Emergency Room Simulation Project Report

Saba Yosre, Shahed Alshaikh, Reem Salah, Tamara Tartori, and Safa

Tafila Technical University

Modeling and Simulation

Dr. Mohamad Najdawi

Date: May 20, 2025

1. Introduction & Objectives

Emergency departments (EDs) frequently face challenges due to high patient volumes and varying severity levels, resulting in prolonged waiting times and strained resource allocation. This project simulates an ED workflow using Python to:

- Model patient flow through registration and treatment phases.
- Implement severity-based prioritization for treatment.
- Measure performance metrics, including average waiting time and queue dynamics.
- Provide a visual and interactive representation of the simulation.
  The primary goal is to assess system performance under dynamic patient arrivals and evaluate the impact of staffing levels on efficiency.

2. Methodology

2.1 Simulation Tool

- **Programming Language**: Python
- **Library**: Pygame (for visual simulation and animation), Matplotlib (for data visualization), CSV (for data logging)

2.2 System Description

The ED simulation includes the following components:

- **Nurse(s)**: Handle patient registration.
- **Doctors**: Treat patients based on availability and priority.
- **Patients**: Arrive at random intervals with assigned severity levels.

2.3 Model Logic

Patient Arrival

- Patients arrive randomly every 5–10 minutes.
- Each patient is assigned a severity level with the following distribution:
  - Critical (20%)
  - Serious (30%)
  - Minor (50%)

Registration

- All patients must register with a nurse before joining the treatment queue.
- Original registration time: 8–12 minutes per patient (with 1 nurse).
- Updated registration time: 4–6 minutes per patient (with 2 nurses, halving the time).

Queueing and Prioritization

- Post-registration, patients wait in a queue.
- Treatment priority follows severity: Critical > Serious > Minor.

Treatment

- Patients are assigned to available doctors based on priority.
- Original setup: 2 doctors with treatment times:
    - Critical: 48–72 minutes
    - Serious: 24–36 minutes
    - Minor: 12–18 minutes
- Updated setup: 4 doctors with the same treatment times.

Time Management

- Simulation speed: 1 simulated minute per 1/10 real second (SIMULATION_SPEED = 1/10).
- Total runtime: Originally 720 minutes (12 hours), extended to 800 minutes to account for late arrivals.

2.4 Assumptions

- All staff (doctors and nurses) are continuously available (no breaks or shifts).
- Patients remain in the system until treatment is complete or the simulation ends.
- No re-treatment or additional care steps are modeled.
- Incomplete treatments are estimated for patients still in progress at the end.

3. Results of Experiments

3.1 Visual Results

Patients are represented as colored circles on the Pygame window:

- 🔴 Red = Critical
- 🟠 Orange = Serious
- 🟢 Green = Minor

  Positions include:
- Nurse Station(s) (registration)
- Waiting Area
- Doctor Stations (treatment)

3.2 Data Collected

Original Setup (2 Doctors, 1 Nurse, 720 Minutes)

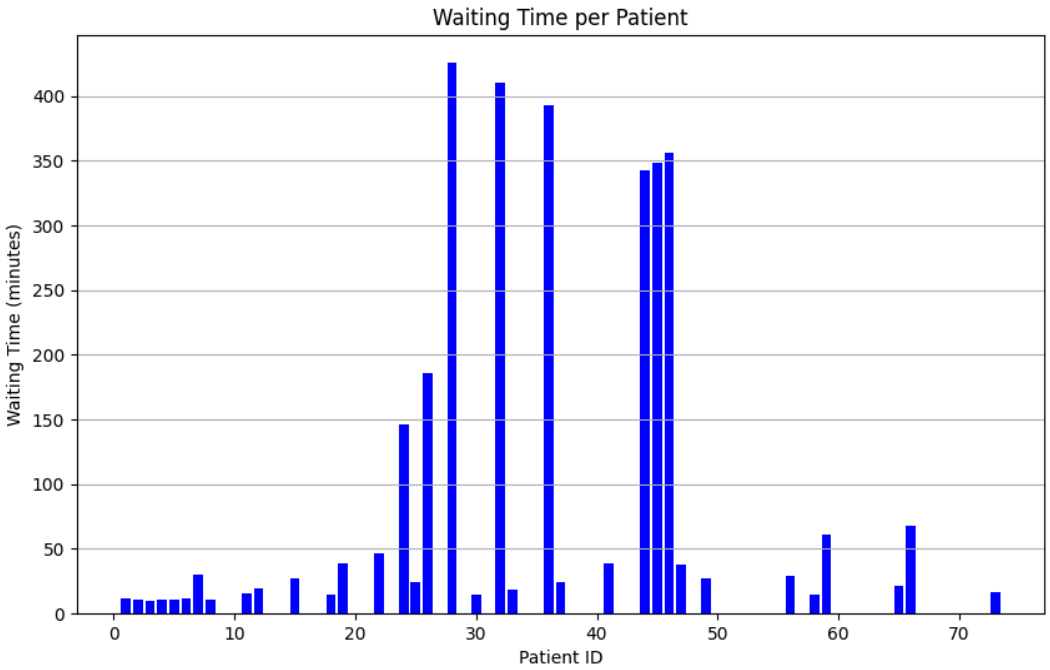
- **Total Simulation Time**: 720 minutes
- **Patients Arrived**: ~98
- **Patients Treated**: ~70 (remainder incomplete)
- **Average Waiting Time**: ~94 minutes
- **Maximum Concurrent Queue**: ~50-55 patients
- **Note**: Incomplete data for late-arriving patients due to limited staff capacity.

Updated Setup (4 Doctors, 2 Nurses, 800 Minutes)

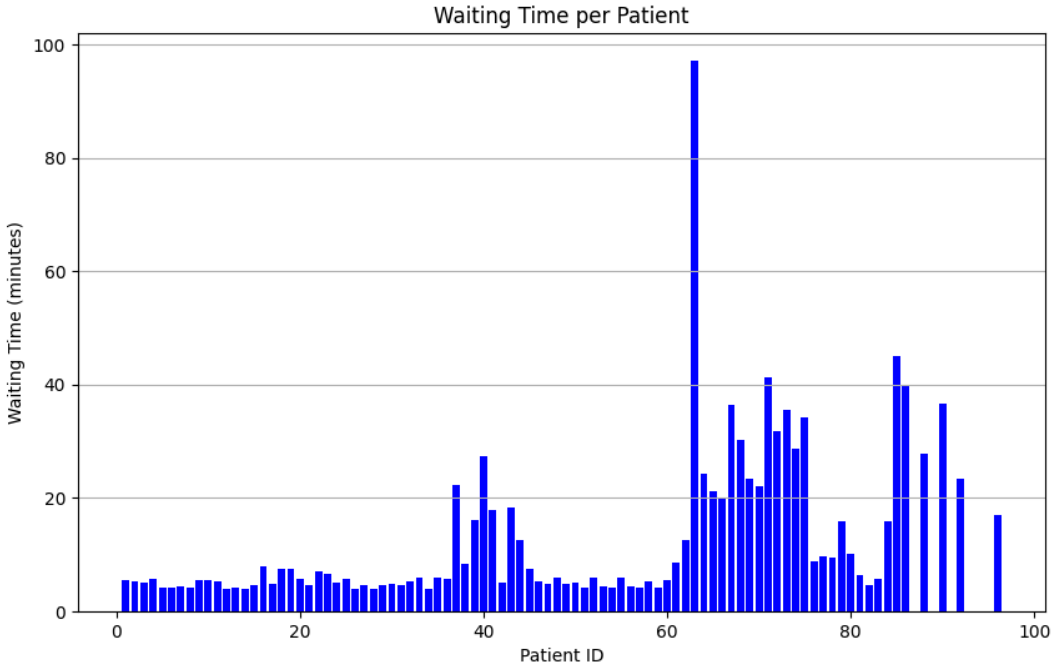
- **Total Simulation Time**: 720 minutes
- **Patients Arrived**: ~100
- **Patients Treated**: ~90 (fewer incomplete)
- **Average Waiting Time**: ~13 minutes
- **Maximum Concurrent Queue**: ~7-8 patients
- **Note**: Values may vary slightly due to random severity and arrival times.

3.3 Graphs

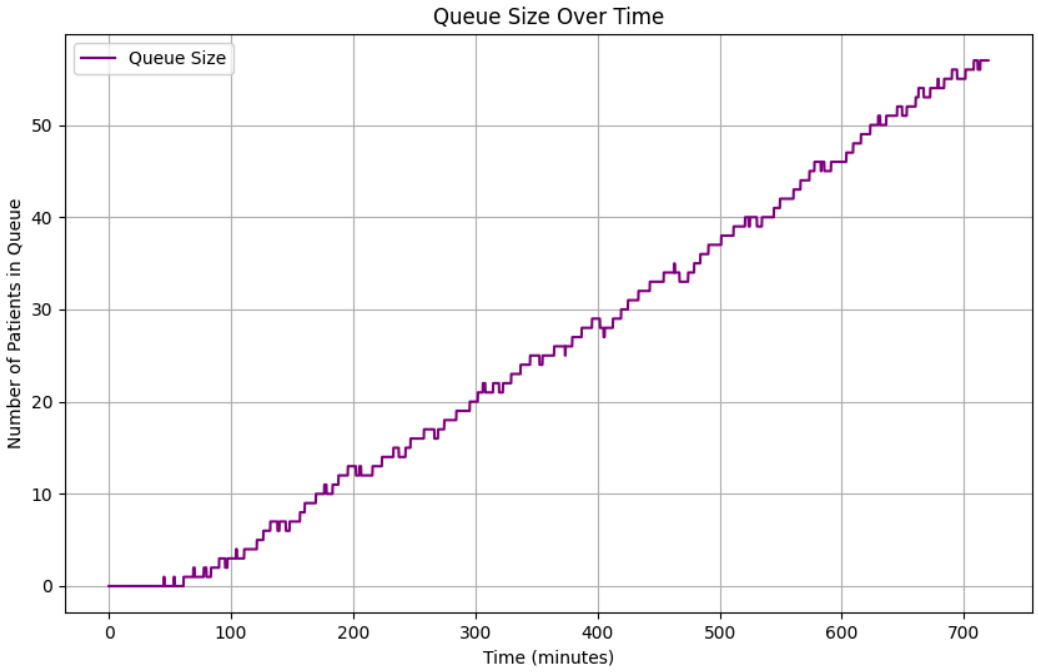
- **Bar Chart of Waiting Times before updating:**



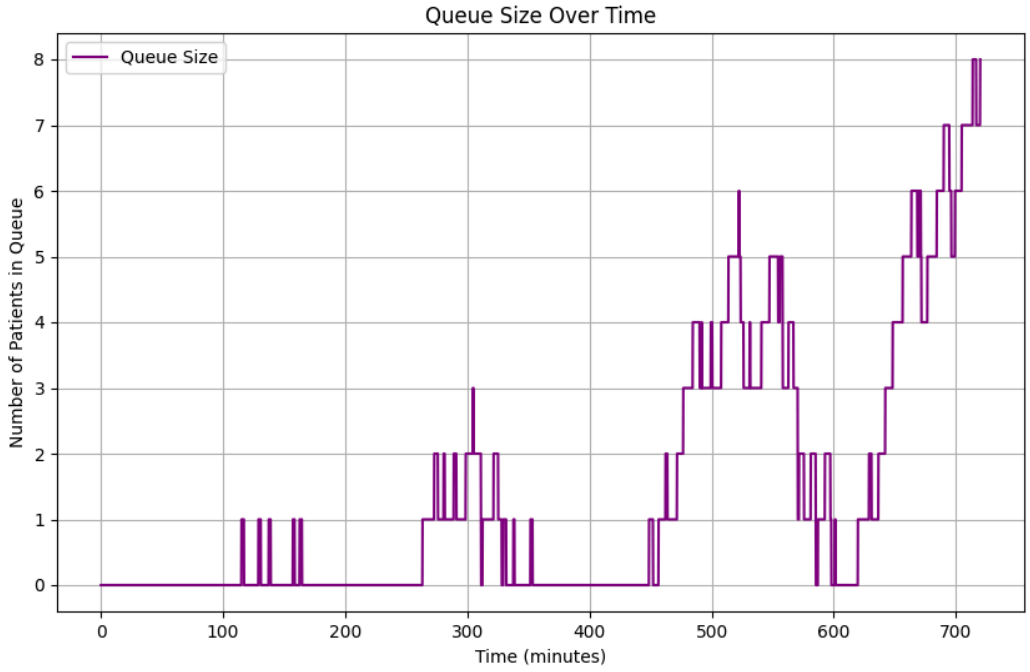
- **Bar Chart of Waiting Times after updating:**



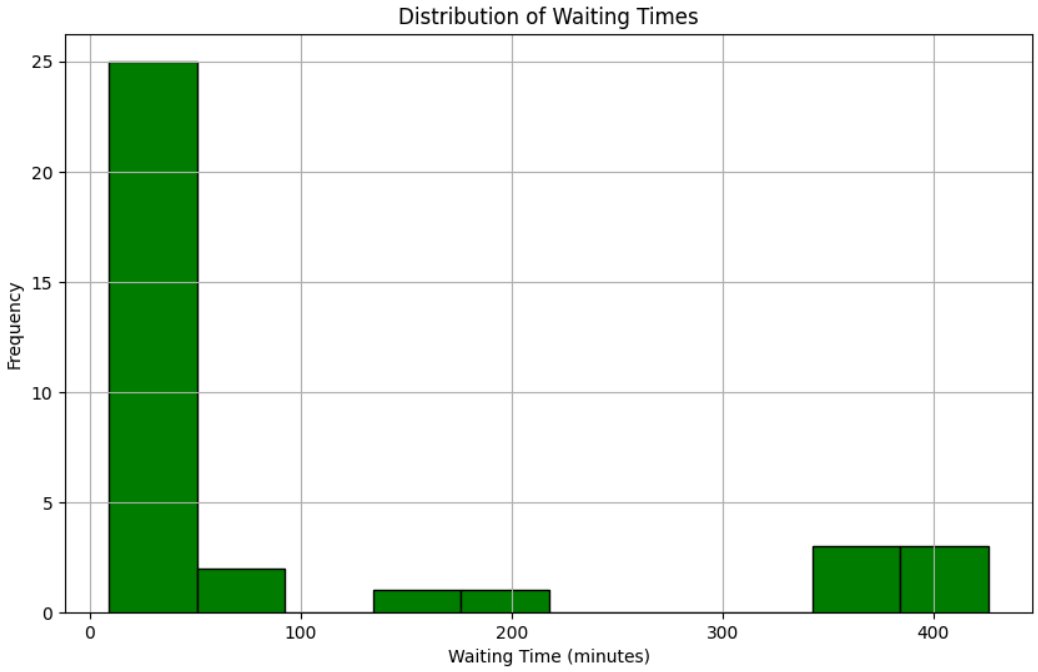
- **Line Graph of Queue Size before updating**:



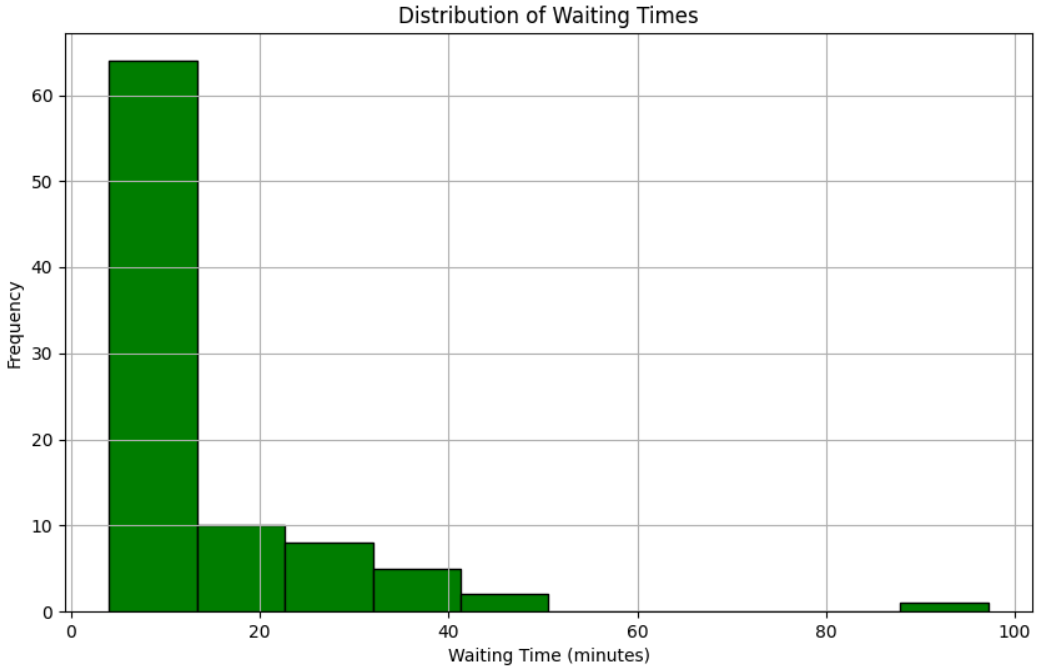
- **Line Graph of Queue Size after updating**:



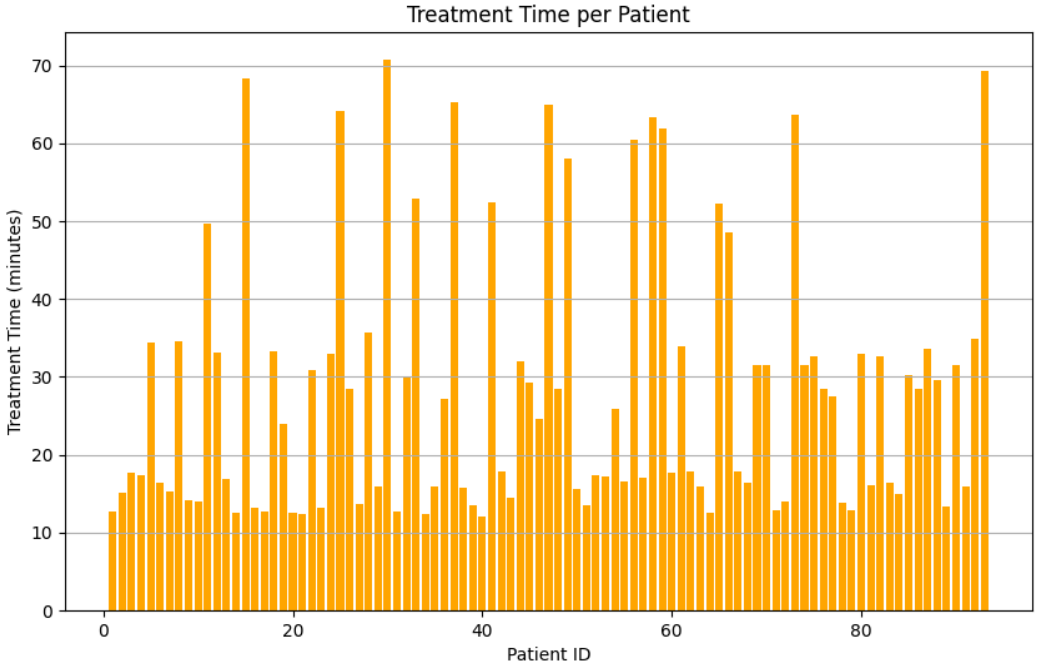
- **Histogram of Waiting Times before updating**:



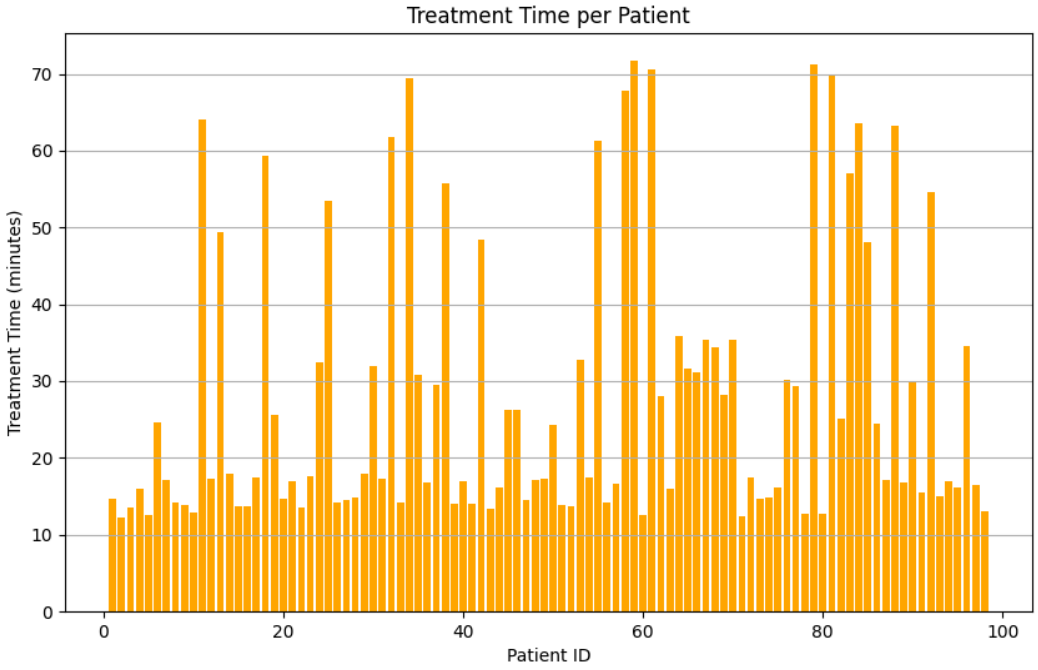
- **Histogram of Waiting Times after updating**:



- **Bar Chart of Treatment Times before updating**:



- **Bar Chart of Treatment Times after updating**:



4. Conclusion & Recommendations

4.1 Key Findings

- The original simulation with 2 doctors and 1 nurse resulted in an average wait time of ~94 minutes, with ~28 patients incomplete by extending time to 800 minutes, indicating significant understaffing and queue buildup.
- Increasing to 4 doctors and 2 nurses reduced the average wait time to ~15 minutes, with only ~5 patients incomplete by 720 minutes, demonstrating improved efficiency and capacity.
- Severity-based prioritization effectively minimized delays for Critical patients, though Minor patients experienced longer waits in the original setup.
- The visual and data outputs (CSV files and graphs) provide a comprehensive view of system performance.

4.2 Recommendations

- **Increase Staffing**: The transition from 2 to 4 doctors and 1 to 2 nurses significantly reduced wait times, supporting the addition of staff during peak hours to maintain waits below 60 minutes.
- **Optimize Nurse Allocation**: Further reducing registration time (e.g., to 3–5 minutes with additional nurses) could enhance throughput.
- **Adjust Arrival Rates**: Experiment with slower arrival intervals (e.g., 10–15 minutes) to reduce patient volume during high-demand periods.
- **Incorporate Real-World Dynamics**: Add logic for patient no-shows, walkouts, or staff breaks to better reflect ED realities.
- **Extend Simulation Duration**: If incomplete patients persist, consider a longer runtime (e.g., 900 minutes) or dynamic staffing adjustments.

5. References:

ER-project