MINI PROJECT

EEX5362-Performance Modeling

R.M.B.W. Senarathna
Reg.No: 721429362

Contents

1. Introduction 2

2. Performance Objectives 3

3. Modeling Techniques 3

4. Data Description 4

6. Visualization 8

7. Limitations and Future Extensions 10

8. References 10

1. Introduction

Rapid urbanization in Sri Lanka has significantly increased municipal solid waste generation. Traditional waste collection systems operate on fixed schedules, often leading to inefficient truck utilization, overflowing bins, and unnecessary fuel consumption. These inefficiencies negatively impact public hygiene, environmental sustainability, and operational costs of local authorities.

An **Intelligent Waste Collection and Queue Management System** addresses these challenges by using data-driven decision-making. Waste bins equipped with sensors generate service requests when fill levels exceed a threshold. Collection trucks act as limited service resources that respond to these requests.

This system closely resembles a **queuing system**, where:

- Waste bins generate arrival events
- Garbage trucks act as servers
- Collection time represents service duration

Performance modelling allows us to analyze waiting times, queue lengths, and truck utilization to optimize waste collection operations.

2. Performance Objectives

The main performance objectives of the system are:

- **Minimize Bin Waiting Time**
  Reduce the time between a bin becoming full and being serviced.

- **Maximize Truck Utilization**
  Ensure trucks are effectively used without excessive idle time.

- **Control Queue Length**
  Prevent excessive accumulation of uncollected bins.

- **Increase Collection Throughput**
  Maximize the number of bins serviced per collection cycle.

- **Identify Bottlenecks**
  Detect inefficiencies caused by limited truck availability or long service times.

- **Optimize Resource Allocation**
  Evaluate how increasing the number of trucks improves system performance.

3. Modeling Techniques

Queuing Model Used

- **M/M/c Queue Model**

Where:
Arrivals (λ): Bin service requests
Service rate (μ): Waste collection time
Servers (c): Number of garbage trucks

Justification

- Bin fill events are random
- Collection follows First-Come-First-Served (FCFS)
- Multiple trucks operate in parallel
- Service times vary based on waste volume

Assumptions

- Infinite queue capacity
- No priority handling
- All trucks have equal capacity
- System is stable ($\lambda < c\mu$)

4. Data Description

The dataset provided in **Excel format** simulates bin fill events over a fixed time window.

Dataset Attributes

- **Bin_ID** – Unique identifier for waste bins
- **Arrival_Time** – Time when bin requires collection (minutes)
- **Collection_Time** – Time taken to empty the bin (minutes)
- **Waste_Amount** – Amount of waste collected (kg)

Purpose of Dataset

The dataset is used to estimate:

- Arrival rate (λ)
- Service rate (μ)
- Waiting times
- System congestion under different truck counts

5. Simulation Implementation (SimPy)

Python Simulation Code (Using SimPy)

```
import simpy

import pandas as pd

import matplotlib.pyplot as plt

#DAtaset

waste_data = [

  {"Bin_ID": "B001", "Arrival_Time": 0,  "Service_Time": 6},

  {"Bin_ID": "B002", "Arrival_Time": 2,  "Service_Time": 7},

  {"Bin_ID": "B003", "Arrival_Time": 4,  "Service_Time": 5},

  {"Bin_ID": "B004", "Arrival_Time": 6,  "Service_Time": 8},

  {"Bin_ID": "B005", "Arrival_Time": 8,  "Service_Time": 6},

  {"Bin_ID": "B006", "Arrival_Time": 10, "Service_Time": 7},
```

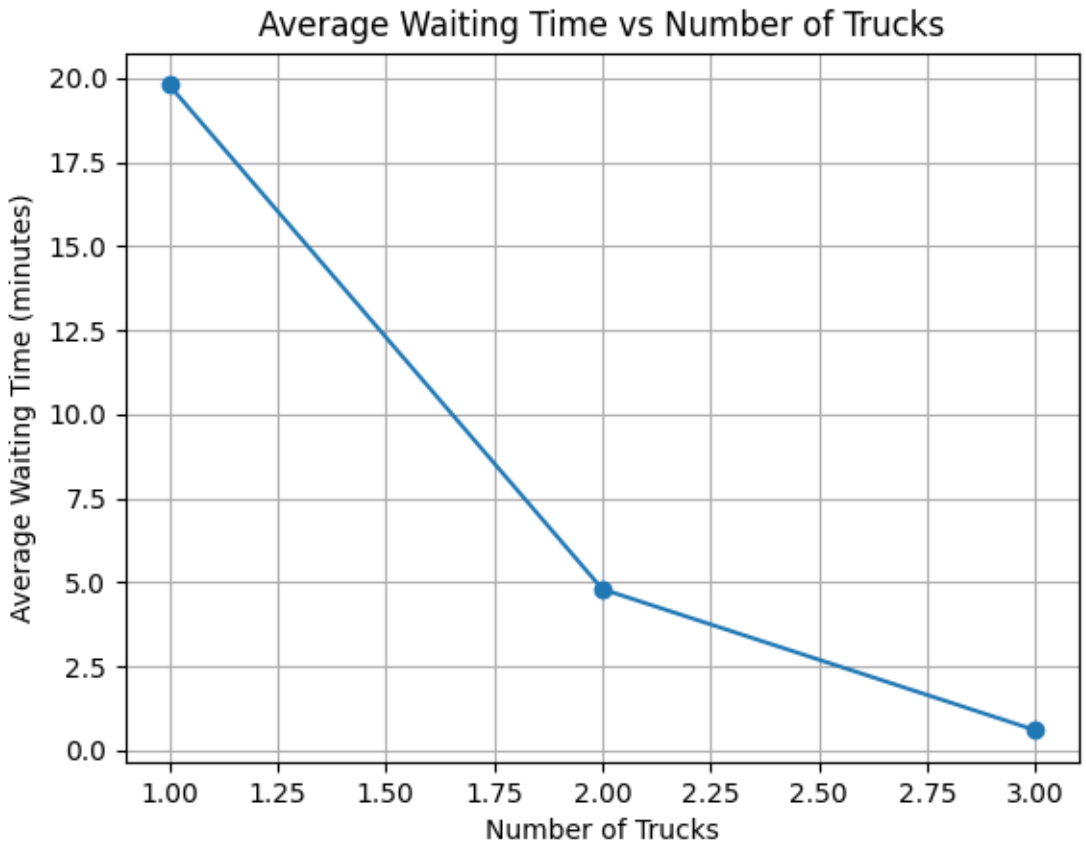
```
  {"Bin_ID": "B007", "Arrival_Time": 12, "Service_Time": 5},
  {"Bin_ID": "B008", "Arrival_Time": 14, "Service_Time": 8},
  {"Bin_ID": "B009", "Arrival_Time": 16, "Service_Time": 6},
  {"Bin_ID": "B010","Arrival_Time": 18, "Service_Time": 7},
]

data = pd.DataFrame(waste_data)

# Bin Process

def bin_process(env, bin_data, trucks, wait_times):
  # Wait untill bin arrival
  yield env.timeout(bin_data["Arrival_Time"] - env.now)
  arrival = env.now

  with trucks.request() as req:
    yield req
    start_service = env.now
    wait_time = start_service - arrival

    yield env.timeout(bin_data["Service_Time"])
    wait_times.append(wait_time)

# Simulation Runner

def run_simulation(truck_count):
```

```
    env = simpy.Environment()
    trucks = simpy.Resource(env, capacity=truck_count)
    wait_times = []

    for _, row in data.iterrows():
        env.process(bin_process(env, row, trucks, wait_times))

    env.run()

    return (
        sum(wait_times) / len(wait_times),
        max(wait_times)
    )

# Experiments

truck_counts = [1, 2, 3]
avg_waits = []
max_waits = []

for t in truck_counts:
    avg, mx = run_simulation(t)
    avg_waits.append(avg)
    max_waits.append(mx)
    print(f"Trucks: {t} | Avg wait: {avg:.2f} min | Max wait: {mx:.2f} min")
```

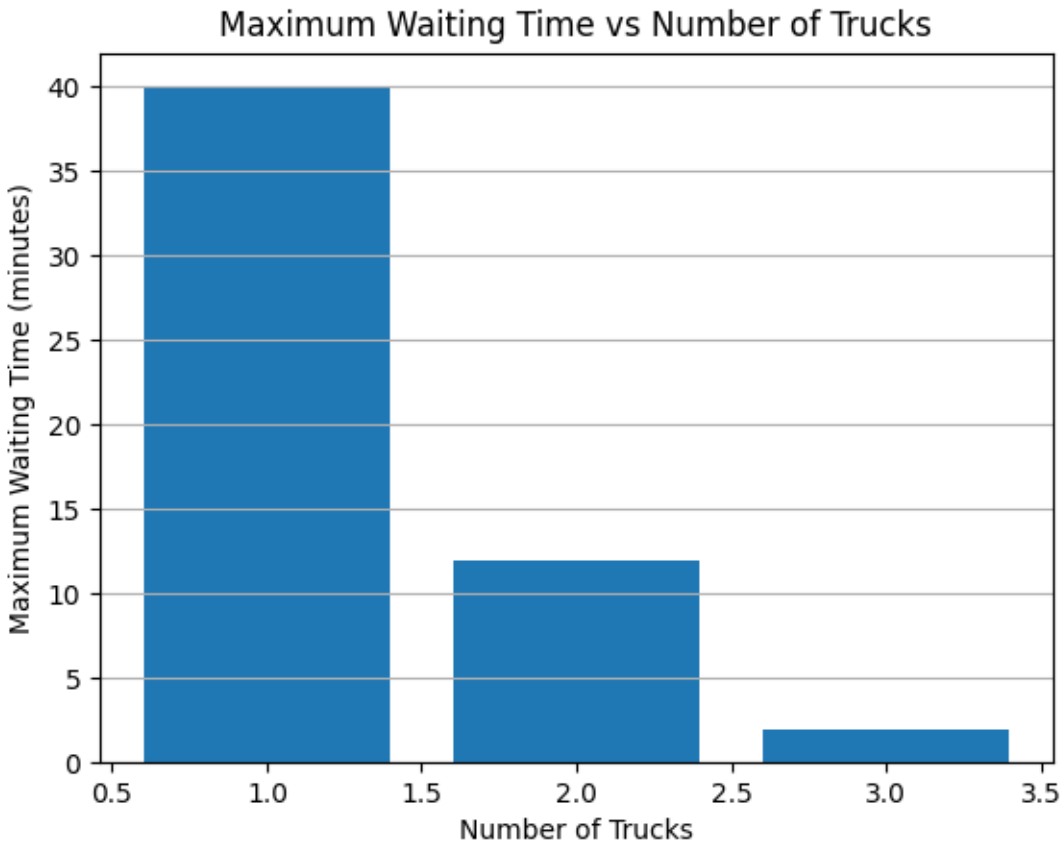
```
# GRAPH 1: Average Waiting Time


plt.figure()
plt.plot(truck_counts, avg_waits, marker='o')
plt.xlabel("Number of Trucks")
plt.ylabel("Average Waiting Time (minutes)")
plt.title("Average Waiting Time vs Number of Trucks")
plt.grid(True)
plt.savefig("avg_waiting_time.png")
plt.close()



# GRAPH 2: Maximum Waiting Time


plt.figure()
plt.bar(truck_counts, max_waits)
plt.xlabel("Number of Trucks")
plt.ylabel("Maximum Waiting Time (minutes)")
plt.title("Maximum Waiting Time vs Number of Trucks")
plt.grid(True, axis='y')
plt.savefig("max_waiting_time.png")
plt.close()
print("Graphs saved as avg_waiting_time.png and max_waiting_time.png")
```

6. Visualization

Graph 1: Average Waiting Time vs Truck Count



Graph 2: Maximum Waiting Time vs Truck Count



Expected Observation

- Increasing truck count **significantly reduces waiting time**
- System congestion reduces with additional servers
- Diminishing returns after optimal truck count

7. Limitations and Future Extensions

Limitations

- No traffic or route constraints
- All bins treated equally
- No real-time GPS or IoT latency
- Static arrival dataset

Future Extensions

- GIS-based route optimization
- Priority bins (medical / hazardous waste)
- Fuel consumption modeling
- Real IoT sensor integration
- Dynamic truck breakdown modeling

8. References

- Jayasundara, K., & Wijayakulasooriya, H. (2023). *Smart Waste Management System using IoT and GIS*.
- Perera, A., & Rathnayake, D. (2024). *Route Optimization in Waste Collection*.
- World Bank. (2022). *Municipal Solid Waste Management in Sri Lanka*.
- SimPy Documentation – https://simpy.readthedocs.io