GreenCap: A Platform for Solar Self-Consumption using IoT Data

Soteris Constantinou*, Nicolas Polycarpou*, Constantinos Costa◇*‡, Andreas Konstantinidis•*,
Panos K. Chrysanthis‡* 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, University of Pittsburgh, PA 15260, USA
{sconst01, npolyc03}@ucy.ac.cy; costa.c@rinnoco.com; panos@cs.pitt.edu; {akonstan, dzeina}@ucy.ac.cy

***Abstract*—In this demonstration paper, we present an innovative IoT data platform, coined *GreenCap*, which utilizes a Green Planning evolutionary algorithm for load shifting of IoT-enabled devices in smart environments that feature renewable energy sources. Particularly, GreenCap deploys a hybrid genetic algorithm with domain-specific local search heuristics, which results in a Memetic Algorithm (MA) that offers users an energy efficient allocation plan of their IoT devices, based on their personal preference rules (e.g., operate AC from 10am - 1pm). Our system allocates operations in the daily time-slots considering devices' energy bounds to minimize the imported energy from the grid, exploit self-consumption and maximize users' comfort. We demonstrate GreenCap using a complete prototype system available on Raspberry Pi, developed in Laravel using MariaDB and linked to openHAB framework. In our demonstration scenario, attendees will be able to observe through mobile devices the benefits of GreenCap by simulating its execution with real data for one week, using pre-configured or custom rules.**

I. Introduction

Home Energy Management Systems (HEMS) enable households to function as the end-use node by providing the ability to adjust their energy demands, thus advancing the utilization of Renewable Energy Sources (RES) and assisting in the mitigation process of climate change. Considering distributed and weather-dependent RES, the time of day people consume energy becomes significantly important in reducing CO_2 emissions (see Figure 1). Residential loads account for a large amount of the utility's load demand, and this number drastically grows along with various involved applications [1]. The number of IoT devices is projected to amount ≈40 billion units by 2025[1], and eventually to ≈100 billion by 2030 [2].

Green Planning denote smart computational techniques aspire to improve environmental quality towards sustainability through load shifting considering peak demand reduction [3]. A key driver for controlling the energy usage and CO_2 emissions is the uptake of *Internet of Things (IoT)* infrastructure, which connects every single intelligent gadget in the world able to perform various operations, as well as communicate using open protocols [4]. Further, the self-consumption of renewable sources continues to be complementary to the present and future need for a cleaner biosphere, as it could be much more beneficial than energy storage batteries where 17% of the energy is lost in AC/DC conversion losses and heat dissipation. Consequently, minimizing the CO_2 pollution in areas where humans are active and spend 80-90% of their time, can impact the environment in a positive way [2]. According to the European Commission Green Deal[2], it was decided to reduce net greenhouse gas emissions by at least 55% by 2030.

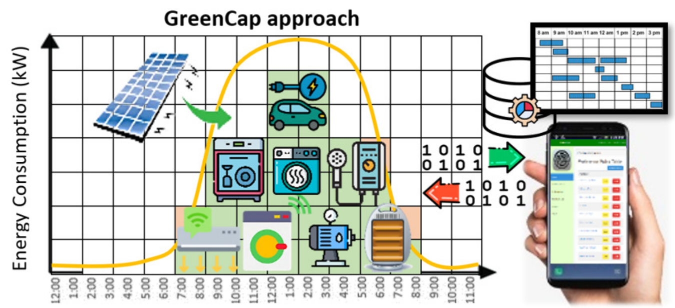

Fig. 1. A GreenCap daily load allocation example considering user preference rules and solar energy production times.

In our prior published works, we introduced the *Energy Planner (EP)* and *Green Planner (GP)*, integrated in a system called $IMCF^+$ [5], [6]. Both approaches adapted off-the-shelf AI algorithms, and focus on "long-term" planning, meaning that they would compute a whole year plan by doing less complex daily computations. For example, $IMCF^+$ generates a residential plan while considering the family's configured annual energy budget and preference rules. On the other hand, the platform proposed in this demo, coined *GreenCap*[3], 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 high demand and energy production times. The challenge is how to fill the daily time-slots with device operations, as retrieved from *Query 1* result-set, considering their maximum energy

[1]Statista., URL: https://tinyurl.com/mw74ku2h

[2]European Green Deal, URL: https://tinyurl.com/3hbypfum
[3]GreenCap, URL: https://greencap.cs.ucy.ac.cy/

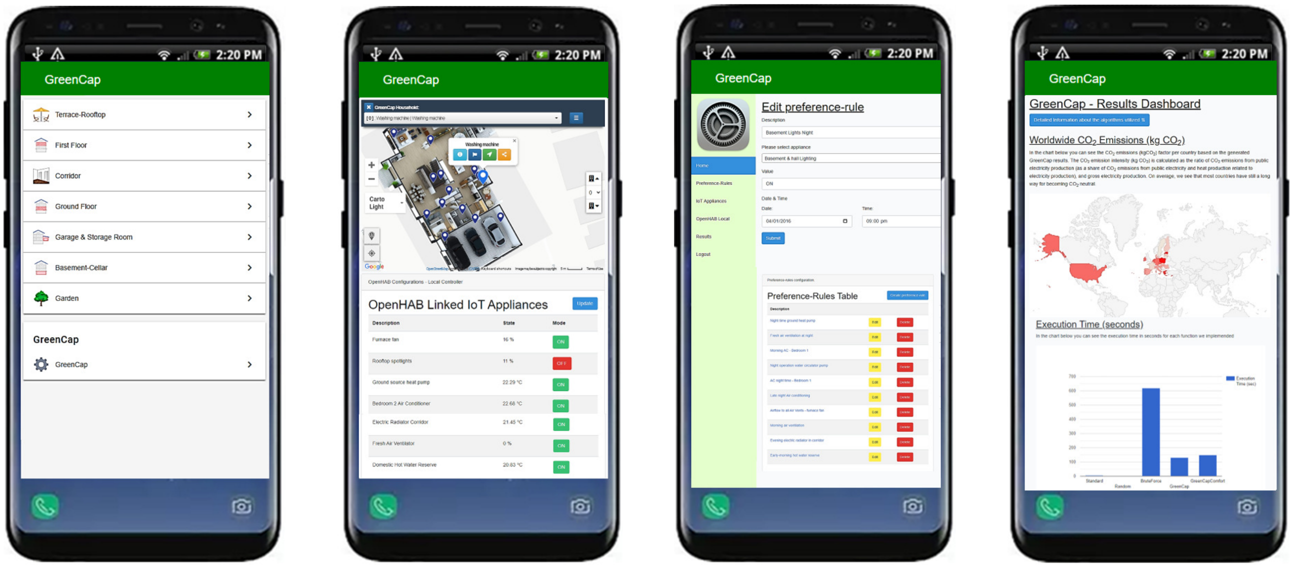

Fig. 2. GreenCap GUI: Dashboards of Preference Rules and Linked IoT Appliances - Interfaces displaying performance analysis results and comfort levels.

bounds (e.g., washing machine: 2 hours ≈1kW). The particular problem is an adaptation of the Bin Packing problem that is NP-hard (i.e., the 2D packing).

Query 1 - Device operations retrieved from database

```
SELECT device_id,start_time,end_time,power
FROM JOBS
WITH GreenCap
EPOCH DURATION 1 day
```

Further, the planning optimization of devices in smart environments is a very difficult task due to the unsophisticated user-defined preference rules. Most existing solutions confront convergence difficulties as they cannot efficiently manage a large number of IoT devices neither complex multi-objective problems. Particularly, the goal of this work, is to compute in real-time a sustainable operating schedule that satisfies the daily operation intervals of various devices in the solar production curve, while considering peak-demand times, Residential Consumption Record (RCR) history, and user comfort levels.

In this demo, we present a novel Green Planning platform, coined *GreenCap* [7], acting as an energy planning framework. Due to the high complexity of the problem's decision space, we utilize an evolutionary algorithm to generate a high-quality solution to a particular search problem by relying on bio-inspired operators. The hybridization of a genetic algorithm with domain-specific local search heuristics, which results in a memetic algorithm (MA), can further improve users' fitness and provide high convergence by reducing the likelihood of trapping in local optima. Due to the lack of labeled training data, our approach does not utilize machine learning techniques. *The demo will allow the audience to experience the intelligent self-consumption of RES notion, which is integrated in our GreenCap platform, through an interactive domain-specific demonstration with visual cues.*

II. The GreenCap Overview

In this section, we describe a prototype system we have developed for GreenCap using Laravel MVC framework along with MariaDB as a relational database, linked to open Home Automation Bus (openHAB)[4], and Anyplace for venue construction [8]. We begin with a discussion of the system architecture, followed by the GreenCap algorithm along with its local search heuristics. The Graphical User Interface (GUI) directly integrates into OpenHAB's web and mobile panels for interactive management of IoT devices and automated management of sustainability-aware preference rules.

A. System Architecture

The system architecture is composed of the elements listed below: (i) a custom main *control unit* that can be linked with either openHAB or Domoticz, acting as an intelligent residential management application; (ii) *GreenCap Controller*, a platform that contains the entire energy management logic; and (iii) the web *Graphical User Interface*.

Control Unit (CU): is a system implemented in JAVA installed on a device, such as a Raspberry Pi, operating in a user's localized network. To manage IoT appliances with respect to the configured by the users Preference Rules (PR), the CU will be communicating directly with them. Normally, after the phone application is downloaded by the users, they will be able to interactively control their IoT appliances through CU. For the design of the CU, one can extend Domoticz or openHAB framework, both falling under the open-source software category, for automated smart residences offering a vast ecosystem of bridges that allow users to directly communicate remotely or locally with IoT appliances (e.g., LG dryer, Samsung Smart A/C).

[4]openHAB, https://openhab.org

To understand the functionality of CU consider, for example, a resident trying to configure the settings of his heating boiler through a smart application. The manual regulation is now undertaken by the CU that eventually interacts with the IoT appliances. In cases where the user's application is away from the smart environment, the Network Address Translation (NAT) and network firewall will prevent interaction with CU. In such scenario, the application connects to a public server through the Cloud Controller (CC) to communicate and manage CU remotely.

GreenCap Controller: is an extension application to CU we have designed to encapsulate the development of the MA along with the GUI and required storage to enable users adapt their preferences and meet an energy-aware planning solution. The settings configured by the users in a local relational MariaDB database are passed as parameters in the GreenCap algorithm, which has been developed as a JAVA module. The user(s) populate the storage layer through the mobile application, which has been regulated to smoothly incorporate the definition of PR via a web GUI portal (see Figure 2). A cron job daemon is utilized for the GreenCap module, which is available on Linux and invokes reliably the load scheduling in fixed intervals of time (e.g., every several minutes). In case appliances need to be switched on/off, according to the allocation plan, the platform adapts either the *Binding-mode*, which utilizes the diverse ecosystem of bridges available on the openHAB framework (open source) for interaction, or the *Extended mode*, which performs custom commands in the localized network. To prevent any suspicious communication activities, we configured the CU network firewall with the `iptables` command in order to discard unauthorized TCP interactions to designated appliances on the localized network.

Graphical User Interface (GUI): is constructed using Laravel MVC framework, HTML, and JavaScript. The web portal relies on a web-server named NGINX, which is available on Raspberry Pi. The GUI consists of the PR interface and the GreenCap planning results obtained by the inspired sustainability-aware algorithm. The PR site prompts users to configure their IoT preference settings for any date-time-slots (see Figure 2). Information in regards to the state of openHAB's pre-configured IoT appliances in a building, is obtained through a Rest API service consisted of sensor measurements, which are displayed on a particular web-view.

B. *The* GreenCap *algorithm*

The GreenCap algorithm consists of an innovative Memetic Algorithm (MA) we have developed along with several local search heuristics to efficiently manage a large number of devices while considering simultaneously the optimization of energy consumption, user costs, comfort and CO_2 emissions. The MA is based on a generic genetic algorithm hybridized with two search heuristics to further improve user's fitness and reduce the likelihood of premature convergence. The chromosomes are adjusted following a residential energy consumption pattern showing the status (ON/OFF) of the smart devices, each time-slot's consumption, and the length of the chromosomes indicating the total number of devices. For every possible solution, the fitness function is calculated and compared based on the imported energy (IE) from the grid and user comfort (UC) levels, while also considering high production times and peak-demand periods. IE is the energy retrieved from the grid at a particular time-slot t so that appliances D can complete the required operations set by residents. It is calculated as the difference of the energy consumption c_i and the produced energy p, given by:

$$IE_t = min \sum_{i=1}^{D} (c_i^t - p^t)/t = 1, ...24 \tag{1}$$

UC is the sum of all executed rules defined by the user. The total set of preference rules PR is of size N. Each rule PR_i=1 when successfully adapted and consequently executed, otherwise PR_i=0, as shown in the equation below:

$$UC = max \sum_{i=1}^{N} (PR_i) \begin{cases} 1, & \text{if } PR_i \text{ is executed} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

GreenCap converts PR to a binary vector, where each vector's position represents a preference rule in PR. Then, it calculates the approximate daily consumption of each appliance based on the Residential Consumption Record RCR history. This calculation supports GreenCap's $ComfortOptimization$ heuristic to retain and balance the energy consumption, due to fluctuations that may occur, by avoiding turning on/off too many devices that could also affect the users' experience. In case PR settings are configured in a way that there is a conflict with the historical record RCR, then the system prioritizes users' comfort by adapting the corresponding PR. A population function randomly generates a solution $s =< s_i, ..., s_N >$, where s_i = 1 means that preference rule at position i will be triggered at a specific time period, and s_i = 0 means that it will be discarded respectively. The solution then is evaluated using the fitness function in respect to IE and UC. Further, a new solution $s*$ is generated by $EnergyOptimization$ heuristic, liable to avoid the allocation of devices during peak demand times by swapping operations to non-peak hours, while also considering high production periods. Both solutions are compared using the evaluation metrics and only the best is forwarded to the next generation. The entire procedure stops when the full cycle of generations is completed.

III. Demonstration Scenario

During the demonstration, the conference attendees will get the chance to appreciate the key elements of GreenCap, the adaptability as well as our propositions performance.

A. *Case Scenario*

An instance of our real prototype system has been deployed in a house with four residents for one week. The residents were able to set-up their preference rules according to their consumption patterns and habits through the mobile platform, which directly interacts with a GreenCap-CU node on a Linux

TABLE I
EVALUATING OUR SYSTEM PROTOTYPE

	GreenCap Algorithm
Time Duration	1 Week
Imported Energy from the grid	$\approx$ 71 kWh
Self-consumed Energy	$\approx$ 61 kWh
Carbon Dioxide Emissions	$\approx$ 32 kg CO_2
User Comfort Level	$\approx$ 99 %

virtual machine on our datacenter. Several PR ($\approx$15 rules) were configured in the system by the residents. One of the rules was to keep turned ON the AC unit during 11:00-14:00 (e.g., high energy generation hours) every day of the week. Another rule was to have the dryer working during off-peak hours (e.g., 10:00-12:00) the first two days of the week. The configuration settings were approximately 180 bytes, stored in the MariaDB persistency layer. The OpenWeather forecast API service was utilized to get various environmental data, such as the high production periods. The performance of the proposed GreenCap system is measured with respect to user comfort levels (UC), imported energy from the grid (IE), self-consumption (SE), and CO_2 emissions $(Z_{(IE,k)})$, where k is the CO_2 intensity of a particular country (kg CO_2 per kWh). Table I summarizes our evaluation results. Our observation is that GreenCap is a sustainable approach as it executes in $\approx$ 5 seconds with high satisfaction $UC \approx 99\%$, great performance with $IE \approx 71$ kWh, $SE \approx 61$ kWh, and $\approx$32 kg CO_2 emissions.

B. Demo Plan

The conference attendees will have the opportunity to interactively engage with the PR web interface and openHAB by configuring preferences using a tablet or smartphone. Numerous synthetic and web-accessible preferences will be pre-loaded to a demo user profile through the back-end of the website. The loaded preferences will capture the demands and structure of realistic household data and will be useful to demonstrate graphically the operation of GreenCap algorithm in real-time through the openHAB framework.

Our technique's main objective is to enable users to find a sustainable allocation plan for the operation of a set of smart appliances, a pool of preference rules and a tentative peak-demand history, while reducing at the same time CO_2 emissions. To demonstrate the benefits of our propositions to the attendees, we will provide visual representations that will allow the audience to get a clear understanding of the performance advantages, showing the reduction of imported energy from the grid, and the increase of self-consumption and comfort levels that we have witnessed during the experiments.

As part of the demonstration, we will handover to participants 4 smartphones that will act like residential appliances, such as an air conditioner, an electric vehicle charger, a ground source heat pump, and an electric radiator. Attendees will be able to observe through the smartphone screens the state and mode of each device. Particularly, the mode indicates whether each device is ON or OFF, and the state shows their actual values (e.g., air conditioner temperature, EV charging power, etc.). The case scenario will be based on a week's duration, thus, the execution time will be fast-forwarded and the audience will notice the actuations on residential appliances.

Moreover, participants will have the chance to form custom preference rule profiles through the web interface. Our hypothesis is that several data engineering practitioners and researchers would like to compose preference profile predicates, as opposed to be restricted within the boundaries of the well-defined provided PR templates. Attendees will also have the opportunity to upload, through Anyplace [8], a real residential venue. The openHAB framework will enable participants to visualize the generated results on a smart device, through fancy graphs. The main goal will be to clearly describe the GreenCap implementation, residing on openHAB, and the load scheduling of operations considering the configured rules.

IV. Future Work

We intend to carry out additional research in the future with respect to different challenges that need to be addressed, such as interoperability, power fluctuations, and interdisciplinarity. Load scheduling considering conflicting interests is something we will investigate further along with cloud extensions that will allow GreenCap to function as a *cloud-meta-controller*. Finally, we plan adjusting our algorithm in the green-mobility sector in order to reduce the environmental impact of mobility in terms of greenhouse gas emissions. Particularly, the adaptation of our algorithm has the potential to significantly reduce the overall carbon footprint of the transportation sector by optimizing charging strategies for electric vehicles.

References

[1] N. G. Paterakis, O. Erdinc, I. N. Pappi, A. G. Bakirtzis, and J. P. S. Catalao, "Coordinated operation of a neighborhood of smart households comprising electric vehicles, energy storage and distributed generation," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2736–2747, 2016.

[2] S. Constantinou, A. Konstantinidis, and D. Zeinalipour-Yazti, "Green planning systems for self-consumption of renewable energy," *IEEE Internet Computing*, pp. 1–1, 2022.

[3] Z. Wen, S. Garg, G. S. Aujla, K. Alwasel, D. Puthal, S. Dustdar, A. Y. Zomaya, and R. Ranjan, "Running industrial workflow applications in a software-defined multicloud environment using green energy aware scheduling algorithm," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5645–5656, 2021.

[4] M. Poess, R. Nambiar, K. Kulkarni, C. Narasimhadevara, T. Rabl, and H.-A. Jacobsen, "Analysis of tpcx-iot: The first industry standard benchmark for iot gateway systems," in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, 2018, pp. 1519–1530.

[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*, jun 2022.

[6] S. Constantinou, A. Konstantinidis, D. Zeinalipour-Yazti, and P. K. Chrysanthis, "The iot meta-control firewall," in *37th IEEE International Conference on Data Engineering*, ser. ICDE'21, April 19 - April 22, 2021, Chania, Crete, 2021, conference, p. 12 pages.

[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 International Conference on Mobile Data Management (MDM)*, July 3- July 6, 2023, Singapore, 2023, conference, [Accepted].

[8] M. Angelidou, C. Costa, A. Nikitin, and D. Zeinalipour-Yazti, "Fms: Managing crowdsourced indoor signals with the fingerprint management studio," in *19th IEEE International Conference on Mobile Data Management*, 2018, pp. 288–289.