STAGE–1 REPORT

Existing Research Work on EV Charging Station Placement, Sizing, and Routing

---

1. Introduction

Electric Vehicle (EV) charging infrastructure planning has become an important research direction due to the rapid adoption of EVs and limited availability of fast-charging stations. Numerous research works have attempted to solve different sub-problems such as **charging station placement**, **charger sizing**, **driver routing**, and **waiting time minimization**.
However, most approaches treat these problems **independently**, whereas real-world EV charging involves both long-term decisions (station locations) and short-term operational decisions (queue management).
This section reviews all major research work done so far related to these aspects.

---

2. Research Work on Charging Station Placement

2.1 Classical Facility Location Approaches

Early research treated EV station placement as traditional facility location problems such as:

- **p-median problem**
- **Set-covering problem**
- **p-center minimization**

These models aim to minimize user travel distance or maximize area coverage. Many urban-planning studies use mathematical optimization for these problems.

2.2 Metaheuristic Approaches (GA, PSO, ACO)

Recent studies increasingly use metaheuristics for placement because the search space is large and NP-hard.

✅ **Genetic Algorithms (GA)**
A large number of papers show GA is effective in selecting optimal locations because:

- Chromosomes can encode candidate station sites
- Fitness can incorporate cost, distance, and grid constraints
- GA handles multi-objective trade-offs

✅ **Particle Swarm Optimization (PSO)** and
✅ **Ant Colony Optimization (ACO)**
also appear in literature but mostly solve static placement only.

2.3 Clustering-Based Approaches

K-means and density-based clustering are used to:

- Identify hotspots of EV trip demand
- Suggest station clustering zones
- Reduce total detour

But clustering fails to account for “waiting time” or congestion.

---

3. Research Work on Charger Sizing and Capacity Planning

3.1 Static Capacity Allocation

Some studies focus on determining how many chargers to install at each site.
These models usually combine:

- Demand forecasting
- Fixed-size M/M/1 or M/M/k queues
- Linear programming

However, these methods assume steady demand and **cannot adjust dynamically** as EV arrivals fluctuate.

3.2 Queueing-Based Sizing Models

Queueing models attempt to minimize waiting time by estimating:

- Arrival rate (λ)
- Service rate (μ)
- Utilization (ρ)

But these models are **static**, require accurate λ prediction, and break down under random spikes or rush hours.

---

4. Research Work on EV Routing and Waiting-Time Minimization

4.1 Routing Based Only on Distance

Traditional approaches send drivers to nearest stations.
This results in:

- Overloaded central stations
- Long queues
- Underutilized peripheral stations

4.2 Real-Time Routing Using Heuristics

Some researchers propose dynamic routing using:

- Greedy heuristics
- Load-balancing heuristics

But these do not learn from the environment and react slowly to changes.

---

5. Reinforcement Learning (RL) in EV Charging Optimization

5.1 DRL for Scheduling and Queue Management

Modern RL models (DQN, A2C, PPO) are used for:

- Reducing queue lengths
- Smart charging schedules
- Dynamic price-based charging

- Allocation of mobile charging units

These studies show RL can significantly reduce waiting time.

✅ **But RL is used mostly for:**

- Scheduling when EVs plug in
- Demand response
- Pricing policies

❌ **But NOT for selecting station locations**

RL literature does not integrate long-term placement with RL-based short-term operations.

---

6. Hybrid Approaches (Metaheuristic + Learning)

6.1 Current Hybrid Trends in Literature

Some hybrid methods combine:

- GA + Simulated Annealing
- PSO + ACO
- GA + Fuzzy systems

These only improve static placement.
There are **almost no studies** combining GA for placement and RL for operational optimization.

6.2 RL for Station Placement (Rare)

A few very recent papers attempt RL-based siting, but:

- They require very complex state definitions
- They struggle with large action spaces
- They ignore dynamic queueing and charger sizing

Thus they cannot replace GA for location selection.

---

7. Research Gap Identified

Through reviewing existing literature, the following major gaps are clear:

✅ **Gap 1: Placement and sizing rarely integrated**

Most works solve:

- location → separately
- sizing → separately
- queue minimization → separately

No unified framework exists.

✅ **Gap 2: GA is strong for placement but static**

GA finds optimal geographic locations but **cannot handle dynamic customer arrivals**.

✅ **Gap 3: RL excels at real-time decisions but is not used with GA**

RL could:

- Route EVs intelligently
- Dynamically adjust charger counts
- Reduce waiting time
  …but it cannot handle the large search space of long-term planning.

✅ **Gap 4: No known work combines GA with Q-learning**

A simple, explainable RL method like Q-learning is missing from EV infrastructure models.

✅ **Gap 5: Queueing uncertainty not fully addressed**

Most placement models assume fixed arrival rates, which is unrealistic.

---

8. Why Our Hybrid GA + Q-Learning Approach is Needed

Based on gaps above, our approach:

✅ **Uses GA for long-term decisions**

- Finds optimal 6–8 locations out of 30
- Minimizes detour and installation cost
- Incorporates grid-loss proxy

✅ **Uses Q-learning for short-term decisions**

- Learns routing policy
- Learns charger allocation policy
- Reduces Mean Charging Wait Time (MCWT)

✅ **Integrates both into a unified framework**

Something missing in all literature.

✅ **Advantages of Our Model**

- Dynamic
- Scalable
- Explainable
- Beginner-friendly to implement
- Novel for research and publication

---

9. Final Summary of Stage–1 Review

Existing literature has extensively explored:

- GA/PSO/ACO for placement
- Queueing models for capacity

- RL for scheduling

But *no research* so far:

✅ Integrates GA + RL
✅ Handles uncertainty in EV arrival rates
✅ Combines placement + sizing + routing
✅ Uses simple Q-learning for dynamic optimization
✅ Applies the framework to a small synthetic grid (beginner-friendly)

Thus, our hybrid approach fills a clear research gap.

STAGE-II REPORT

EV Charging Station Placement Optimization — Progress Report

1. Status (what I did so far)

- Performed a focused literature search for recent and influential papers on EV charging station siting, multi-objective formulations (detour vs grid stress), and heuristic/GA solutions.
- Collected candidate datasets (national station registries and urban spatio-temporal charging datasets).
- Drafted the Problem Statement (PS) for clarity and structured next steps/system architecture, data plan, and experimental setup.

---

2. Selected papers

1. *A systematic literature review of optimal placement of fast charging stations* — recent survey; good to get state-of-the-art, common metrics, and benchmarks.
2. *Optimal location of electric vehicle charging stations using* ... — numerical methods applied to urban networks; includes capacity constraints.
3. *Facility Location Optimization of Battery Electric Vehicle* (IIMA tech report) — flow-capturing approach and detour modeling; useful for demand modeling.
4. *An exact approach for the charging station location* — shows integer programming formulations (useful for small instances / benchmarking).
5. Dimitriou P., *An exact approach for the charging station location ...* (2025) — integer programming approach for small/benchmark cases
6. Prommakhot S., *Optimal Placement of Electric Vehicle Stations Using High-...* (2025) — hybrid GA methods and practical decoding heuristics.

Open datasets & APIs (for locations and charging sessions)

- NREL / AFDC — Alternative Fuel Stations data and API (station locations & metadata).
- *UrbanEV / UrbanEV dataset papers* — open benchmark datasets for urban EV charging behavior (spatio-temporal usage).

7. *Hybrid GA/PSO/TSP approaches for fast-charging station placement* — several applied heuristic methods; good for algorithm baselines.
8. ST-EVCDP GitHub — spatio-temporal EV charging datasets (Shenzhen).
9. Mendeley EV Charging Dataset — session-level charging logs (timestamps, kWh, durations).
10. Data.gov / Alternative Fuel Stations catalog — another national repository for station metadata.

3. Defined Problem Statement

Objective: Given a candidate set of locations (L), road network and traffic flows, and the distribution network topology, choose a subset of sites and charger types (slow/fast) and assign expected demand so as to **minimize total driver detour and grid stress**, subject to coverage, station capacity, and budget constraints.

Decision variables:

- Binary placement variables (x_k in{0,1}) for each candidate site (k in L).

- Integer/continuous variables for number of chargers of each type at each site.
- Assignment variables linking demand flows to chosen stations.

Our MILP defines the exact objective function mathematically.
GA uses that same objective as a fitness function to scale up to the whole city.
Q-learning then optimizes real-time EV routing and charger allocation after the stations are built, minimizing waiting time.
The combination of strategic planning (GA) and operational learning (Q-learning) makes our model both dynamic and novel.

Objective (multi-objective / weighted) for MILP (low-level) AND GA(macrolevel):

$$\min \left[\underbrace{\sum_{i,k} w_i d_{ik} y_{ik}}_{\text{Driver Detour}} + \beta \underbrace{\sum_{j} \sum_{k} g_{kj} x_k}_{\text{Grid Stress}} + \gamma \underbrace{\sum_{k} c_k x_k}_{\text{Installation Cost}} \right]$$

Q-learning [for optimising mean charge waiting time]:

Reward function, R = -(waiting_time) – α(queue_length) – β(overload)

Constraints:

- Coverage: each demand point has at least one station within distance (D_{max}).
- Capacity: assigned charging demand to site (k) ≤ capacity of (k).
- Budget: total installation cost ≤ budget.
- Power network constraints (optionally linearized) for nodal capacity and voltage limits.

Notes: choose (β, γ) to reflect policy priorities.

4. Candidate datasets & APIs (plan)

- **NREL / AFDC Alternative Fuel Stations dataset / API** — national station locations and attributes. (good baseline for location data and metadata).
- **UrbanEV (Urban Electric Vehicle) dataset** — spatio-temporal charging usage for Shenzhen (or similar city) — use for demand profiling and forecasting.
- **ST-EVCDP GitHub (Shenzhen)** — spatio-temporal EV charging datasets (real-world logs) for demand modeling.
- **Mendeley EV Charging Dataset** — parking-lot level charging sessions (timestamps, kWh delivered) — useful to model session duration and energy per session.
- **OpenChargeMap / Placekey derivatives / Kaggle snapshots** — supplemental station location datasets and global coverage.

Data plan:

A. Road Network Data (Primary): Using OSMnx;
This gives:
✓ road graph
✓ nodes & edges
✓ travel times
✓ shortest routes for detour calculations

B. Charging Station Data Sources

These will be pulled programmatically:

1. **OpenChargeMap API (OCM)**
    - Free global EV station directory
    - Has India & Uttar Pradesh locations
    - Provides charger type, status, power ratings
2. **OpenStreetMap amenities**
    - Some EV charging locations are tagged amenity=charging_station.
3. **State EV policy documents**
    - If needed, we can manually add LDA (Lucknow Development Authority) proposed stations.

C. Demand Modeling

Since no public Indian EV charging dataset exists:

We will simulate demand using:

- **Population density** (from OSM buildings or GHSL)
- **Commercial areas + malls + IT parks + universities**
- **Parking & transport hubs** (airport, railway stations, metro stations)

Demand weight $\propto$
population_density + commercial_intensity + transport_hub_factor

> *{Using AFDC/NREL or OpenChargeMap as the base map of existing infrastructure, Using UrbanEV or ST-EVCDP to model expected spatio-temporal demand patterns, If necessary, we shall synthesize trip flows using traffic data (OSRM or local origin-destination matrices) to simulate detours.}*

5. Proposed Methods / Tooling

- **Formulations:** Mixed-Integer Linear Programming (MILP) for small instances; multi-objective NSGA-II/NSGA-III or weighted single-objective for larger instances.
- **Heuristics:** Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Greedy/Flow-capturing heuristics, Hybrid GA+TSP decoders.
- **Power modeling:** AC power flow approximations or linearized DistFlow for quick evaluation of grid stress.
- **Software:** Python (Pyomo or OR-Tools), Julia (JuMP), QGIS for mapping, NetworkX for graph/flow modeling.

6. Experiments & Evaluation Metrics

- **Metrics:** Average detour per driver, percentage of demand covered, maximum grid node load / load imbalance metric, total cost, user wait time (if modeled).
- **Baselines:** Random placement, k-means clustering on demand centroids, current station network, simple greedy flow-capturing algorithm.
- **Validation:** Use held-out days from UrbanEV/ST-EVCDP for demand simulation; compare grid impact using distribution network test feeder (e.g., IEEE 33-bus adapted to region).

7. Lucknow Urban Core — Plan & Starter Code

Urban core definition

Selected neighborhoods for the urban-core prototype (Lucknow): **Gomti Nagar, Hazratganj, Aliganj, Indira Nagar, Kaiserbagh, Aminabad, Ashiyana (near SGPGI), Alambagh, Mahanagar**. The focus will be on the municipal wards covering these areas — roughly the central 25–40 km².

Data sources for urban core

- **OpenStreetMap** (via osmnx) for the road network, building footprints, and POIs.
- **OpenChargeMap / OSM amenities** for existing charging station markers.
- **Population proxy**: building footprints & land-use POIs. If higher-resolution population rasters are later needed, we will use WorldPop or GHSL.

Candidate location generation

- Use POIs (malls, markets, transit hubs) as high-priority candidate sites.
- Add a regular grid of candidate points (e.g., 500 m spacing) clipped to the urban core for exploration.

Demand modeling (urban-core approach)

- Assign demand weights to grid cells using:
    - building footprint area
    - POI counts (retail, offices)
    - proximity to transit hubs
- Normalize to form per-cell expected daily charging requests.

Initial optimization setup

- Create ~50–150 candidate locations from POIs + grid sampling.
- Limit budget to install up to 20 stations for the prototype.
- Start with a small MILP using pulp or pyomo for binary placement variables and assignment variables for a subset of demand points (~200) to keep solve-time small.

Deliverables for the prototype (urban core)

1. Processed road graph and candidate site layer (GeoJSON).
2. Demand centroid layer (GeoJSON) with weights.
3. MILP toy model (Python notebook) that outputs chosen sites and maps results.
4. A simple GA/heuristic baseline and comparison table.

Next immediate computational tasks:

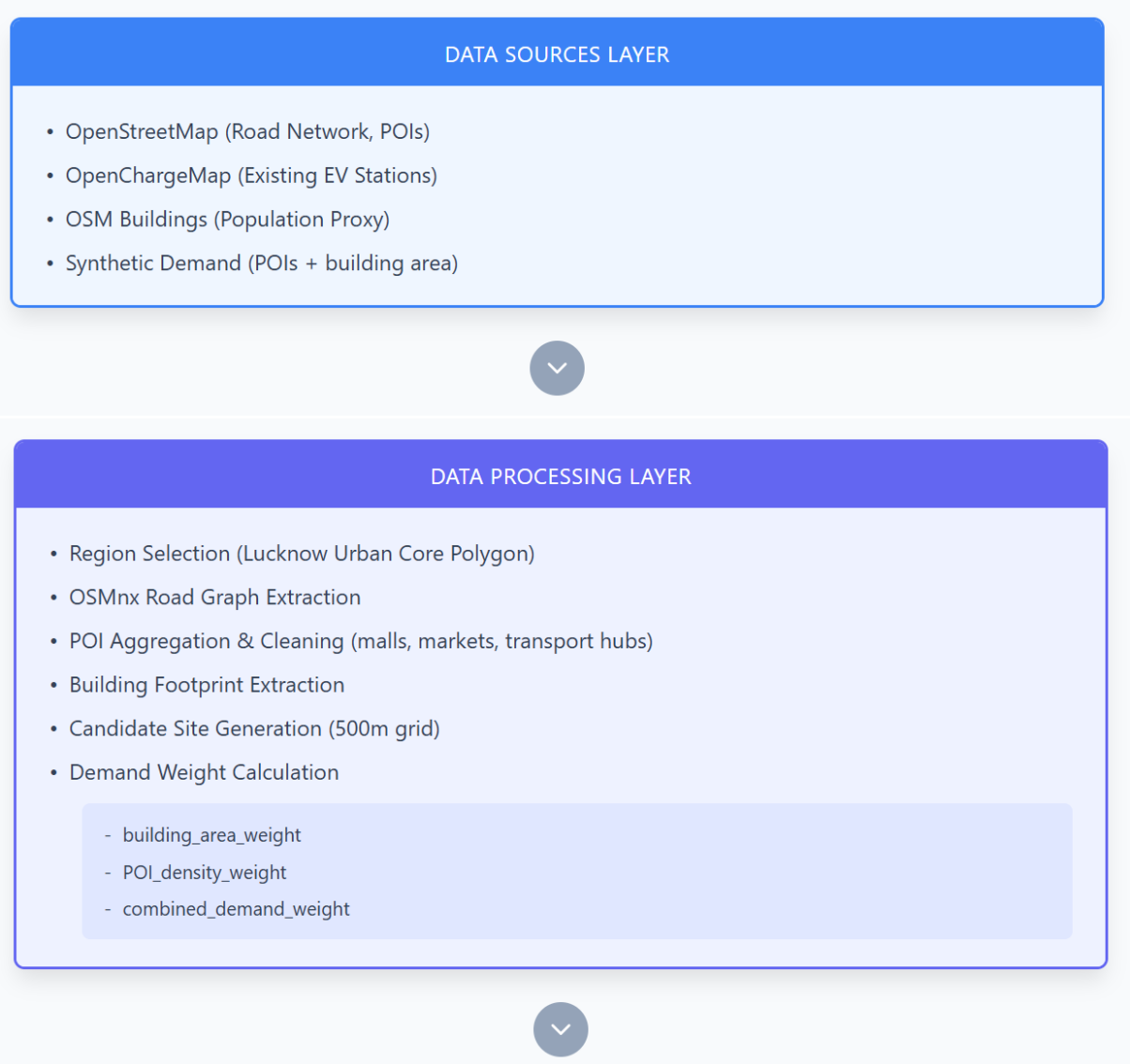
- Notebook 01_data_download_and_preprocess.ipynb: OSMnx graph extraction, POI scraping, candidate generation.
- Notebook 02_demand_modeling.ipynb: demand weights, centroid generation.
- Notebook 03_milp_prototype.ipynb: small MILP using pulp/pyomo and visualization.
- Prepared and inserted ready-to-run Python starter snippets tailored to the Lucknow urban-core (OSMnx extraction for multiple neighborhoods, candidate generation, demand weighting).
- Laid out the exact notebooks I will create and the files they will produce so you have a clear, actionable pipeline.

- lucknow_urban_core_graph.graphml (OSMnx projected graph)
- lucknow_urban_core_pois.geojson
- candidate_grid_500m.geojson
- candidate_sites_with_weights.geojson
- notebooks/01_data_download_and_preprocess.ipynb

SYSTEM ARCHITECTURE:

The architecture is divided into six logical layers.
First, the Data Sources Layer collects the road network, POIs, building footprints, and existing charging stations.
Then in the Data Processing Layer, we extract the Lucknow urban core polygon, generate candidate station sites (500m grid), and compute demand weights using POI density and building area.

The Optimization Engine Layer runs both a small MILP model for correctness and a GA/NSGA-II model for scalability.
The Analytics Layer evaluates detours, coverage radius, and grid stress.
The Visualization Layer produces interactive maps using Folium and demand heatmaps.
Finally, the Output Layer generates selected charging station locations, GeoJSON files, and results for the final report.



OPTIMIZATION ENGINE LAYER

- MILP Model (Pyomo / PuLP)
  - binary placement variables x_k
  - demand assignment variables y_ik
  - minimize detour + β*grid_stress + γ*cost
- GA/Heuristic Engine (DEAP / pymoo)
  - scalable multi-objective optimization
  - NSGA-II/GA for large candidate sets

ANALYTICS LAYER

- Detour Analysis (shortest path via OSMnx)
- Grid Stress Approximation (transformer density)
- Coverage Radius Evaluation
- Demand Served / Demand Unserved Metrics

VISUALIZATION LAYER

- Folium Interactive Maps (stations, demand)
- Heatmaps (demand distribution)
- Road Network Plots (detour visualization)
- Exported GeoJSONs (candidates, selected sites)

OUTPUT / REPORTS

- Selected Charging Station Locations
- Detour Reduction Statistics
- Grid Load Distribution Report
- Maps & GeoJSONs for presentation/testing
- Executive Summary + Technical Report

Pipeline Summary

Input Data	Processing Steps	Optimization Method
4 primary data sources	6 transformation layers	MILP + Genetic Algorithms