !")
if __name__ == "__main__":
    main()
```

Figure 5. Old route.

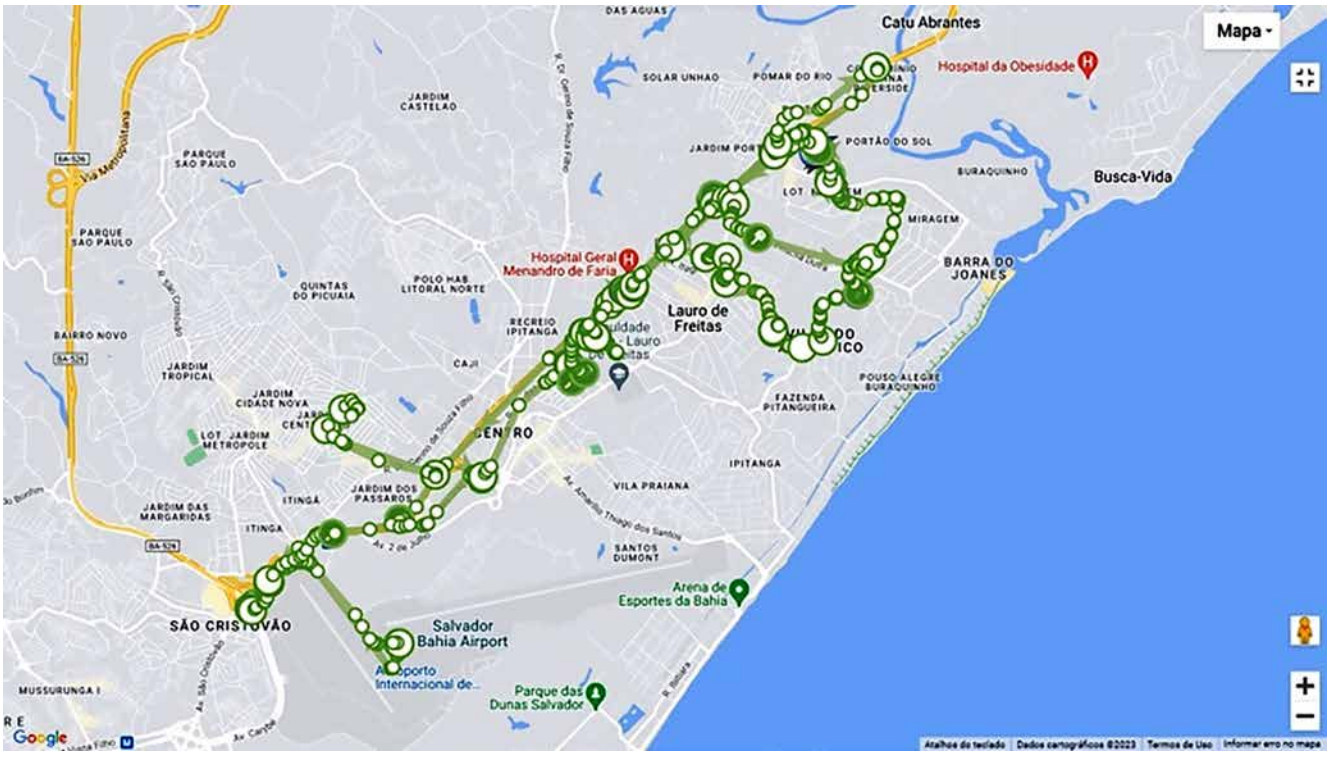
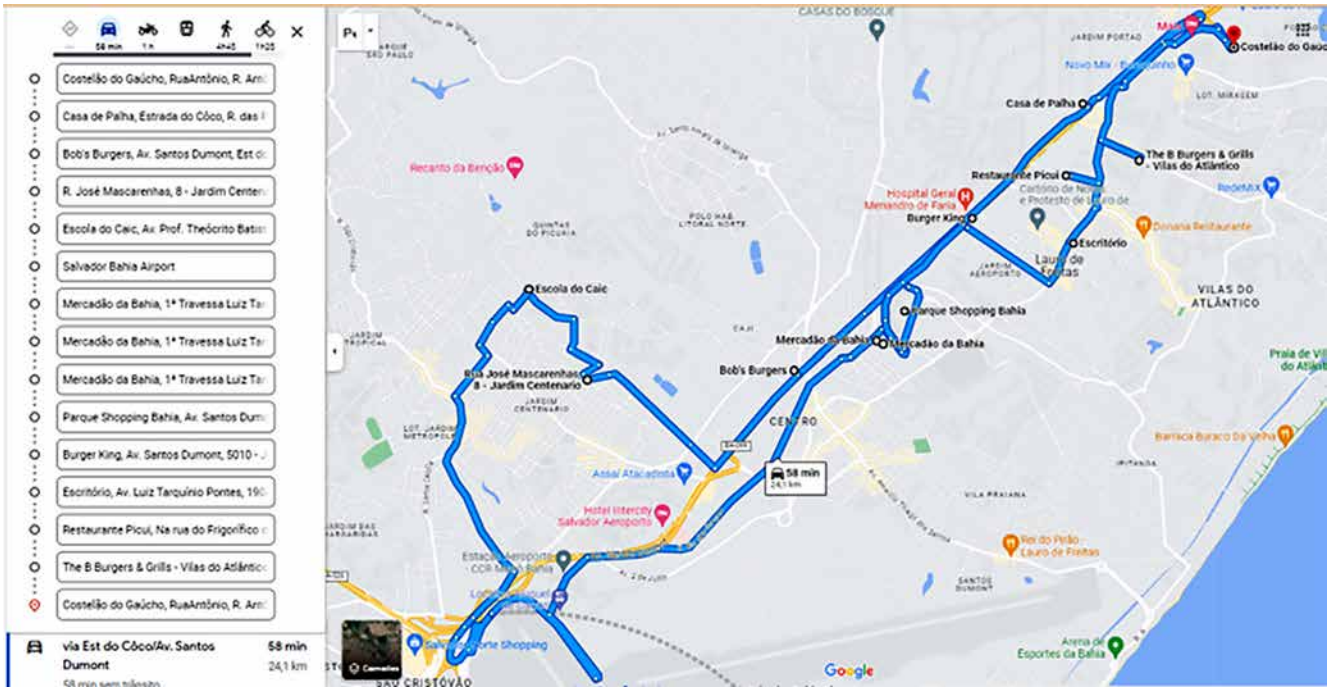


Figure 6. Optimized route designed.



Such initiatives facilitate efficiency gains without incurring substantial costs, such as fuel consumption, vehicle depreciation, labor, and potential toll charges.

Conclusion

Through the conducted research, it became evident that implementing the traveling salesman algorithm and optimizing the vegetable oil collection route represent significant strides toward efficiency and cost savings in logistics processes. A comparison between the current and optimized routes revealed a substantial saving of 23.4 km, underscoring the positive impact of technology utilization in reducing costs and enhancing logistics operations. The case demonstrates how innovation, coupled with free software adoption, can give companies competitive advantages. A notable illustration is the optimization of collection and delivery routes, resulting in decreased travel expenses and concurrently providing customers with a more satisfactory experience characterized by swift service. Furthermore, it underscores the importance of comprehending and addressing logistical considerations, such as using algorithms, in pursuing solutions that optimize the flow of materials and information, ensuring more precise management.

The quest for intelligent and efficient solutions becomes imperative in a landscape where logistics encounters persistent challenges. Route optimization is just one instance among the myriad possibilities technology presents for enhancing logistics management. Thus, the adoption of innovative approaches and the relentless pursuit of improvements are indispensable for the success of companies in today's dynamic market environment.

References

1. Ballou RH. Gerenciamento da cadeia de suprimentos/ logística empresarial (5th ed.). Foz do Iguaçu: Bookman, 2006.
2. Barreto RCP, Ribeiro AJM. Logística no Brasil: uma análise do panorama dos modais rodoviários e ferroviários no cenário nacional demonstrando as vantagens e desvantagens das referidas modalidades. Revista Livre de Sustentabilidade e Empreendedorismo 2020;5(3):145–176. Retrieved from http://www.relise.eco.br/index.php/relise/article/view/355 51
3. Borcherds PH. Python: a language for computational physics. Computer Physics Communications 2007;177(1-2):199–201. doi: 10.1016/j.cpc.2007.02.019.
4. Dantzig G, Fulkerson R, Johnson S. Solution of a large-scale traveling-salesman problem. Journal of the Operations Research Society of America 1954;2(4):393–410. Retrieved from http://www.jstor.org/stable/16669558.
5. Deggeroni R. Uma introdução à teoria dos grafos no ensino médio. Licenciatura em Matemática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2010. Retrieved from https://lume.ufrgs.br/handle/10183/29152.
6. de Melo GS. Introdução à teoria dos grafos. Mestrado profissional em Matemática, Universidade Federal da Paraíba, João Pessoa, 2014. Retrieved from https://repositorio.ufpb.br/jspui/bitstream/tede/7549/5/arquivototal.pdf.
7. Dias MA. Introdução à logística: fundamentos, práticas e integração. São Paulo: Atlas, 2017.
8. Lopes GR, Almeida AWS, Delbem A, Toledo CFM. Introdução à análise exploratória de dados com python. Minicursos ERCAS ENUCMPI 2019:160–176.
9. Santos HG, Toffolo TAM. Tutorial de desenvolvimento de métodos de programação linear inteira mista em python usando o pacote python-mip. Pesquisa Operacional para o Desenvolvimento 2019;11(3):127–138.
10. Pereira D, Silva MA. Introdução a logística. Revista Gestão em Foco 2017;9:291–304.