



**UPM**  
UNIVERSITI PUTRA MALAYSIA  
BERILMU BERBAKTI



**FACULTY OF COMPUTER SCIENCE AND  
INFORMATION TECHNOLOGY**

# **Intelligent Bus System (IBS UPM)**

**TITLE : FINAL PROJECT (INTELLIGENT BUS SYSTEM)**

**COURSE : ARTIFICIAL INTELLIGENCE**

**COURSE CODE : CSC3600**

**LECTURER'S NAME : DR. ERZAM BIN MARLISAH**



# Meet Our Best Team



**Ramesh Arvin**  
213111



**Nor Syakila**  
212927



**Ponmugillan**  
210301

A photograph showing a row of modern university buses parked at a bus stop. There are three visible buses: one green, one white, and one blue. The blue bus is in the foreground, showing its front and side. The white bus is behind it, and the green bus is further back. They are all parked on a paved area with a zebra crossing in the foreground.

# Introduction

The implementation of a Genetic Algorithm (GA) is used to optimise bus routes in our university's bus system, Universiti Putra Malaysia (UPM). The current UPM bus system has various faults, including uneven route frequencies and the lack of a trustworthy bus schedule for students to refer to.

This implementation attempts to increase the consistency of bus frequencies and offer students with an effective transportation system within the university campus by applying the genetic algorithm to the bus route optimisation problem at UPM.

# Problem Statement



**Inconsistent bus frequencies**

**Absence of a reliable bus schedule**

**Minimization of waiting time**

**Efficient transportation between bus stops**

The successful implementation of the GA for bus route optimization at UPM will lead to a more consistent bus service, improved transportation efficiency, and reduced waiting times for students.

# Objectives

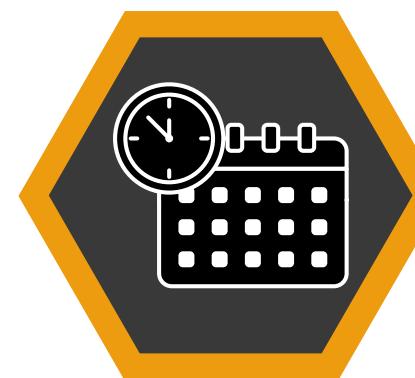
**Optimize bus routes**



**Minimize waiting time**



**Improve consistency of bus frequencies**



**Enhance transportation efficiency**



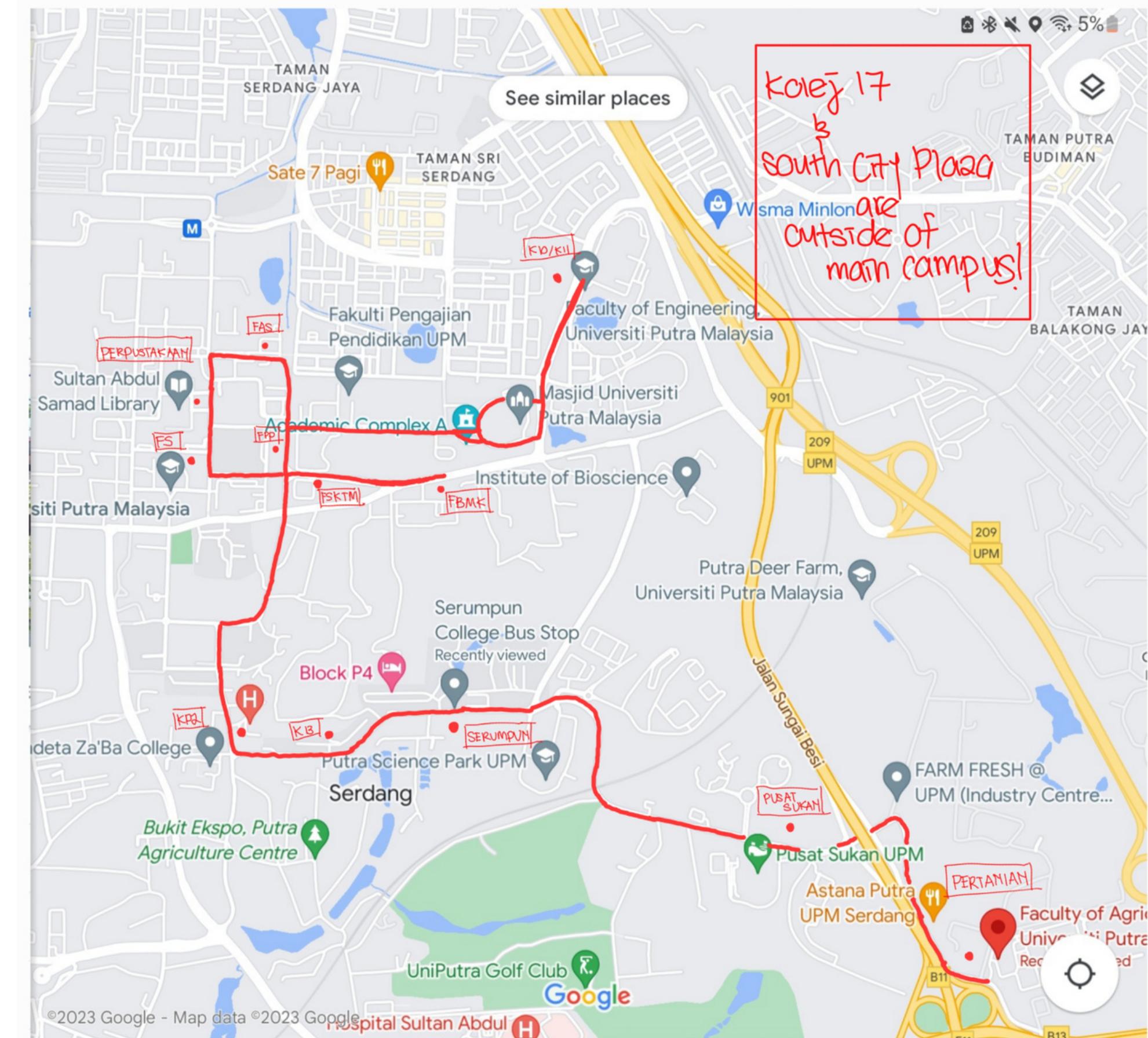
**Generate a reliable bus schedule**



**Track and analyze optimization results**



# Locations Covered



● **Bus Stops in UPM**



# How does Genetic Algorithm Works in Bus Routes Optimization?

## Generation

Think of generations as different iterations of bus routes.



## Population Size

The population size refers to the number of different bus routes that exist within each generation.



## Max Route Size

The maximum route size is the limit on the number of bus stops a single route can have.



## Mutation Probability

Mutation probability represents the likelihood of introducing random changes to the bus routes during each generation.



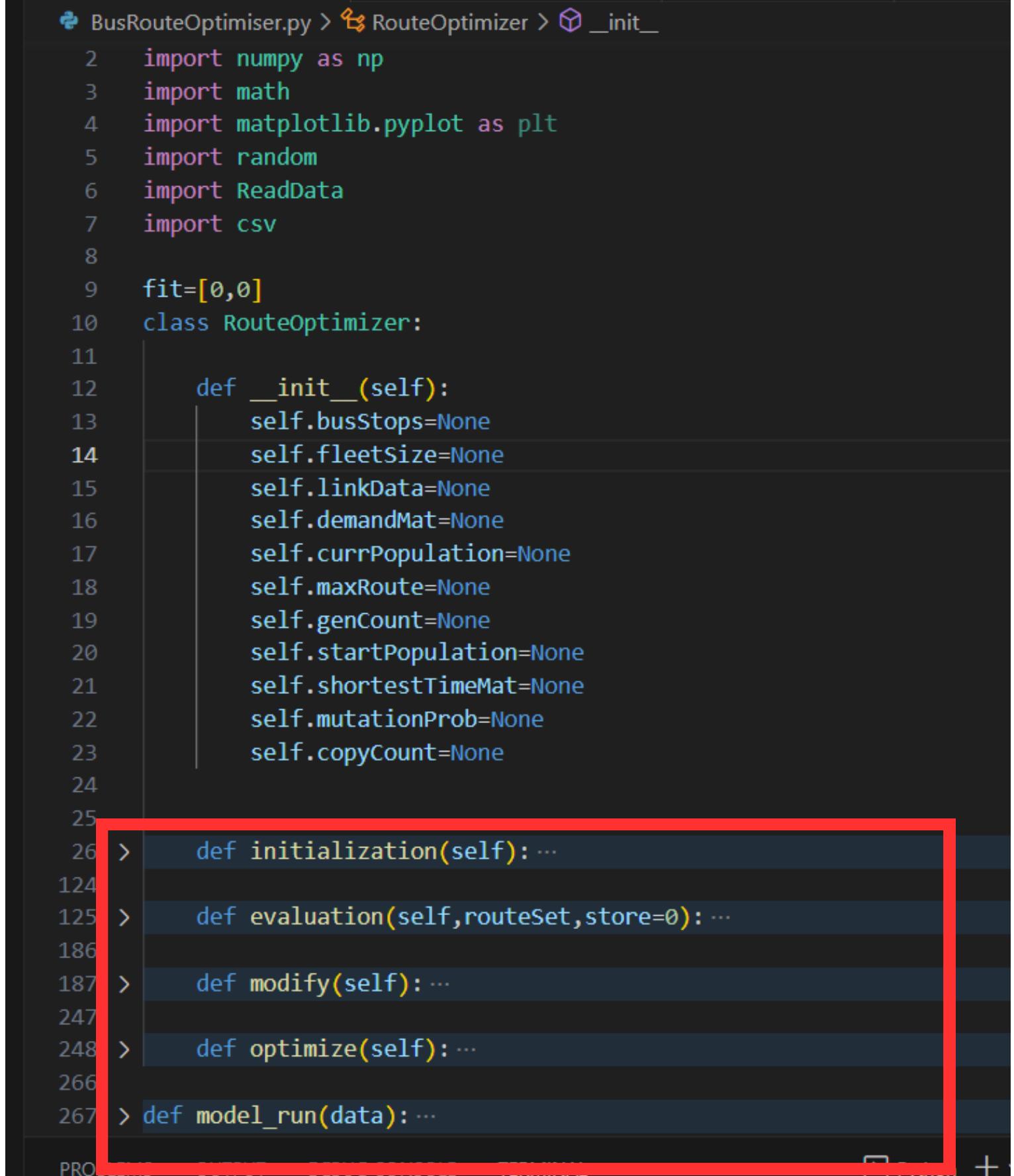
## Copy Count

The copy count determines how many copies of the best-performing routes are carried forward to the next generation.



# Pseudocode of GA

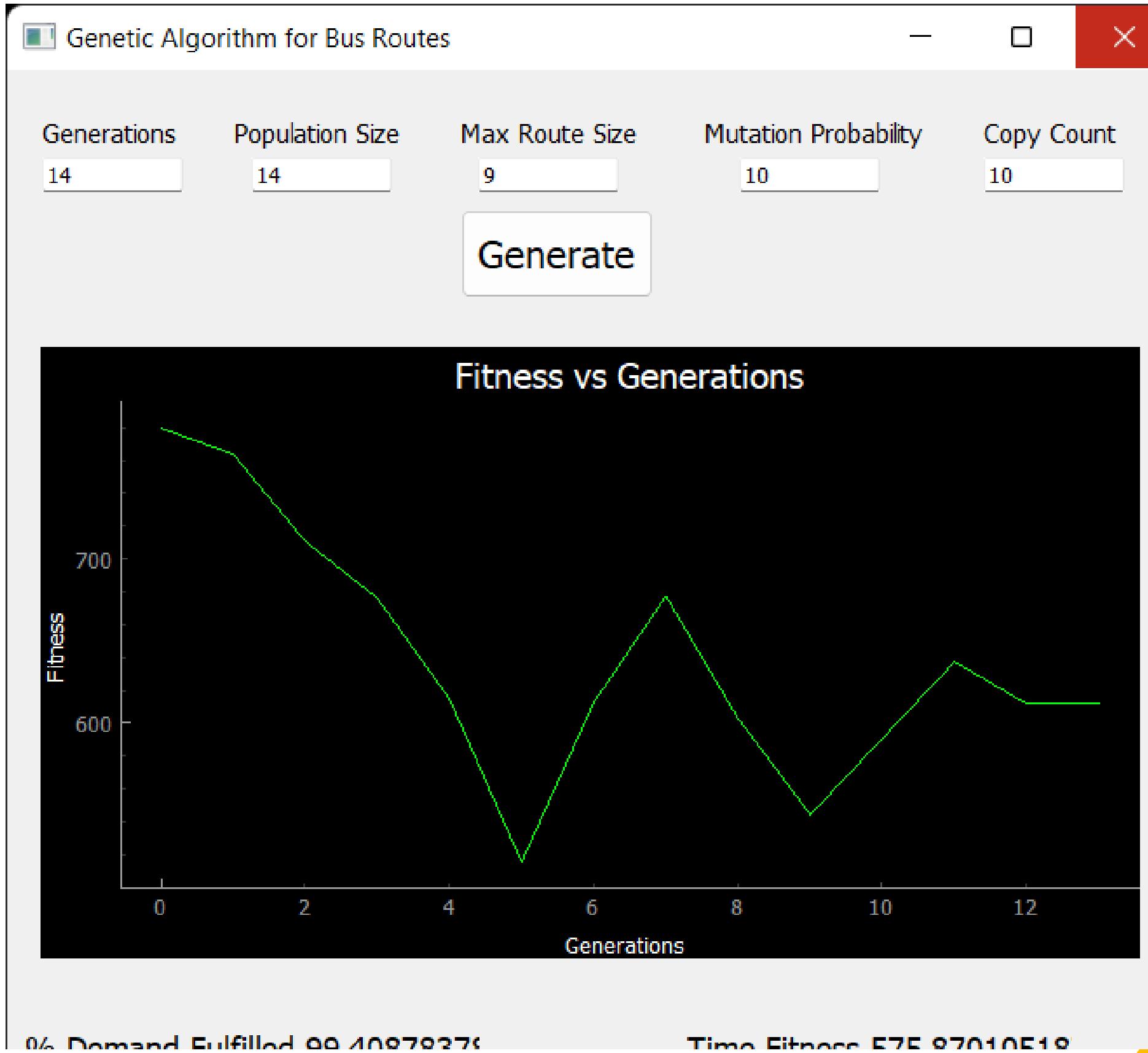
1. Choose initial population
2. Evaluate the fitness of each individual in the population  
Repeat
3. Select best-ranking individuals to reproduce
4. Breed new generation through crossover and mutation (genetic operations) and give birth to offspring
5. Evaluate the individual fitnesses of the offspring
6. Replace worst ranked part of population with offspring
7. Until <terminating condition>



```
# BusRouteOptimiser.py > RouteOptimizer > __init__
2 import numpy as np
3 import math
4 import matplotlib.pyplot as plt
5 import random
6 import ReadData
7 import csv
8
9 fit=[0,0]
10 class RouteOptimizer:
11
12     def __init__(self):
13         self.busStops=None
14         self.fleetSize=None
15         self.linkData=None
16         self.demandMat=None
17         self.currPopulation=None
18         self.maxRoute=None
19         self.genCount=None
20         self.startPopulation=None
21         self.shortestTimeMat=None
22         self.mutationProb=None
23         self.copyCount=None
24
25
26     > def initialization(self): ...
27
28     > def evaluation(self,routeSet,store=0): ...
29
30     > def modify(self): ...
31
32     > def optimize(self): ...
33
34     > def model_run(data): ...
35
36
37 PRO
```

## Implementation in Our Codes

# Code Demonstration



```
path('..')/.config/runner
use truncation of the maximum
  environment its running environment
spec_helper'
'e 'rspec/rails'

require 'capybara/rspec'
require 'capybara/rails'

Copybara.javascript_driver = :webkit
Category.delete_all; Category.create!
Shoulda::Matchers.configure do |config|
  config.integrate do |with|
    with.test_framework :rspec
    with.library :rails
  end
end

# Add additional requires below this line if needed
# Requires supporting ruby files with custom matchers
# spec/support/ and its subdirectories
# run as spec files by default. This means you can
# in _spec.rb will both be required and run
# run twice. It is recommended that you name
# convention on the command line to
# and for 'mongoid'

# buffer
```

# Code Demonstration

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** MainGui.py - BusRouteGenerator-master - Visual Studio Code.
- Explorer Sidebar:** Shows the project structure under "BUSROUTEGENERATOR-M...".
- Code Editor:** Displays MainGui.py with code related to a Genetic Algorithm for Bus Routes. It includes functions like `retranslateUi`, `gen_equ`, `pop_equ`, `mut_equ`, `route_equ`, and `route_gen`.
- Terminal:** Shows the command line output: PS C:\AI\BusRouteGenerator-master> & C:/Users/user/AppData/Local/Programs/Python/Python311/python.exe c:/AI/BusRouteGenerator-master/MainGui.py
- Floating Window:** A modal window titled "Genetic Algorithm for Bus Routes" displays a line graph titled "Fitness vs Generations". The x-axis is "Generations" (0-9) and the y-axis is "Fitness" (680-760). The graph shows a jagged line with a red play button icon overlaid. Below the graph, status text reads "% Demand Fulfilled 74.57770270" and "Time Fitness 707.01454501".
- Status Bar:** Includes file navigation icons, a search bar, and system status indicators like battery level, signal strength, and date/time (9:13 PM, 25/6/2023).

# ADVANTAGES & DISADVANTAGES

## Optimization Capability

- Genetic algorithms excel at solving optimisation problems
- Can search for better solutions and gradually converge towards an optimised route



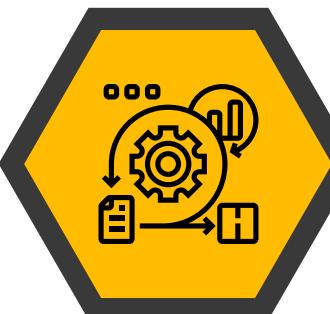
## Flexibility

- Genetic algorithms offer great flexibility when it comes to handling complex problems with multiple variables and constraints



## Adaptability

- Generate fresh optimized routes using the updated information
- Generate efficient and effective bus routes that accommodate the latest changes and requirements.



## Algorithm Complexity

- Make them resource-intensive and time-consuming
- Genetic algorithms may demand considerable computational resources



## Dependence on Parameters

- Reliant on various parameters
- Selecting the right parameter values can be a difficult task



## Limited Accuracy

- Provide approximate solutions rather than exact solutions
- Algorithm's effectiveness depends on the fitness evaluation function and the representation of the problem domain





## Limited Information

- Scarcity of information for optimization was a significant drawback.
- Limited data included bus stop details, stop frequency, travel times, bus capacity, and route limits per bus.

# Improvements



## Implementation Approach

- Recognizing the availability of other tools, algorithms, and AI techniques is crucial for further improving the optimization process.
- Alternative optimization algorithms like Ant Colony Optimization, Simulated Annealing, or Particle Swarm Optimization can be explored.



## Accuracy of Results

- The project's results relied on personal experience and feedback from a small group, along with travel time data from Google Maps.
- Conducting a comprehensive survey or gathering data from a wider range of UPM students is recommended for achieving more accurate and reliable results.



## Basic Implementation

- The project should be considered as a starting point for enhancing the bus system at UPM, rather than a complete implementation.
- Further research, development, and collaboration with stakeholders are essential to address the limitations of the existing system.

# Business Models



## Software Licensing Model

A potential business model for the project is to license the bus route optimization software, allowing users or organizations to pay a fee for using the software and improving their transportation efficiency.



## Consultancy and Implementation

The project can generate revenue by offering consultancy and implementation services to bus companies, transportation authorities, or educational institutions seeking to optimize their bus routes using genetic algorithms.



## Partnerships and Collaborations

The project can form partnerships with bus service providers, transportation companies, or smart city initiatives to integrate the genetic algorithm optimization solution.



# Conclusion

This project utilized a Genetic Algorithm (GA) to optimize bus routes at Universiti Putra Malaysia (UPM), aiming to improve transportation efficiency for students. However, limitations arose due to limited information on bus stop data, frequency, travel time, and capacity. Further improvements could be done to enhance the accuracy of optimization. To summarise, this project demonstrates the potential of GA for bus route optimization.



Syakila Mugi Ramesh  
Company

# Thank You

For Your Attention



Visit Our Website

[kilamugirameshcompany.com](http://kilamugirameshcompany.com)

