EcoCharge: A Framework for Sustainable Electric Vehicles Charging

Soteris Constantinou[*], Dimitris Papazachariou[*], Constantinos Costa[◇*], Andreas Konstantinidis[•*], Mohamed F. Mokbel[‡], and Demetrios Zeinalipour-Yazti[*]

[*] Department of Computer Science, University of Cyprus, 1678 Nicosia, Cyprus
[◇] Rinnoco Ltd, 3047 Limassol, Cyprus
[•] Department of Computer Science & Engineering, Frederick University, 1036 Nicosia, Cyprus
[‡] Department of Computer Science and Engineering, University of Minnesota, MN55455 Minneapolis, USA
{sconst01,dpapaz02}@ucy.ac.cy; costa.c@rinnoco.com; com.ca@frederick.ac.cy; mokbel@umn.edu; dzeina@ucy.ac.cy

***Abstract*—In this demonstration paper, we present an innovative framework for sustainable Electric Vehicles (EVs) charging, dubbed *EcoCharge*, which utilizes an intelligent energy hoarding approach. Particularly, *EcoCharge* employs a *Continuous k-Nearest Neighbor query*, where the distance function is computed using *Estimated Components (ECs)* (i.e., a query we term CkNN-EC). An EC defines a function that can have a *fuzzy value* based on some estimates. Specific *ECs* used in this work are: (i) the *(available clean) power at the charger*, which depends on the estimated weather; (ii) the *charger availability*, which depends on the estimated busy timetables that show when the charger is crowded; and (iii) the *derouting cost*, which is the time to reach the charger depending on estimated traffic. Our framework combines these multiple non-conflicting objectives into an optimization task providing user-defined ranking means through an intuitive spatial application. The algorithm utilizes lower and upper interval values derived from *ECs* to recommend the top ranked EV chargers and present them through a map interface to users. We demonstrate *EcoCharge* using a complete prototype system developed using the Leaflet - OpenStreetMap library. In our demonstration scenario, attendees will have the opportunity to observe through mobile devices the benefits of *EcoCharge* by simulating its execution over various scheduled trips with real data retrieved from API requests (i.e., *ECs*).**

***Keywords*-Mobile Data Management, Green Mobility, Renewable Self-Consumption, Electric Vehicles, Charging.**

I. Introduction

In recent years, the market penetration of electric vehicles (EVs) has exponentially expanded due to their notable advantages in sustainable transportation and cost-effectiveness compared to conventional internal combustion vehicles. Cities play a pivotal role in the pursuit of climate neutrality by 2050, a core objective of the European Green Deal[1], as they bear responsibility for over 65% of the world's energy consumption and contribute to 70% of global CO_2 emissions. Lately, a growing interest has been seen in the incorporation of Renewable Energy Sources (RES) into EV charging infrastructure, such as wind turbines and photovoltaic panels (PV) [1], [2].

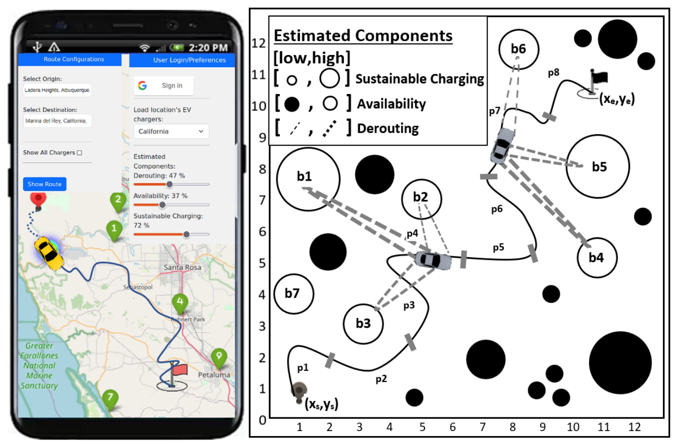

Fig. 1. EcoCharge application: An example of a moving vehicle and available chargers (b) based on a scheduled trip (P). The ranking selection is derived from each EV charger's rate and solar production curve at a certain time, considering also the estimated time of arrival (ETA), and the return trip time.

People often engage in the practice of *"energy hoarding"*, where they charge their EVs during periods of inactivity (i.e., idle time), even when the battery is not substantially depleted, to ensure that the vehicle will be charged for future travel. Even though EVs are seen as a way to reduce CO_2 emissions, thus, *energy hoarding* with non-renewable energy sources is negating environmental benefits. In the U.S., the energy demand for EV charging was estimated at 4.7 TWh in 2020, with a projected increase to ≈107 TWh by 2035[2]. Current applications focus on allowing users know where to recharge but do not list the environmental impact of the charging process (i.e., energy coming from fossil fuel burning).

A renewable hoarding technique can be applicable in scenarios with *idle time* (i.e., while an EV user is waiting or parked). For example, consider the following real-life scenarios: (i) electric taxis (e.g., Lyft, Uber, Bolt) during idle periods are waiting to be called or booked online; (ii) parents waiting

[1]EU Climate-Neutral Smart Cities, `https://tinyurl.com/57tzjmyk`

[2]Statista-EV charging demand, `https://tinyurl.com/mtc6w8nt`

in their idle EVs while their children attend after-school activities; and (iii) an EV user going for groceries or clothing shopping. Consequently, in all aforementioned scenarios, users could stop at some nearby charging station to efficiently charge their EVs using power generated from renewable sources, thus, reducing the carbon footprint of their daily routine.

One technical challenge, is that the decision of where to sustainably hoard depends on a variety of *Estimated Components (ECs)* on where and when to charge (see Figure 1). Examples of these estimations are: the *(available clean) power at the charger* that depends on the estimated weather, the *charger availability* that depends on the estimated busy timetables showing when the charger is crowded, and the *derouting cost* to reach the charger that depends on estimated traffic. To solve this problem, a *Continuous k-Nearest Neighbor (CkNN) query* [3] can be utilized to answer questions like which EV chargers are closer regarding a path. However, CkNN does not consider the estimation of various components. Our work falls under the concept of renewable hoarding techniques exploiting ECs. The objective is to optimize EV charging by utilizing only RES and focusing solely on short-term traveling, ignoring multi-stop planning and traffic scheduling (e.g., congestion balancing).

In this demo, we demonstrate an innovative renewable hoarding application for charging EVs, dubbed *EcoCharge*[3][4]. We model the problem as a new *CkNN-EC* query that retrieves the k nearest neighbors of every point on a path segment (e.g., "find all my nearest EV chargers during my route from source to end-point."), while considering ECs by employing a distance function that can express a fuzzy value. *The demo will allow the audience to experience the intelligent renewable hoarding notion, which is integrated in our EcoCharge framework, through an interactive demonstration with visual maps.*

In our previous publications, we have presented *Energy Planner (EP)* and *Green Planner (GP)*, integrated in a Home Energy Management System called *IMCF*$^+$ [5], [6]. Both, *EP* and *GP*, adapted off-the-shelf AI algorithms (hill climbing and simulated annealing), and focus on "long-term" planning, meaning that they would compute a whole year plan by doing less complex daily computations. Furthermore, we developed a system called *GreenCap* [7], which refers to "daily" planning as it attempts to find the best combination for allocating and shifting appliances during a day by minimizing the imported energy from the grid, while considering peak demand and high energy production times.

II. The EcoCharge Overview

In this section, we describe the architecture of a prototype system that we have developed followed by our algorithm.

A. System Architecture

The core of our system resides in an *EcoCharge Client* supported by a centralized server, which interacts with external APIs to retrieve essential data (see Figure 2). Leveraging external APIs, our *EcoCharge Information Server (EIS)* acquires

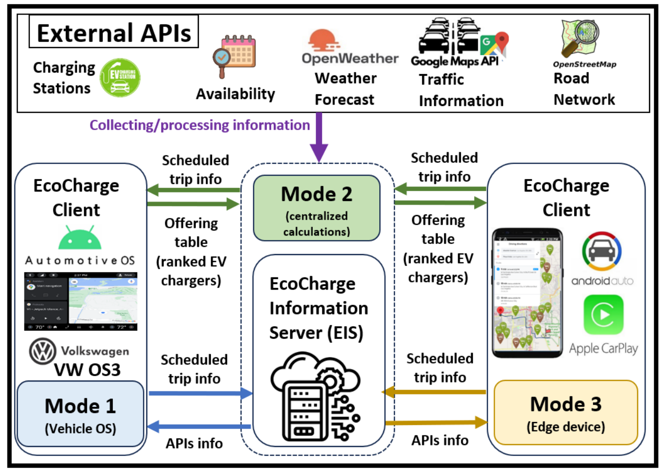

Fig. 2. **EcoCharge Architecture:** the server takes as an input all available EV chargers, weather forecast, availability, traffic data, and road network information.

real-time weather forecast data, detailed road network information, and a comprehensive list of all available EV charging stations based on the user's location. This centralized approach allows the server to efficiently consolidate the required data and distribute to individual clients as per request.

The service can be provided to the users with three modes of operation: (i) *Mode 1*, where EcoCharge operates in a vehicle's embedded operating system (e.g., Automotive OS, Volkswagen OS3); (ii) *Mode 2*, where *EIS* takes over EcoCharge calculations centrally; and (iii) *Mode 3*, where EcoCharge functionalities are managed by an edge device (e.g., smart phone using Android Auto or Apple CarPlay).

B. EcoCharge Prototype

EcoCharge Information Server (EIS): *EIS* is designed using the Laravel PHP Framework, ensuring a robust and organized structure, and it is deployed with the high-performance Nginx web server for efficient handling of HTTP requests. It efficiently retrieves road network information by integrating with OpenStreetMap[4], which facilitates advanced functionalities such as route planning, enabling users to obtain optimal directions and navigate seamlessly on a road network. PlugShare[5] is used to gather information about EV stations based on users' location. To gauge environmental conditions such as sunlight availability, we rely on data from OpenWeatherMap[6], an API service offering up-to-the-minute weather information for diverse global locations.

EcoCharge Client: Upon receiving, through an API call, the weather forecast, road network details, and EV charging station information from the *EIS*, the client application takes on the pivotal role of processing this data. Tasked with the responsibility of hoarding optimization, the client application employs a novel algorithmic approach to calculate the most efficient route considering sustainable charging and derouting

[3]EcoCharge, `https://ecocharge.cs.ucy.ac.cy/`

[4]OpenStreetMap: `https://www.openstreetmap.org/`

[5]PlugShare-EV Charging Stations, `https://www.plugshare.com/`

[6]OpenWeatherMap, `https://openweathermap.org/`

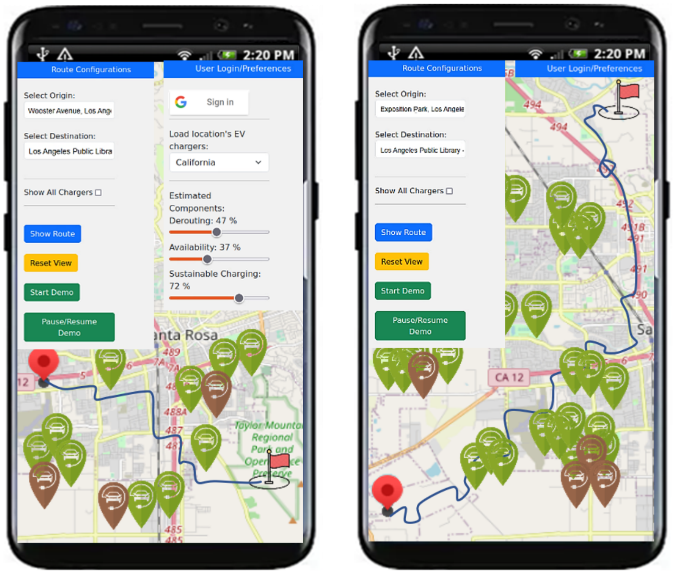

Fig. 3. **EcoCharge Client Graphical User Interface** - Prompts users to add *EC* preferences and destination so that the framework provides a ranking of the most sustainable chargers identified in each path segment of the trip.

cost based on the user's scheduled trip. This process involves dynamically identifying EV chargers along the route, considering factors such as real-time sunlight conditions, road network intricacies, and availability.

EcoCharge Client, implemented in Python 3, leverages the capabilities of the Folium library - a robust tool designed for creating diverse Leaflet maps. The utilization of Folium is integral to our system's functionality, providing a dynamic and interactive mapping component. Through the integration of Leaflet, HTML, and JavaScript, we ensure that our system not only delivers powerful functionality but also presents information in a visually engaging and accessible manner. Our mobile-based application enhances user experience by integrating with the device's location services. Through an intuitive Graphical User Interface (GUI), users can easily set their desired destination for a trip and receive comprehensive route information, leveraging the application's functionality for efficient navigation (see Figure 3).

C. The EcoCharge Algorithm

This work aims to develop an intelligent technique facilitating sustainable EV charging through an energy hoarding algorithm. By identifying charging stations that offer the most green (i.e., clean) energy, users take advantage of RES self-consumption, while reducing CO_2 emissions.

The intention of the EcoCharge is to optimize an objective function to achieve a trade-off between the vehicle's sustainable charging level L, the chargers' availability A, and the derouting travel distance D to a charger.

In order to continuously monitor the result of kNN, the *CkNN-EC* method necessitates partitioning the entire route distance into separate segments, which are sequentially considered for the kNNs determination of the query object. The partitioning procedure is responsible for dividing the scheduled trip into segments (e.g., ≈5km each segment; can be modified in settings as per preference). It is essential to note that the road network distances between all chargers and the query object (i.e., EV vehicle) have to be updated every time the query object reaches a segment intersection of the scheduled trip. For each segment, the process of finding the kNNs is composed of two phases. The first one is called *Filtering phase*, which is used to discard non-qualifying chargers. The second phase is called *Refinement*, where an evaluation is conducted to determine the eligibility of candidate chargers as *CkNN-EC*.

According to the driver's location, the *Filtering phase* ensures that only the k most suitable chargers are considered, while pruning all the rest. The particular phase loops through the entire pool of EV chargers and examines each one based on the following *Estimated Components (ECs)*:

Sustainable Charging Level (L): Each EV charging station b has a different charging rate and power generation levels s_t depending on time and location. Further, the weather forecast (e.g., sunny, cloudy) is retrieved by a cloud service (e.g., OpenWeather, Windy, WindFinder), which utilizes weather models like Global Forecast System (GFS) and European Centre for Medium-Range Weather Forecasts (ECMWF), both with an accuracy of 95–96% for up to 12 hours and 85–95% for three days. L consists of lower and upper estimation values, thus, the final result is an interval L_{min} to L_{max}.

$$L(B) = max\{s_t^b \mid \forall\ b \in B\ \} \tag{1}$$

Availability (A): Each EV charger's *availability* is estimated using a third-party service (e.g., Google Maps POI busy timetables), enabling the determination of real-time accessibility on a given time t. Therefore, an interval is produced A_{min} to A_{max}.

$$A(B) = max\{A_b \mid \forall\ b \in B\ \} \tag{2}$$

Derouting Cost (D): A route path from starting point v_0 to target charger v_k is a sequence of nodes $P =< v_0, v_1, ..., v_k >$, where every edge represents a weight w in terms of CO_2 emissions and is calculated based on the additional energy an EV needs to travel. The *derouting* considers real-time traffic information (e.g., congestion) at a given time and location retrieved from a cloud Geographic Information System service (e.g., Google Maps, Waze, HERE Maps), thus, D consists of lower and upper estimation values. Therefore, the final result is an interval D_{min} to D_{max}. The minimum derouting cost for a segment p of the path P to all chargers B is:

$$D(B) = min\{\sum_{i=0}^{|p|} w_{v_i,b} * distance(v_i, b) \mid v_i \in\ p,\ \forall\ b \in B\} \tag{3}$$

Each charging station selected in the pool of filtered candidates undergoes through the *Refinement Phase* to evaluate its *Sustainability Score (SC)*. In this work, we evaluate *SC* as a weighted sum function, where w_1 is the weight of *Sustainable*

Charging Level (L) objective, w_2 is the weight of *Availability* (A) objective, and w_3 is the weight of the *Derouting Cost* (D) objective, respectively. Using *CkNN* with *SC* as the distance function, *EcoCharge* produces two result-sets, one based on SC_{min} and another on SC_{max}, until the final output of their intersection consists of k chargers.

$$SC_{min} = (L_{min} * w_1) + (A_{min} * w_2) + ((1 - D_{min}) * w_3) \quad (4)$$

$$SC_{max} = (L_{max} * w_1) + (A_{max} * w_2) + ((1 - D_{max}) * w_3) \quad (5)$$

$$SC(B) = sort(SC_{max}(b) \cap SC_{min}(b)), \forall b \in B \quad (6)$$

III. Demonstration Scenario

During the demonstration, the conference attendees will get the chance to appreciate the key elements of *EcoCharge*, its adaptability as well as our proposition performance.

A. Case Scenario

An instance of our real prototype system has been deployed in Nicosia, Cyprus, surrounded by approximately 50 EV charging stations. Through the *EcoCharge* GUI, a test user configured the k parameter to 3 (i.e., retrieve 3 nearest/most sustainable chargers). The user wanted to do some shopping, thus, a scheduled trip was set up starting at 18:00 o'clock from the city center and ending up at the city's mall (i.e., ≈22-25 minutes travel time). During the trip, the user received multiple recommendations for EV chargers generated by the proposed framework, occurring every ≈5 minutes. However, the user preferred to utilize the idle time of their EV at the final destination for charging while shopping. The *EcoCharge*'s performance was measured with respect to the *Sustainability Score (SC)* and time. Considering the *ECs* retrieved from *EIS* mentioned in Section II-C (e.g., availability at current time, traffic, weather conditions), the application produced a set of 3 EV chargers nearby the mall, in a reasonable response time $t \approx 0.5$ seconds, with the following scores: (i) $SC_{ChargerA}$ = 98% and 14 meters from destination; (ii) $SC_{ChargerB}$ = 80% and 57 meters from destination; (iii) $SC_{ChargerC}$ = 72% and 3.2km from destination. Consequently, the user saved time by having multiple available options for parking spots where they could sustainably charge their vehicle.

B. Demo Plan

The conference attendees will have the opportunity to interactively engage with the *EcoCharge* application interface by configuring preferences and setting up a scheduled trip using a tablet or smartphone. A number of synthetic preference configurations will be pre-loaded to several demo user accounts through the application's back-end. A demonstration will take place over real road network maps to graphically expose the applicability of the *EcoCharge* algorithm in real-time.

The main objective of our approach is to enable EV users to sustainably charge their vehicles through a renewable hoarding process, by leveraging renewable energy sources, optimizing charging strategies, and reducing operational costs. To showcase the advantages of our technique to the audience, we will provide visual representations to help them gain a clear insight into the performance benefits. These visuals will illustrate the increasing levels of sustainability charging and the corresponding improvements in self-consumption observed during our experiments.

As part of the demonstration, we will hand out to attendees three mobile devices that will act like real EV users, each one with a different scheduled trip in the road network of California. Participants will have the opportunity to view on their smartphone displays the status of the simulated moving EV vehicles and the recommended chargers, which will appear every few seconds (i.e., in every segment of the scheduled trip) sorted based on *SC*. The three cases will vary since different locations and times will be adjusted on the scheduled trips, such as weather conditions, chargers' availability, and traffic congestion. The *EC* values are going to be based on real life data retrieved by *EIS*. The aforementioned case scenarios will be based on 25, 35, and 45 minutes scheduled trips, respectively. Therefore, the execution time will be fast-forwarded so that the audience can observe the updates of charger recommendations throughout the trip.

Furthermore, participants will have the opportunity to form custom scheduled trips through the mobile application. Our hypothesis is that data engineering practitioners and researchers would like to compose their own user preference profiles, as opposed to be restricted within the boundaries of the well-defined provided templates. Particularly, we will provide attendees with the possibility to set up scheduled trips based on various times throughout the day to notice the impact during peak-demand periods. The goal will be to clearly describe the *EcoCharge* and the intelligent renewable hoarding approach employed, which makes our implementation an environmentally friendly alternative to traditional charging methods.

References

[1] Y. Li, J. Luo, C. Y. Chow, K. L. Chan, Y. Ding, and F. Zhang, "Growing the charging station network for electric vehicles with trajectory data analytics," in *2015 IEEE 31st ICDE*, 2015, pp. 1376–1387.

[2] S. N. Basharzad, F. M. Choudhury, E. Tanin, L. L. H. Andrew, H. Samet, and M. Sarvi, "Electric vehicle charging: It is not as simple as charging a smartphone (vision paper)," in *Proceedings of the 30th ACM International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '22, New York, NY, USA, 2022.

[3] Y. Tao, D. Papadias, and Q. Shen, "Chapter 26 - continuous nearest neighbor search," in *VLDB '02: Proceedings of the 28th International Conference on Very Large Databases*, P. A. Bernstein, Y. E. Ioannidis, R. Ramakrishnan, and D. Papadias, Eds. San Francisco: Morgan Kaufmann, 2002, pp. 287–298.

[4] S. Constantinou, C. Costa, A. Konstantinidis, M. Mokbel, and D. Zeinalipour-Yazti, "A framework for continuous knn ranking of ev chargers with estimated components," in *40th IEEE ICDE*, May 13 - 16, 2024, Utrecht, Netherlands, 2024, conference, p. 13.

[5] S. Constantinou, A. Konstantinidis, P. K. Chrysanthis, and D. Zeinalipour-Yazti, "Green planning of iot home automation workflows in smart buildings," *ACM Trans. Internet Things*, vol. 3, pp. 1–30, jun 2022.

[6] S. Constantinou, A. Konstantinidis, D. Zeinalipour-Yazti, and P. K. Chrysanthis, "The iot meta-control firewall," in *37th IEEE ICDE*, April 19 - April 22, 2021, Chania, Crete, 2021, conference, p. 12.

[7] S. Constantinou, N. Polycarpou, C. Costa, A. Konstantinidis, P. K. Chrysanthis, and D. Zeinalipour-Yazti, "An iot data system for solar self-consumption," in *24th IEEE MDM*, July 3- July 6, 2023, Singapore, 2023, conference, p. 10.