[](https://github.com/MafaldaPaco/cifo/tree/main)Optimization of Dynamic Bus Routes to Enhance Urban Mobility and Sustainability for EcoPioneer

[Repository](https://github.com/MafaldaPaco/cifo)

Flavia D Motta, 20230574; Flavio Ivo Riedlinger, 20230571and Mafalda Paço, 20220619.

Keywords: Vehicle Routing Problem, Green Vehicle Routing Problem, Genetic Algorithm, Game Strategies, EcoPioneer.

Statement of Contribution: All authors contributed equally to the study. They were responsible for data collection, writing the report, designing the study, conducting data cleaning, data analysis, and contributing to the discussion of the results.

# Introduction

Urban mobility presents significant challenges in modern cities, with the excessive use of personal vehicles leading to severe traffic congestion and increased carbon emissions, which negatively impact the environment. To address these issues, a novel approach involves implementing dynamic and customized bus routes optimized based on real-time passenger demand. This strategy promotes sustainability and enhances community engagement by encouraging shared transportation solutions [1]. Problem Statement- The Vehicle Routing Problem (VRP) involves scheduling vehicles to transport goods between a depot and customers, with the goal of identifying optimal routes that minimize travel costs while meeting customer demands and adhering to vehicle capacities. Due to computational constraints, metaheuristic approaches such as genetic algorithms, adapted from the Traveling Salesman Problem (TSP), are often employed. Traditionally, VRP solutions focus on minimizing economic costs for logistics service providers by reducing the number of vehicles and drivers and the financial cost associated with distance traveled. However, recent developments indicate that environmental considerations are becoming integral to corporate social responsibility policies. This shift is driven by specific targets set for the transport sector and the competitive advantage for companies that can demonstrate environmentally friendly practices. Consequently, incorporating green logistics into VRP solutions is becoming increasingly important, aiming not only for cost efficiency but also for reduced environmental impact [2].

# Project Objective

This research project aims to solve a VRP using four vehicles and a single depot for *EcoPioneer*, focusing on visiting each location precisely once. Leveraging data from Google developers [[3](https://developers.google.com/optimization/routing/vrp?hl=pt-br)], the goal is to determine the most efficient routes for the fleet of vehicles to minimize the overall distance traveled. Additionally, this project seeks to dynamically and customizable optimize bus routes. Instead of following predetermined paths, buses will generate routes based on passenger reservations made the previous day. Passengers will specify their departure and arrival times, along with their starting and destination points. The primary objective is to plan bus routes that minimize reliance on personal cars, reduce carbon emissions, and enhance the overall efficiency of public transport systems. You can find the project repository [here](https://github.com/MafaldaPaco/cifo/tree/main).

# Implementation

To address the complexity of the VRP, this research will employ genetic algorithms, leveraging their proven efficacy in similar optimization problems such as the TSP. The study will begin with a comprehensive literature review to identify the most effective genetic algorithm components and strategies. These components will then be adapted and integrated into a tailored algorithm for dynamic bus routing.

Key building blocks and design options for the genetic algorithm include:

1. **Representation** - Data Structure for Chromosomes: This involves selecting an appropriate data structure to represent each chromosome in the population. The individual is represented by a list of lists – each inner list represents the route of a vehicle. For a 4-vehicle problem our individual will be a list of up to 4 routes. The number of vehicles allocated for the day is chosen randomly between 1 and however many vehicles the fleet has – in our case 4. A route is a list of locations, in order of visit.

[ [16, 1, 3, 14], [8, 5, 11, 7], [12, 9, 15, 6], [4, 2, 10, 13] ]

1. **Initialization Procedure:** This step involves creating the initial population of chromosomes, ensuring a diverse set of potential solutions. **Fitness Function:** There is a penalty added for each vehicle after the first, to incentivize using less vehicles to create a greener fleet.
2. **Evolution:**  it was adapted the evolve function to better suit our needs. We altered the way we applied **elitism** - elitism tends to improve the fitness of the generations, by saving the best value and, therefore, guaranteeing that the fitness will never decrease through the generations. With the goal of avoiding local optimums, we altered our elitism implementation to save not only the best individual, but an *x* number of elites. The intuition behind this is that while the first elite might be the best at a certain stage of the evolution, it might lock the development to a local optimum. With more options, we hope to get a higher chance at the best fitness available. With this goal in mind, it also was implemented **a plateau tolerance**, that will change our parameters after n generations without improvement. This technique was brought to our attention by a student who farms and used this grafting technique on his plants. Grafting combines two plants to get the characteristics of both. When this logic is applied to our problem what we get is once the plateau threshold is reached, the values on our crossover and mutation rate are altered. Explicar se aumentamos ou diminuimos os valores e porquê (testar).
3. **Evaluation Procedure:** Each chromosome is evaluated to determine its fitness, which reflects how well it meets the objectives of the VRP, such as minimizing travel. For a given chromosome, representing a potential route configuration, the fitness function calculates the total distance traveled by the vehicles. The coordinates of the passengers (xi, yi) are used to determine the distance between two passengers, which is computed using the Euclidean distance formula. The fitness function f(x) for a chromosome is thus defined as the sum of the Euclidean distances between consecutive points in the route.
4. **Evolutionary Operators:** These operators include mutations, which introduce small random changes to individual chromosomes, and crossovers, which combine parts of multiple chromosomes to create new offspring. Mutations help maintain genetic diversity, while crossovers enable the combination of successful traits from different parents.
5. **Selection Procedure:** This process identifies the chromosomes that will be used for crossover based on their fitness, as well as those that will be discarded from the population to ensure continuous improvement. Selection is the choice of individuals to which apply the genetic operators, i.e. selecting the individuals that will generate offspring for the next generation. It needs to strike a balance – too-strong and the highly fit individuals will take over, reducing diversity; too-weak and the evolution will be too slow. In class we implemented the Fitness-Proportionate Selection and the Tournament Selection, so for this project we implemented the Sigma Scaling and the Rank Selection – inspired by Mitchell (1996) [4]. Sigma Scaling keeps the selection pressure somewhat constant throughout the evolution process. An individual’s expected value is a function of its fitness, and the population mean and standard deviation. It gives individuals whose fitness is significantly different higher weights - it’ll give an individual with fitness one standard deviation above the average 1.5 expected offspring. In order to maintain diversity, it doesn’t eliminate individuals, by avoiding attributing expected values of zero. We selected the individual based on cumulative probability, meaning that the individuals are chosen with a probability that is proportional to their scaled fitness.

Rank Selection’s purpose is to prevent a quick convergence. The individuals are ranked according to their fitness, and their expected value depends on their rank rather than on absolute fitness. This reduces the selection pressure when variance is high, as this method is insensitive to differences in fitness – only the position in the rank matters. For the rank we calculated it as an arithmetic series, and then selected the individual based on cumulative probability.

1. **Tuning**: Genetic algorithms perform well in scenarios where finding a good solution is more important than finding the best solution. The population size affects the amount of genetic material available during the search. If it is too small, the search cannot adequately cover the space; if it is too large, the GA wastes time evaluating chromosomes. The crossover probability (pc) is set to xx or yy, and the mutation probability (pm) to xx or yy, after continuous fine-tuning. Inversion Mutation was chosen after comparing it with reciprocal exchange mutation. The initial population is generated using a random function from the C library, and pseudo-random numbers are used in the selection method. Pure genetic algorithms are employed, with selection using a roulette wheel or SUS with elitism.The maximum number of generations for all problems was ???.
2. **Termination Condition:** The evolutionary process continues until a predetermined condition is met, such as reaching a maximum number of generations or achieving a satisfactory level of fitness.

The proposed algorithm will be tested through a series of simulations using real-world data to ensure practical applicability. Key performance indicators (KPIs) will include total distance traveled, fuel consumption, carbon emissions, and passenger satisfaction. The results will be analyzed to iteratively refine the algorithm, ensuring optimal performance across various urban environments [5].

# Results

The stopping criterion is the number of generations evolved. The GA stops after completing a specified number of iterations. The solution is the best chromosome in the final generation, which may be a local or global optimum. It is important to note that GA often provides approximate solutions rather than optimal ones. The reported optimal value measures solution quality.

# Open Issues and Challenges

xxxxxx

# Conclusions

The study focused on various crossover and mutation operators and their effects on optimizing vehicle routes. Experimental results indicated that different operators have varying impacts on VRP, and the choice of operators significantly influences the quality of solutions. Future work may explore hybrid approaches and other evolutionary strategies to improve genetic algorithms’ performance in solving complex routing problems.

# References

[1] Paulo Oliveira de Oliveira da Costa et al. A recent review of solution approaches for green vehicle routing problem and its variants. Electronic Notes in Discrete Mathematics 64 (2018) 65–74.

[2] Paolo Toth and Daniele Vigo. The vehicle routing problem. SIAM, 2002.

[3] https://developers.google.com/optimization/routing/vrp?hl=pt-br

[4] Mitchell, M. (1996). An introduction to genetic algorithms. https://doi.org/10.7551/mitpress/3927.001.0001

[5] Vanneschi, L., Silva, S. (2023). Particle Swarm Optimization. In: Lectures on Intelligent Systems. Natural Computing Series. Springer, Cham. https://doi.org/10.1007/978-3-031-17922-8\_4.