**A logo with blue lines

Description automatically generated**

**University Of Niagara Falls Canada**

Master of Data Analytics

Group Project

**ROUTE OPTIMIZATION OF URBAN PUBLIC TRANSPORTATION**

Course: Operations Analytics

(DAMO610-1) Winter 2025

Group Member:

Setu Chaudhari (NF1009018)

Heta Chavda (NF1014555)

Enejo, David Colonel

Professor: Cosimo Girolamo

Table of Contents

[Chapter 1: Understanding the Challenges in Public Transport 3](#_Toc193688396)

[1.1 Introduction 3](#_Toc193688397)

[1.2 Defining the Problem 3](#_Toc193688399)

[1.2.1 Demand Variations at Different Times of the Day 4](#_Toc193688400)

[1.2.2 Reducing Passenger Waiting Time 5](#_Toc193688401)

[1.2.3 Optimizing the Use of Available Buses 5](#_Toc193688407)

[1.2.4 Real-Time Road Conditions and Congestion 6](#_Toc193688413)

[1.2.5 Operational Costs and Environmental Impact 6](#_Toc193688420)

[1.3 Why This Matters 7](#_Toc193688421)

[1.3.1 Service Reliability and Passenger Satisfaction 7](#_Toc193688422)

[1.3.2 Increased Ridership 7](#_Toc193688424)

[1.3.3 Operational Cost Reduction 7](#_Toc193688426)

[1.3.4 Sustainability of the Environment 8](#_Toc193688427)

[1.3.5 Social Equity 8](#_Toc193688428)

[Chapter 2: Review of Bus Scheduling Problem and Optimization Model 9](#_Toc193688429)

[2.1 Problem Review: The Importance of Efficient Bus Scheduling 9](#_Toc193688430)

[2.2 Mathematical Model for Bus Scheduling Optimization 9](#_Toc193688433)

[2.2.1 Decision Variables 9](#_Toc193688434)

[2.2.2 Objective Function 10](#_Toc193688435)

[2.2.3 Constraints 10](#_Toc193688436)

[2.3 Coding the Model Using Google OR-Tools 11](#_Toc193688437)

[2.4 Solving the Model Using Synthetic Data 13](#_Toc193688438)

[Chapter 3: Extending the Mathematical Model 14](#_Toc193688439)

[3.1 Enhancing the Mathematical Model 14](#_Toc193688440)

[3.2 New Constraints and Variables 14](#_Toc193688441)

[3.2.1 Additional Decision Variables 14](#_Toc193688442)

[3.2.2 Revised Objective Function 15](#_Toc193688443)

[3.2.3 New Constraints 15](#_Toc193688444)

[3.3 Extended Python Implementation 16](#_Toc193688445)

[3.4 Results and Observations 19](#_Toc193688446)

[VISUALIZATION 21](#_Toc193688447)

[1. Route Visualization Using NetworkX and Matplotlib 21](#_Toc193688448)

[2. Heatmap of Travel Costs Using Seaborn 22](#_Toc193688449)

[3. Bar Chart of Passenger Demand Over Time 23](#_Toc193688450)

[Appendix A: Python Script 24](#_Toc193688451)

[Conclusion 28](#_Toc193688452)

# Chapter 1: Understanding the Challenges in Public Transport

## 1.1 Introduction

Public transport systems form a core aspect of city life, providing millions of commuters with vital mobility service daily. Buses are among the prevailing means of transport within such systems, especially in highly populated urban cities. On-time, efficient, and convenient bus service is provided through adequate bus scheduling. Public bus systems, however, have several challenges that influence both the quality of service and operational cost efficiency.

Having an efficient bus schedule is a delicate balancing act. It's a question of having buses arrive on time but also operate frequently enough to meet demand. An efficient schedule must take a variety of considerations into account—how many patrons need the service, traffic patterns, operating costs, and even environmental factors. If schedules are ill-conceived, it translates into delays, overloading, wasted resources, and a whole unpleasant experience for passengers.

These are compounded by a series of external and internal variables. Urbanization is inducing the demand for public transport to increase, while urban traffic is worsening road conditions. Further, weather conditions, accidents, and maintenance on roads affect the punctuality of buses. These all contribute to making it increasingly difficult to institute and maintain an efficient bus scheduling system.

## This chapter will cover the inherent issues that public transportation agencies face in attempting to optimize bus schedules, such as understanding passenger demand patterns, operational constraints, and external factors such as traffic.

## 1.2 Defining the Problem

The bus scheduling problem is intricate with multiple interrelated challenges. A holistic solution is necessary to resolve these challenges so that buses can be utilized effectively, have adequate passenger coverage, and ensure reliability. The primary challenges include:

The **bus network scheduling problem** involves determining:

* **Which routes should be serviced?** (Route selection)
* **How frequently should buses operate on each route?** (Frequency optimization)
* **How should buses be assigned to routes to balance demand and operational efficiency?** (Fleet allocation)

### 1.2.1 Demand Variations at Different Times of the Day

Passenger demand for bus transportation is different during the day, with peaks of high demand at peak hours, typically morning and evening. Demand at off-peak hours is much lower, and buses in some cases become underutilized. Such variations require dynamic scheduling methods to best utilize buses.

**Major Factors Determining Demand:**

•Work Hours: Peak demand in morning and evening peak hours

• Weather Conditions: Inclement weather (snow, rain) can increase demand as passengers avoid walking.

• Special Events: Public events, festivals, and holidays tend to create short-run demand increases.

• Seasonality: Traveler season, school breaks, or festivals during holidays are among those occurrences which impact passengers.

Transit authorities are responsible for such fluctuations by varying the frequency of buses, dispatching more buses during peak times, and route-based demand.

### 1.2.2 Reducing Passenger Waiting Time

### Long bus waits are a significant source of complaints among passengers and may lead to dissatisfaction, lower ridership, and crowding on other transportation systems. Bus timetables must minimize waiting while balancing frequency and resources utilization.

### Strategies to Reduce Waiting Times:

### •Optimize Bus Intervals: Bus timings should be arranged such that buses arrive often enough in peak hours to fulfill demand without becoming congested.

### • Real-time Tracking: Digital signs or cell phone applications may give riders real-time bus arrival times, reducing perceived waiting time.

### •On-time Departures: Efficient planning to allow buses to depart on schedule, reducing delays.

### 1.2.3 Optimizing the Use of Available Buses

### In every public transport system, there are limited bus fleets. Bus deployment thus must be maximized to deal with fluctuating passenger demand without over-allocating assets. This involves discovering the optimum number of buses on each route, maximizing bus scheduling to reduce idle time, and maximizing bus utilization.

### Key Considerations in Optimization:

### • Fleet Size: Having sufficient buses during peak demand periods and avoiding overuse during off-peak demand periods.

### • Bus Allocation: Correct allocation of buses to different routes and having the right number of buses at the right time.

### • Efficiency of Usage: Minimizing idle times between routes and maximizing bus availability.

### 1.2.4 Real-Time Road Conditions and Congestion

### Traffic congestion is perhaps the most dreaded nightmare of urban bus operations. Buses are greatly delayed by traffic conditions, accidents, road work, and overall traffic congestion, especially during peak hours. Delays translate into late arrival times, increased operating costs, and low customer satisfaction.

### To overcome this, bus routes must be planned with real-time traffic information. With GPS and traffic management combined systems, buses can be routed through the bottlenecks, and probable delays can be predicted and communicated to travelers.

### Major Strategies in Road Condition Management:

### • Dynamic Routing: Route reconfiguring in real time based on current traffic and road blockages.

### •Delay Forecasting: Utilizing real-time and historical traffic data to predict delays and schedule bus timetables.

### • Right Passenger Information: Advising passengers on punctual arrivals or diversions gives them sufficient information to organize their trips.

### 1.2.5 Operational Costs and Environmental Impact

Ineffective bus scheduling can lead to excessive operating costs. This includes increased fuel cost due to idling time, inefficient routes, and unnecessary maintenance cost. Furthermore, ineffective timetables translate to increased carbon footprints, and these are some of the reasons for environmental pollution.

Through the reduction of bus schedules, agencies can reduce fuel consumption, lower wasteful routes, and increase the overall efficiency of bus fleets.

**Cost and Environmental Efficiency Strategies**

* Fuel Economy: Schedule routes and timetables to reduce wasteful fuel utilization
* Minimize Empty Trips: Do not run half-full or empty buses to save on fuel costs
* Environment Sustainability: Include green buses to reduce emissions and increase sustainability.

## 1.3 Why This Matters

Optimization of public transportation systems is not only required to improve service quality but also to make it sustainable in the long term, reducing operating costs, and improving social equity. An optimally designed bus scheduling system is advantageous to transit agencies and riders in numerous ways.

### 1.3.1 Service Reliability and Passenger Satisfaction

### There is space for higher ridership and satisfaction due to the timely arrival and reduced waiting time of the bus service. As the passengers have less waiting time and less delay, their confidence in the system is gained, which provides space for higher ridership and satisfaction in total.

### 1.3.2 Increased Ridership

### Well-organized bus timetables promote more people to use public transport. Efficient timetables can appeal to passengers by offering timely, frequent, and comfortable transport. This reduces the number of vehicles on the road and eventually lessens urban congestion and pollution.

### 1.3.3 Operational Cost Reduction

Transit organizations can reduce labor, maintenance, and fuel expenses by streamlining bus timetables. For example, agencies can save idle time, fuel consumption, and off-peak staff by making the best use of buses.

### 1.3.4 Sustainability of the Environment

Public transportations aid in reducing city carbon footprints. Efficient planning of bus schedules for minimal fuel waste and promoting the use of green-friendly buses are measures transit businesses can undertake to contribute to the environment-friendly city transportation systems..

### 1.3.5 Social Equity

Optimizing bus timings provides public transport to all people in society. Proper bus services ensure equal access to jobs, education, healthcare, and other essential facilities for car non-owners. It is crucial in suppressing social inequality and promoting inclusive urban growth.

# Chapter 2: Review of Bus Scheduling Problem and Optimization Model

## 2.1 Problem Review: The Importance of Efficient Bus Scheduling

Efficient bus scheduling aim and main role is in the operation of public transportation systems. A weak designed bus schedule can lead to increased costs, delays, passenger dissatisfaction, and environmental degradation. The need for optimized bus scheduling arises from the desire to balance several factors, including passenger demand, operational constraints, and external influences such as road conditions and traffic congestion.

## To solve these issues, several methods of optimization are employed. These aims are to design bus timetables that will minimize cost of operation, enhance service quality, and meet environmental and social goals. The model applied in solving bus scheduling issues typically involves analysis of the demand variability, minimizing waiting times, optimizing fleet usage, and accounting for real-time traffic data.

## The importance of optimized bus scheduling is evident in the numerous benefits that it confers, for example, passenger waiting time is reduced to its minimum, improved utilization of the buses, less congestion, and reduced operational cost.

## 2.2 Mathematical Model for Bus Scheduling Optimization

In the context of the bus scheduling problem, the mathematical model involves the integration of constraints related to demand, fleet capacity, operational costs, and external factors like road conditions.

A typical mathematical model used for bus scheduling involves decision variables, objective functions, and constraints. Below is an outline of the model proposed in the article "Modeling and Analysis of Bus Scheduling Systems of Urban Public Bus Transport" by Berhan E. et al. (2014).

### 2.2.1 Decision Variables

The decision variables represent the various elements that can be adjusted to achieve optimal bus schedules. In the context of bus scheduling, the decision variables can include:

* **X\_ij:** The number of buses assigned to route ii during time period jj.
* **Y\_ij:** The arrival time of the bus at stop ii during time period jj.

### 2.2.2 Objective Function

The scheduling model's objective is represented by the objective function. The objective of bus scheduling is usually to balance meeting passenger demand by minimizing overall operating costs.

The objective function can be represented as:

A black and white text

AI-generated content may be incorrect.

Where:

* C1 is the cost coefficient associated with assigning buses to routes.
* C2 is the cost coefficient associated with bus delays.
* Xij and Yij are decision variables related to bus assignment and arrival times, respectively.

### 2.2.3 Constraints

The model is subject to several constraints, which include:

1. **Demand Constraints:** The number of buses assigned to each route at any given time must be sufficient to meet passenger demand.

**Xij≥Dij**

Where Dij represents the demand for buses on route ii at time period j.

1. **Capacity Constraints:** The number of buses assigned to a route should not exceed the available fleet size.

**A black text with a white background

AI-generated content may be incorrect.**

1. **Time Constraints:** The bus arrival times at different stops must meet the necessary travel times and account for external factors such as traffic.

**Yij = Yi−1, j + Travel Time + Delay**

**Non-Negativity Constraints:** The number of buses assigned, and arrival times cannot be negative.

**Xij≥0, Yij≥0**

### 2.3 Coding the Model Using Google OR-Tools

Now, let’s implement the **mathematical model in Python** using **Google OR-Tools**.

**Installation of OR-Tools**

Before running the optimization, install OR-Tools:

pip install ortools

**Python Implementation**

from ortools.linear\_solver import pywraplp

def optimize\_bus\_scheduling():

solver = pywraplp.Solver.CreateSolver('SCIP')

# Parameters

num\_stops = 5 # Example with 5 stops

cost\_matrix = [[0, 10, 15, 30, 25], # Travel costs

[10, 0, 20, 35, 40],

[15, 20, 0, 25, 30],

[30, 35, 25, 0, 15],

[25, 40, 30, 15, 0]]

num\_buses = 3 # Available buses

max\_transfers = 2 # Maximum allowed transfers

# Decision Variables

X = {} # Route assignment

for i in range(num\_stops):

for j in range(num\_stops):

if i != j:

X[i, j] = solver.BoolVar(f'X[{i},{j}]')

T = solver.IntVar(0, max\_transfers, 'T') # Transfers

# Objective Function: Minimize total cost

solver.Minimize(solver.Sum(cost\_matrix[i][j] \* X[i, j] for i in range(num\_stops) for j in range(num\_stops) if i != j) + T)

# Constraints

for i in range(num\_stops):

solver.Add(sum(X[i, j] for j in range(num\_stops) if i != j) == 1)

# Solve the model

status = solver.Solve()

if status == pywraplp.Solver.OPTIMAL:

print("Optimal solution found:")

for i in range(num\_stops):

for j in range(num\_stops):

if i != j and X[i, j].solution\_value() > 0.5:

print(f'Bus route: {i} -> {j}')

print(f'Minimum Transfers: {T.solution\_value()}')

else:

print("No optimal solution found.")

# Run the optimization

optimize\_bus\_scheduling()

### 2.4 Solving the Model Using Synthetic Data

For testing and demonstrating the model, synthetic data can be generated. The synthetic dataset can include values for bus demand at different times of the day, available fleet size, and travel times between routes. These datasets can be created using AI tools manually, depending on the available resources.

In the example above, the demand matrix specifies the number of buses needed for each route and time period, the fleet\_size limits the total number of buses available, and the travel\_times matrix represents the estimated travel times between stops.

# Chapter 3: Extending the Mathematical Model

Now we have implemented a **basic optimization model** for bus scheduling, we will extend the model by adding **real-world complexities**. The goal is to enhance its applicability by integrating additional business objectives, constraints, and dynamic adjustments based on real-time factors.

## 3.1 Enhancing the Mathematical Model

The original model in Chapter 2 focused on **minimizing cost and passenger transfers**. However, real-world scenarios involve additional considerations:

1. **Traffic Congestion** – Bus travel times vary based on real-time traffic conditions.
2. **Dynamic Passenger Demand** – Demand fluctuates based on time of day, season, and special events.
3. **Multi-Objective Optimization** – A trade-off between **minimizing operational costs, improving service reliability, and reducing emissions**.
4. **Fuel Efficiency and Sustainability** – Optimizing bus schedules to reduce fuel consumption and emissions.

## 3.2 New Constraints and Variables

### 3.2.1 Additional Decision Variables

| **Variable** | **Description** |
| --- | --- |
| Tij | Travel time between stops ii and jj (dependent on traffic conditions) |
| Dt | Passenger demand at time tt |
| Er | Emissions per bus route rr |
| Fij | Fuel consumption on route ii to jj |

### 3.2.2 Revised Objective Function

We now define a **multi-objective function** balancing cost, congestion, and environmental impact:

A black and white math equation

AI-generated content may be incorrect.

Where:

* λ1,λ2,λ3 are weights for **transfer minimization, emission reduction, and fuel efficiency**.
* Eij accounts for **carbon emissions per route** (e.g., CO2 per kilometer).
* Fij models **fuel efficiency based on route distance and congestion levels**.

### 3.2.3 New Constraints

**1. Real-Time Traffic Constraints**

Bus travel times depend on dynamic congestion levels:

A black and white math equation

AI-generated content may be incorrect.

Where:

* Dij is **road density** (vehicles per km).
* Lij is **route length**.
* Vij is **bus speed**, which changes based on congestion data.

To ensure buses avoid high-traffic routes, we impose:

A black text on a white background

AI-generated content may be incorrect.

**2. Passenger Demand Variation**

Passenger demand fluctuates with time:

A black text on a white background

AI-generated content may be incorrect.

Where f(t)f(t) is a function modeling peak vs. off-peak variations. This helps allocate buses dynamically.

**3. Sustainable Operations**

To ensure environmentally friendly scheduling:

A black text on a white background

AI-generated content may be incorrect.

This limits total emissions per route, ensuring compliance with **green transit policies**.

## 3.3 Extended Python Implementation

Now, let's enhance our Python code to include **dynamic traffic conditions and demand variations**.

import numpy as np

from ortools.linear\_solver import pywraplp

def optimize\_extended\_bus\_scheduling():

solver = pywraplp.Solver.CreateSolver('SCIP')

# Parameters

num\_stops = 5 # Example with 5 stops

cost\_matrix = np.array([[0, 10, 15, 30, 25],

[10, 0, 20, 35, 40],

[15, 20, 0, 25, 30],

[30, 35, 25, 0, 15],

[25, 40, 30, 15, 0]])

traffic\_matrix = np.array([[0, 1.2, 1.1, 1.5, 1.3],

[1.2, 0, 1.3, 1.6, 1.4],

[1.1, 1.3, 0, 1.2, 1.1],

[1.5, 1.6, 1.2, 0, 1.2],

[1.3, 1.4, 1.1, 1.2, 0]])

fuel\_consumption = np.array([[0, 5, 4, 8, 7],

[5, 0, 6, 9, 8],

[4, 6, 0, 5, 4],

[8, 9, 5, 0, 6],

[7, 8, 4, 6, 0]])

num\_buses = 3

max\_transfers = 2

max\_emissions = 50 # Example emission cap

# Decision Variables

X = {} # Route assignment

for i in range(num\_stops):

for j in range(num\_stops):

if i != j:

X[i, j] = solver.BoolVar(f'X[{i},{j}]')

T = solver.IntVar(0, max\_transfers, 'T') # Transfers

E = solver.IntVar(0, max\_emissions, 'E') # Emissions

# Objective Function: Minimize cost, transfers, and emissions

solver.Minimize(

solver.Sum(cost\_matrix[i][j] \* X[i, j] \* traffic\_matrix[i][j] for i in range(num\_stops) for j in range(num\_stops) if i != j) +

T + E

)

# Constraints

for i in range(num\_stops):

solver.Add(sum(X[i, j] for j in range(num\_stops) if i != j) == 1)

# Emission Constraints

solver.Add(solver.Sum(fuel\_consumption[i][j] \* X[i, j] for i in range(num\_stops) for j in range(num\_stops) if i != j) <= max\_emissions)

# Solve the model

status = solver.Solve()

if status == pywraplp.Solver.OPTIMAL:

print("Optimal solution found:")

for i in range(num\_stops):

for j in range(num\_stops):

if i != j and X[i, j].solution\_value() > 0.5:

print(f'Bus route: {i} -> {j}')

print(f'Minimum Transfers: {T.solution\_value()}')

print(f'Total Emissions: {E.solution\_value()}')

else:

print("No optimal solution found.")

# Run the optimization

optimize\_extended\_bus\_scheduling()

## 3.4 Results and Observations

* **Dynamic Traffic Adjustment**: The model **automatically avoids high-congestion routes**.
* **Emission Control**: Limits total emissions, promoting sustainable transit.
* **Optimized Scheduling**: Balances cost, transfers, and service quality.

# VISUALIZATION

## 1. Route Visualization Using NetworkX and Matplotlib

You can use **NetworkX** (a Python library for graph visualization) to display the optimized bus routes as a network graph.

**CODE:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Output:**

**A diagram of a bus route

AI-generated content may be incorrect.**

## 2. Heatmap of Travel Costs Using Seaborn

To visualize travel costs between stops, use a **heatmap**:

Code:

A screenshot of a computer program

AI-generated content may be incorrect.

Output:

A diagram of a bus stop

AI-generated content may be incorrect.

## 3. Bar Chart of Passenger Demand Over Time

If you modeled **dynamic demand**, a bar chart can show how passenger volume changes over time.

Code:

A computer screen shot of a program

AI-generated content may be incorrect.

Output:

A graph of different time periods

AI-generated content may be incorrect.

# Appendix A: Python Script

This appendix includes the complete **Python code** implementing the **extended optimization model** using **Google OR-Tools**. The code integrates:

1. **Route Assignment and Scheduling Optimization**
2. **Traffic Congestion Adjustment**
3. **Emission and Fuel Efficiency Constraints**

**Code:**

import numpy as np

from ortools.linear\_solver import pywraplp

def optimize\_extended\_bus\_scheduling():

solver = pywraplp.Solver.CreateSolver('SCIP')

# Parameters

num\_stops = 5 # Example with 5 stops

cost\_matrix = np.array([[0, 10, 15, 30, 25],

[10, 0, 20, 35, 40],

[15, 20, 0, 25, 30],

[30, 35, 25, 0, 15],

[25, 40, 30, 15, 0]])

traffic\_matrix = np.array([[0, 1.2, 1.1, 1.5, 1.3],

[1.2, 0, 1.3, 1.6, 1.4],

[1.1, 1.3, 0, 1.2, 1.1],

[1.5, 1.6, 1.2, 0, 1.2],

[1.3, 1.4, 1.1, 1.2, 0]])

fuel\_consumption = np.array(

[

[0, 5, 4, 8, 7],

[5, 0, 6, 9, 8],

[4, 6, 0, 5, 4],

[8, 9, 5, 0, 6],

[7, 8, 4, 6, 0]]

)

num\_buses = 3

max\_transfers = 2

max\_emissions = 50 # Example emission cap

# Decision Variables

X = {} # Route assignment

for i in range(num\_stops):

for j in range(num\_stops):

if i != j:

X[i, j] = solver.BoolVar(f'X[{i},{j}]')

T = solver.IntVar(0, max\_transfers, 'T') # Transfers

E = solver.IntVar(0, max\_emissions, 'E') # Emissions

# Objective Function: Minimize cost, transfers, and emissions

solver.Minimize(

solver.Sum(cost\_matrix[i][j] \* X[i, j] \* traffic\_matrix[i][j] for i in range(num\_stops) for j in range(num\_stops) if i != j) +

T + E

)

# Constraints

for i in range(num\_stops):

solver.Add(sum(X[i, j] for j in range(num\_stops) if i != j) == 1)

# Emission Constraints

solver.Add(solver.Sum(fuel\_consumption[i][j] \* X[i, j] for i in range(num\_stops) for j in range(num\_stops) if i != j) <= max\_emissions)

# Solve the model

status = solver.Solve()

if status == pywraplp.Solver.OPTIMAL:

print("Optimal solution found:")

for i in range(num\_stops):

for j in range(num\_stops):

if i != j and X[i, j].solution\_value() > 0.5:

print(f'Bus route: {i} -> {j}')

print(f'Minimum Transfers: {T.solution\_value()}')

print(f'Total Emissions: {E.solution\_value()}')

else:

print("No optimal solution found.")

# Run the optimization

optimize\_extended\_bus\_scheduling()

**Output:**

A screenshot of a computer code

AI-generated content may be incorrect.

# Conclusion

Improving passenger experience, cutting expenses, and increasing operational efficiency all depend on public bus network schedule optimization. To accomplish the following, this research investigated computational optimization and mathematical modeling techniques:   
✅ Reduced journey Costs: We cut down on needless journey lengths by utilizing Mixed Integer Programming (MIP).   
✅ Optimized Bus Routes: Taking demand variations into account, the model allocated buses to routes effectively.   
✅ Fewer Passenger transitions: We improved service quality by reducing route transitions.   
✅ Integrated Traffic Data: For dynamic scheduling, real-time congestion considerations were considered.   
✅ Sustainable Transportation: To encourage environmentally friendly activities, emission limits were included in the expanded model.   
By using visualization approaches, Python scripting, and Google OR-Tools, we were able to effectively illustrate a data-driven strategy for optimizing urban bus networks.

# Appendix B- fill the attached Individual Contribution & AI Usage sheet

Setuben Chaudhari: Code and Report

Heta Chavda: Presentation

David Conial: Presentation and Report