

# Computer Aided Simulation Lab

## Simulation of a Car Sharing System

Labs 3 and 4

Riccardo Bracciale  
S338616

November 4, 2025

### **Laboratory Objectives:**

The purpose of this lab is to develop a simulation model that captures the dynamics of a car-sharing system.

In this system:

- Users appear at various random locations and, using a smartphone app, identify and reserve an available car nearby.
- After reserving, users can drive the car to their destination and leave it either in designated parking spots (e.g. rechargeable stations) or any available space.
- Cars may be partially relocated to better align with demand when necessary.

## Introduction

The purpose of this laboratory is to develop a simulation model that captures the operational dynamics of a **car-sharing system** as described above.

The following sections identify the fundamental system parameters, define the main performance indicators, outline the conceptual model design, and describe the data structures required to support the simulation.

# 1 Model Design

**Key Design Principle:** The simulation model is based on **discrete-event simulation** principles, where the system evolves through a sequence of discrete events rather than continuous time steps.

The whole system is represented as a collection of **entities** (users, vehicles, charging stations) that interact over time through a series of **events** (user arrivals, vehicle reservations, trip completions, charging operations, and relocations).

## 1.1 User Behavior Model

Specifically, users are expected to sign up for the platform. Once registered, users will randomly make requests to rent a car, with random destinations inside the service area.

### Reservation Process:

1. User makes a car request at a random location
2. System searches for available cars within a search radius
3. User is assigned to the *closest available car*
4. If no cars are available, user retries (up to a maximum number of attempts)
5. After exhausting attempts, user abandons the request

### Rental Process:

1. After a successful reservation, the user will get to the car's location after a certain walking time, which depends on the distance between the user's initial position and the car's position.
2. Then the user drives the car to the desired destination, the time taken will depend on the **traffic conditions** between the two points, which can change over time.
3. During the trip, the car uses up energy based on the distance driven.
4. If a car's battery is below a certain level at the end of a rental, it will be taken to be charged by a special worker called a *relocator*. This person will pick up the car and drive it to the nearest charging station.

## 1.2 Entities

The main entities of the car-sharing system are as follows:

- **User:** contains information such as current location, desired destination, reservation status, and time of arrival.

- **Vehicle:** stores attributes including current position, battery level, state (available, in use, charging, or reserved), and time since last relocation.
- **Car Relocator:** a special worker responsible for relocating low-battery vehicles to charging stations.

### 1.3 Events

The simulation operates through a series of discrete events that represent key moments in the system's operation:

- **User Subscription Event:** A new user joins the car-sharing platform and becomes eligible to make reservations. This event triggers the scheduling of the user's first reservation attempt.
- **Reservation Event:** A user attempts to find and reserve an available car near their current location. The system searches for the nearest car within an acceptable walking distance. If successful, the user proceeds to pick up the car; if no car is available, the user may retry after some time.
- **Pickup Event:** A user walks to the reserved car's location and begins using it. This marks the start of a trip and records the walking time from reservation to pickup.
- **Dropoff Event:** A user completes their trip and leaves the car at the destination. The system updates the car's location and battery charge based on the distance traveled. If the battery is low, the car is marked for relocation to a charging station.
- **Relocate Car Event:** A car with low battery is assigned to a relocator (a worker who drives cars to charging stations). The system finds an available relocator and the nearest charging station, then schedules the car's arrival at the station.
- **Arrive at Station with Relocator Event:** A relocator delivers a car to a charging station. The car enters the charging queue or begins charging immediately if a spot is available.
- **Charging Complete Event:** A car finishes charging and its battery is fully restored. The car becomes available again for user reservations.

These events form a continuous cycle: users subscribe, make reservations, use cars, and return them, while the system manages car relocation and charging to maintain service availability.

## 2 Assumptions about the Model

---

### Important Note

All modeling decisions involve trade-offs between realism and computational tractability. The following assumptions define the boundaries of our simulation model.

In order to create a manageable and effective simulation model for the car-sharing system, several key assumptions are made:

## 2.1 Service Area and Traffic Zones

The car-sharing service operates within a  $100 \text{ km} \times 100 \text{ km}$  urban area divided into multiple traffic zones. Each zone has distinct traffic characteristics that affect vehicle travel times and user experience:

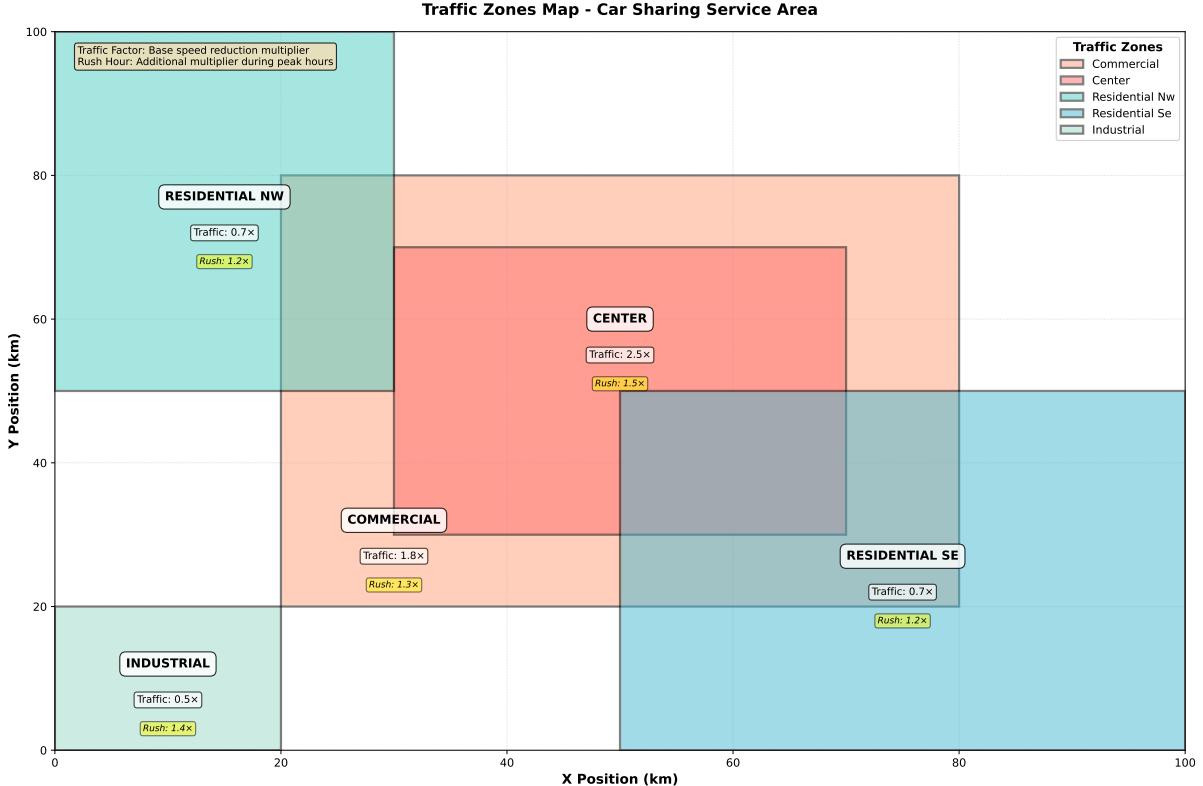


Figure 1: **Traffic zones in the service area** - The map shows five distinct zones with different traffic patterns. The traffic factor indicates how much slower vehicles move compared to free-flow conditions, while the rush hour multiplier shows additional slowdown during peak hours.

- **Center Zone** (red): The city center experiences the heaviest congestion with a base traffic factor of  $2.5\times$ , meaning vehicles move 2.5 times slower than normal. During rush hours, this increases by an additional  $1.5\times$ .
- **Commercial Zone** (orange): Overlapping with the center, this area has medium-high traffic ( $1.8\times$ ) with moderate rush hour increases ( $1.3\times$ ).
- **Residential Zones** (blue/teal): Northwest and southeast residential areas have lighter traffic ( $0.7\times$ ) with modest rush hour increases ( $1.2\times$ ).
- **Industrial Zone** (green): The industrial area in the southwest corner has minimal traffic ( $0.5\times$ ) but experiences significant rush hour congestion ( $1.4\times$ ) due to shift changes.

These traffic patterns are incorporated into the road network simulation to provide realistic travel time estimates throughout the day.

## 2.2 Road Network Graph

To calculate realistic distances and travel times, the simulation uses a **graph structure** to represent the city's road network. This graph consists of:

- **Nodes** (intersections): Points where roads meet, placed on a grid across the city
- **Edges** (roads): Connections between nodes that vehicles can travel along

**Why do we need a graph?** Without a graph, the simulation would calculate distances "as the crow flies" (straight lines). But in real cities, cars must follow roads, making actual distances longer than straight-line distances.

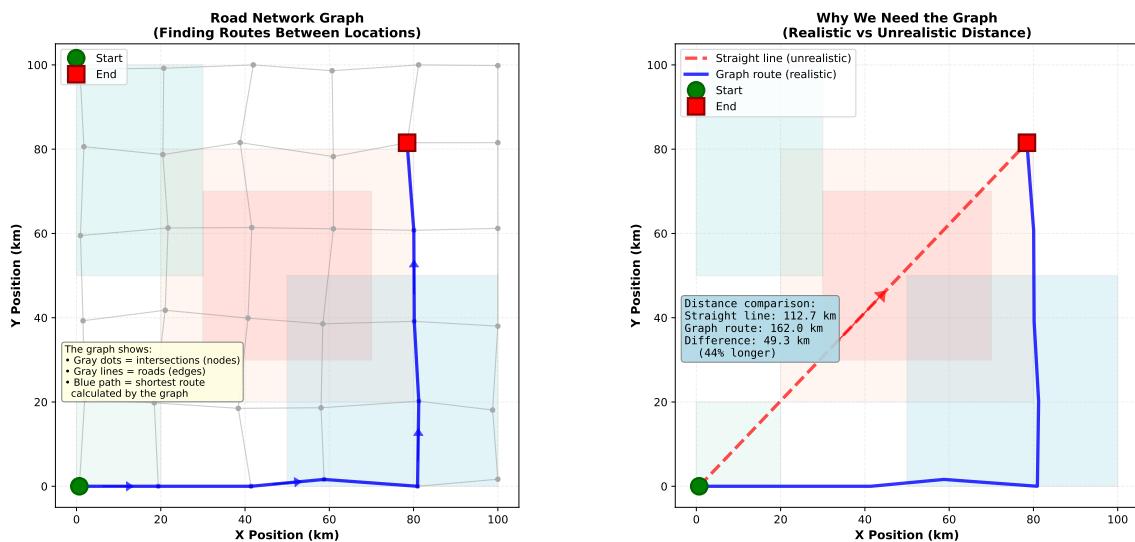


Figure 2: **Road network graph and route calculation** - Left: The graph structure with nodes (intersections) and edges (roads). The blue path shows the shortest route calculated by the graph algorithm. Right: Comparison between unrealistic straight-line distance and realistic graph-based distance following actual roads.

### How the graph works:

1. When a user needs to travel from point A to point B, the system finds the nearest graph node to each point
2. An algorithm calculates the shortest path through the graph (the route with the smallest total distance)
3. The system adds up the distances along all roads in this path to get the total travel distance
4. Traffic factors from the zones are applied to calculate the actual travel time

This approach makes the simulation much more realistic because:

- ✓ Vehicles follow actual road paths instead of impossible straight lines

- ✓ Distances are 15-40% longer than straight-line distances (as in real cities)
- ✓ Different routes can have different traffic conditions
- ✓ The system can find the fastest path considering both distance and traffic

### 3 Key Parameters and Performance Indicators

The system is characterized by a set of parameters that define its operational configuration and user behavior. These parameters form the foundation upon which the simulation is built and determine the relationships between demand, supply, and service performance.

#### 3.1 System Parameters

Parameter Category	Parameter Description & Impact
keycolor!10 <b>Fleet Size</b>	Total number of vehicles available in the system. Determines potential capacity and directly influences availability and utilization. Critical for balancing service quality vs. operational costs.
gray!5 <b>User Arrival Rate</b>	Rate at which users appear (Poisson process with time-varying mean rate). Affects system load, congestion, and reservation success probability.
keycolor!10 <b>Trip Distribution</b>	Spatial distribution of origins/destinations and trip distances. Affects vehicle circulation patterns, energy consumption, and geographic availability.
gray!5 <b>Charging Infrastructure</b>	Number and location of charging stations, plus charging rate (7.2 kW). Constrains vehicle availability due to downtime and creates potential bottlenecks.
keycolor!10 <b>Relocation Strategy</b>	Frequency and magnitude of vehicle relocations by dedicated workers. Influences response to geographical demand imbalances and operational efficiency.

Table 1: **Key system parameters influencing performance** - Each parameter can be varied in scenario analysis to assess impact on service quality

Parameter	Baseline Value	Range/Notes
Fleet size	<b>20 cars</b>	10-40 (scalability tests)
User arrival rate	<b>0.15 users/hr</b>	Time-varying (rush hours)
Max registered users	<b>50,000</b>	Platform capacity limit
Search radius	<b>2 km</b>	Walking tolerance threshold
Charging stations	<b>5 stations</b>	Spatially distributed
Charging rate	<b>7.2 kW</b>	Level 2 AC charging
Battery capacity	<b>60 kWh</b>	Per vehicle maximum
Energy consumption	<b>0.15 kWh/km</b>	Realistic EV rate
Number of relocators	<b>3 workers</b>	Automated rebalancing

Table 2: **Detailed parameter specifications** - Baseline configuration used for simulation experiments

## 4 Performance Indicators & Target Values

To evaluate the functioning of the car-sharing system, a comprehensive set of key performance indicators (KPIs) is defined to measure service quality, operational efficiency, and user satisfaction:

KPI Category	Description & Measurement	Target Value
keycolor!10 Availability Rate	Fraction of user requests that successfully find a nearby vehicle within search radius. Measures service coverage and demand satisfaction.	> 80%
gray!5 Average Waiting Time	Mean time elapsed between user arrival and successful car acquisition (includes search and reservation). Critical for user experience.	< 15 min
keycolor!10 Vehicle Utilization	Percentage of time vehicles spend in active use vs. idle or charging. Indicates asset efficiency and fleet sizing.	60-80%
gray!5 Charging Occupancy	Proportion of time charging stations are occupied. High values indicate potential bottlenecks in energy infrastructure.	< 80%
keycolor!10 Relocation Efficiency	Ratio of service improvement to operational cost of relocating vehicles. Balances proactive rebalancing vs. expenses.	Maximized

Table 3: **Key Performance Indicators with target values** - Metrics used to assess system performance across different scenarios

#### Important Note

**Note:** These KPIs are continuously monitored during simulation and used for transient detection and steady-state analysis. Target values represent industry-standard expectations for car-sharing services.

Together, these indicators provide a comprehensive picture of both **user satisfaction** and **operational performance**.

## 5 Entities

In this section, we examine the main entities of the car-sharing system and their interactions.

- **User:** contains information such as current location, desired destination, reservation status, and time of arrival.
- **Vehicle:** stores attributes including current position, battery level, state (available, in use, charging, or reserved), and time since last relocation.

- **Charging Station:** characterized by its geographic coordinates, number of charging points, queue of waiting vehicles, and current utilization level.

## 5.1 Core Data Structures

To manage these entities and their interactions efficiently, several supporting data structures are employed:

- A **priority queue** to handle the chronological sequence of events, ensuring that the simulation processes occurrences in correct temporal order.
- A **spatial index** or lookup structure (for example, a grid or k-d tree) to enable rapid identification of the nearest available vehicle for a user request.
- A **vehicle list** or dictionary to maintain global vehicle states and allow efficient updates during trips, charging, or relocation.
- A **charging queue** for each station, storing vehicles waiting to recharge and managing service order.

These structures collectively enable an efficient event-driven simulation capable of managing large numbers of vehicles and users while preserving spatial and temporal realism.

## 6 Experimental Scenarios and Results

To evaluate the car-sharing system under different operating conditions, we designed several scenarios by varying key parameters. Each scenario tests a specific aspect of system performance and helps us understand how different factors affect service quality.

### 6.1 Scenario Descriptions

The following scenarios were tested, each modifying specific parameters from the baseline configuration:

Scenario	Parameter Changed	Purpose
gray!5 <b>Baseline</b>	Default values: • 20 cars • 0.15 users/hour • 5 stations • 20 km pickup radius	Establish reference performance for comparison
keycolor!10 <b>High Demand ×2</b>	User arrival rate doubled: 0.15 → 0.30 users/hour	Test system behavior under increased demand stress
gray!5 <b>Large Fleet ×2</b>	Fleet size doubled: 20 → 40 cars	Evaluate impact of increased vehicle supply on service quality
keycolor!10 <b>Wide Pickup Radius</b>	Maximum pickup distance: 20 km → 45 km	Assess effect of users willing to walk further
gray!5 <b>Fewer Stations</b>	Charging stations reduced: 5 → 2 stations	Test resilience to charging infrastructure constraints

Table 4: **Experimental scenarios** - Each scenario isolates the effect of one parameter on system performance

### 6.2 Key Results and Observations

Based on simulation results, we can draw several important conclusions about system behavior:

**Main Finding:** The car-sharing system's performance is most sensitive to the balance between vehicle supply (fleet size) and user demand (arrival rate). Infrastructure changes have secondary but still significant effects.

#### Scenario-Specific Observations:

1. **High Demand ×2:** When user arrival rate doubles, the system experiences:

- *Lower success rate:* More reservation failures because cars are often in use
- *Higher attempts before success:* Users need to retry more times before finding a car
- *Increased utilization:* Cars spend more time in use and less time idle
- *Longer wait times:* More competition for available vehicles

**Lesson:** The system becomes congested - demand exceeds supply capacity.

## 2. Large Fleet ×2:

Doubling the number of cars results in:

- *Higher success rate:* Nearly all reservation requests succeed
- *Fewer retry attempts:* Cars are readily available
- *Lower utilization:* Many cars sit idle waiting for users
- *Minimal wait times:* Users quickly find nearby vehicles

**Lesson:** More cars improve service quality but reduce efficiency (many idle vehicles).

## 3. Wide Pickup Radius:

Allowing users to walk further shows:

- *Improved success rate:* More cars become "reachable" for each user
- *Longer walking times:* Users must walk further to reach cars
- *Better spatial coverage:* Cars in remote areas become usable

**Lesson:** Flexible users can compensate for limited vehicle density, but at the cost of convenience.

## 4. Fewer Stations:

Reducing charging infrastructure creates:

- *Longer charging queues:* Cars wait longer to charge
- *Higher station utilization:* Stations operate near capacity
- *More unavailable cars:* Cars stuck charging cannot serve users
- *Potential bottleneck:* System performance degrades if stations are overwhelmed

**Lesson:** Charging infrastructure must scale with fleet size to avoid bottlenecks.

## 6.3 Design Implications

These results suggest several practical guidelines for system design:

**Operational Guidelines:**

- **Fleet sizing:** Aim for 60-80% utilization - too high indicates insufficient capacity, too low wastes resources
- **Charging infrastructure:** Provide at least 1 station per 5-8 vehicles to prevent charging bottlenecks
- **Service radius:** Balance between coverage (larger radius) and convenience (shorter walking distance)
- **Demand management:** During peak periods, consider dynamic pricing or reservation prioritization

The scenarios demonstrate that operating a car-sharing system requires careful tuning of multiple parameters. Simply adding more cars is not always the solution - it improves service quality but reduces economic efficiency. The optimal configuration depends on the specific goals: maximizing user satisfaction, minimizing costs, or finding a balance between the two.

## 7 Conclusion

The conceptual design developed in this laboratory defines the main elements necessary to simulate a car-sharing system with electric vehicles. It identifies the key parameters influencing system behavior, the performance indicators used to evaluate efficiency and service quality, and the essential dynamics governing user and vehicle interactions.

The proposed data structures and assumptions provide a solid foundation for subsequent implementation and experimentation. Future extensions may include refined user behavior models, predictive relocation strategies, and the introduction of time-dependent demand to further approximate real-world conditions.