2023 24th IEEE International Conference on Mobile Data Management (MDM) | 979-8-3503-4101-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/MDM58254.2023.00022

An IoT Data System for Solar Self-Consumption

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*—Energy efficiency has become a primary optimization objective due to the global energy crisis and high levels of CO_2 emissions. Climate and energy targets have been leading to a growing utilization of solar photovoltaic power generation in residential buildings. As the number of IoT devices drastically increases, their automation through an intelligent home energy management system can provide energy and peak demand savings. The planning optimization of devices can be very challenging due to the unsophisticated user-defined preference rules. Existing solutions face convergence difficulties due to the management of multiple IoT devices tackling multi-objective problems. In this paper, we propose an innovative IoT data system, coined GreenCap, which utilizes a Green Planning evolutionary algorithm for load shifting of IoT-enabled devices, considering the integration of renewable energy sources, multiple constraints, peak-demand times, and dynamic pricing. We have implemented a complete prototype system available on Raspberry Pi and linked with openHAB framework. Our experimental evaluation with extensive real traces shows that the GreenCap prototype system efficiently generates a sustainable plan obtaining high levels of user comfort 92-99% along with ≈52% of self-consumption, while reducing ≈35% of the imported energy from the grid and ≈40% of CO_2 emissions.**

***Keywords*-Green Planning, Rule Automation, Renewable Self-Consumption, Internet-of-Things, Load Shifting.**

I. Introduction

Home Energy Management Systems (HEMS) make a residence act as the end-use node by allowing flexible energy demand, thus advancing the utilization of Renewable Energy Sources (RES) and assisting in the mitigation process of climate change [1]. 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 [2]. The global HEMS market has increased from USD 864.2 million in 2015 to USD 3.15 billion by 2022 [3].

Green Planning refers to computational approaches that aim to make rapid progress towards sustainability through load shifting considering peak demand reduction [4]. A key driver for controlling the energy usage and CO_2 emissions is the uptake of *Internet of Things (IoT)* infrastructure, which

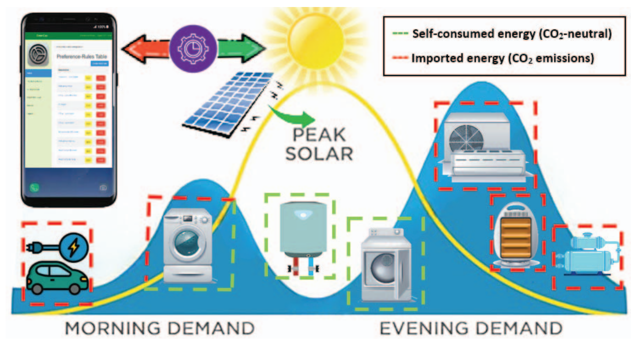

Fig. 1. An energy demand example during a day along with user preference rules and solar energy production. The red dashed lines around the appliances represent the energy used from the grid, and the green dashed lines illustrate the self-consumption of the generated energy from renewable sources.

connects every single intelligent gadget in the world able to perform various operations, as well as communicate using open protocols [5]. Further, the self-consumption of renewable sources remains complementary to present and future requirement for a cleaner environment, 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 [6], [7]. 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 [8], [9].

Through our previous publications, we have presented *Energy Planner (EP)* and *Green Planner (GP)*, integrated in a system called $IMCF^+$ [10], [11]. 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. For example, $IMCF^+$ generates a residential plan while considering the family's configured annual energy budget (e.g., 11500 kWh) and Rule Automation Workflow (RAW) pipelines. The high-level system's objective is to identify the rules that must be dropped so that users stay within the desired annual energy budget. On the other hand, the system proposed in this study, coined *GreenCap*[1], refers to "daily" planning as it attempts to find the best combination for

[1]GreenCap, URL: https://greencap.cs.ucy.ac.cy/

DOI 10.1109/MDM58254.2023.00022

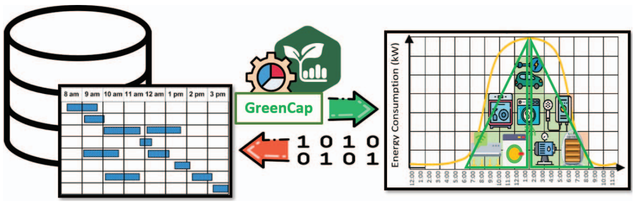

Fig. 2. GreenCap gets as an input the devices' operations and returns as an output a green scheduling plan.

allocating and shifting appliances during a day by minimizing the imported energy from the grid, while considering high demand and energy production times. This could have been perceived as a much simpler case considering our prior work, however, the given problem is NP-complete, calling for an intelligent algorithm that can approach the optimal solution in a computable time.

To illustrate the complexity of the *GreenCap* problem, let us now for ease of exposition provide a realistic example. Consider 10 hours of solar radiation on a given day x 60 minutes = 600 time slots on the x-axis. A residential solar system is 10kWp at most in most places, thus, let us assume this peak production during around noon time, as indicated in the graph of Figure 2. We can approximate the curve by 2 triangles of size: height = 10kWp / 1kW = 10 and base = 5x60 = 300 minutes, meaning that we have a rectangle (height x base) of 3000 cells to plan each day (see Figure 2). The challenge is how to fill these cells with device operations, as retrieved from *Query 1* result-set, considering their maximum energy bounds, e.g., a washing machine 2 hours ($\approx$1kW), a water heater ($\approx$3kW), etc. $GreenCap$ acts as an energy planning framework and generates a sustainable plan for an output, while incorporating the input data from *Query 1*. The particular problem is an adaptation of the Bin Packing problem [12] that is NP-hard (i.e., the 2D packing). This means that there is no polynomial time algorithm for providing a quick solution. The brute force solution (doing back-tracking) would compute the optimal solution, yet requires a lot of time, but more importantly be infeasible on low-end computing nodes (such as Raspberry Pi - 1.5GHz CPU). The complexity of the problem enforces us to use some sort of randomized algorithm to yield a good approximation.

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 [13]. Particularly, the goal of this study, is to compute in real-time a sustainable operating schedule that satisfies the daily operation intervals of the listed devices in the solar production curve, while considering peak-demand times, Residential Consumption Record (RCR) history, and user comfort levels.

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 such as mutation, crossover and selection. Additionally, 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. The proposed Green Planning MA algorithm has been integrated into our *GreenCap* system, linked with openHAB framework and evaluated in a variety of datasets with approximately 527.000 readings (408 MB in total). The extensive experimental evaluation shows that the intelligent self-consumption of renewable energy idea, integrated in our prototype system, generates an energy-efficient plan, which achieves up to 52% self-consumption and $\approx$96% user comfort, while reducing $\approx$35% of the imported energy from the grid and $\approx$40% of CO_2 emissions. In summary, in this paper we make the following contributions:

- We present *GreenCap*, a comprehensive IoT data system acting as an energy planning framework, designed and incorporated in openHAB.
- We propose a memetic algorithm to manage user comfort preferences and reduce the imported energy from the grid by considering costs, and CO_2 emissions.
- We evaluate our system through an extensive experimental series on real datasets with measurements from a residential house that comprises of a variety of IoT devices, peak electricity demand and solar panel data, showing that GreenCap can be suitable for sustainability-aware smart actuations in the future.

The remainder of the article is organized as follows: Section II presents the system model. Section III describes the proposed algorithms, and Section IV outlines the proposed system. Our experimental methodology and findings are presented in Section V, the related work in Section VI, and the article is concluded in Section VII.

II. System Model and Problem Formulation

In this section, we formalize our system model, problem formulation, and the basic terminology used throughout this manuscript. The main symbols and their respective definitions are summarized in Table I.

A. System Model

We consider a residential household incorporated with a net-metering photovoltaic (PV) system. The residence contains numerous shiftable smart appliances D such as electric heater, washing machine, air conditioner, lights, heat pump, etc. We exclude specific appliances, like the refrigerator, since it is of a high importance and should always be turned on. Therefore, the residents can consume the energy generated in the household EPT, and only request power from the grid when needed (i.e., power excess is not stored). We assume that

TABLE I
NOTATION USED THROUGHOUT THIS WORK

Notation	Description
d, D	IoT device d, Count of all d
U_d, L_d	Upper/Lower power consumption bounds of device d
C_d	Energy consumption of d
Z_d	CO_2 emission intensity of d per country
PR_i^d, PRT, N	Preference Rule i for d, Set of all PR_i^d, $N = \|PR\|$
t	Time granularity
P^t	Solar power generation at a certain time
ECT	Energy Consumption Table
EPT	Energy Production Table
GDT	Grid Demand Table
RCR	Residential Consumption History Record

the building is equipped with a Home Energy Management System (HEMS), such as GreenCap. Our HEMS takes as an input the corresponding *Query 1* result-set from database (e.g., operations ECT) and schedules smart appliances to different times or spread their operation over a longer period based on the configured constraints. Our main intention is to optimize an objective function to achieve a trade-off between energy consumption, CO_2 emissions, and comfort, by intelligently planning the operation of appliances to off-peak hours.

We assume that there are D smart devices in the residential household that need to be planned sub-optimally. Let C indicates the hourly energy consumption planning vector, the elements of which ($C_d, d \in [1, D]$) indicate the energy consumption of various devices in the residential building under consideration. Further, let Z represent the hourly CO_2 emission, the elements of which ($Z_d, d \in [1, D]$) denote the CO_2 emissions of various devices in the house. All smart devices have their own upper U_d and lower bounds L_d in regards to power consumption levels. The solar energy generation at a certain time is P^t. We also assume that a user has identified a set of preference rules PR_i^d for each device $d = 1, \ldots, D$, and $N = |PR|$. N is recorded with a meta-service, like the GreenCap system we propose in this work and stored in a database table. GreenCap engages to regularly execute these rules on the IoT appliances. Every PR relies on a specific input context (e.g., location, peak-demand hours, user-configured operation hours), which are also maintained by our system. The proposed system ensures that green planning does not affect to a large extend the user comfort levels, and attempts to retain historical energy consumption levels by considering the Residential Consumption Record (RCR).

B. Problem Formulation

Our proposed technique is optimizing the following **objectives**: (i) *Imported Energy*; and (ii) *User Comfort*.

- **Imported Energy (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 the difference of the consumption C_i and the power generation P, given by:*

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

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

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

Equation 1 ensures the minimization of IE from the grid and consequently the reduction of CO_2 emissions since they are correlated. Equation 2 maintains the UC at a high level according to the pre-configured preference rules table PRT.

We evaluate our objective function as a weighted sum function, where w_1 represents the IE objective and w_2 the UC objective. The sum of w_1 and w_2 is 100%, expressing the trade-off between IE and UC.

$$Weight = w_1^{IE} + w_2^{UC} \tag{3}$$

Additionally, the proposed approach is also discussed with respect to the following:

- **Self-consumed Energy (SE):** occurs when a household consumes energy produced by installations of production, such as photovoltaic panels or mini wind generators.
- **CO_2 Emission ($Z_i(IE_i, k)$):** is the CO_2 emission produced by the actuation of device d given the imported energy consumption IE_i, as well as, the CO_2 emission intensity k of a particular country.
- **CPU Execution Time (F_t):** is the processing time required by the controller for running the optimization fitness function and calculating the output.

III. THE GREENCAP ALGORITHM

In this section, an overview of our algorithmic approach follows along with a local search heuristic we propose in our work, presented in Algorithms 1 and 2, respectively.

A. Overview

The research goal of this work is to develop an intelligent technique that enables users to find a sustainable allocation plan for the operation of smart appliances, a pool of preference rules and a tentative peak-demand history, reducing at the same time CO_2 emissions and the imported energy from the grid.

The GreenCap algorithm is composed of an innovative Memetic Algorithm (MA) we have developed along with a local search heuristic. The MA is based on a traditional genetic algorithm extended by a search technique to further improve user's fitness that may keep high population, diversity and reduce the likelihood premature convergence. Various different approaches have been used in previous publications for the scheduling and planning challenges including Linear Programming, Mixed-Integer Linear Programming, Dynamic Programming, etc. However, these methods face numerous convergence difficulties and they cannot efficiently manage a large number of devices while considering simultaneously the optimization of IE, UC, user costs, and CO_2 emissions.

Algorithm 1 $GreenCap$: generates an energy-efficient plan

Input: ECT: Energy Consumption Table; EPT: Energy Production Table; GDT: Grid Demand Table; PR: Rule Table; g_{max}: max generation; p: population size; ev: Evaluation Function
Output: An energy plan solution ECT^*

1: **GreenCap(**ECT, p, g_{max}, ev, PR**)** ▷ **GreenCap Routine**
2: $RCR \leftarrow getHistoricalConsumptionRecord(ECT)$ ▷ RCR: daily consumption per appliance based on historical data
3: $popul \leftarrow population(ECT, p)$ ▷ $popul$: init p random solutions
4: $fitness(popul, PR)$ ▷ calculate fitness for each chromosome in population
5: **While** $(g = 0; g < g_{max})$ **do** ▷ g: current generation
6: **While** $(i = 0; i < p)$ **do** ▷ i: current iteration
7: $(O_1, O_2) \leftarrow selection(popul)$ ▷ O_1, O_2: random selected from $popul$
8: $(O_1, O_2) \leftarrow crossover((O_1, O_2))$ ▷ crossover at a random point with 90% chance
9: $(O_1, O_2) \leftarrow mutation((O_1, O_2))$ ▷ mutation at a random point with 1% chance
10: $ComfortOptimizationHeuristic(O_1, O_2, RCR)$ ▷ fix consumption of offsprings
11: $(F_1, F_2) \leftarrow evaluate(O_1, O_2, EPT, ev)$ ▷ F_1, F_2: fitness of O_1, O_2
12: $O_3 \leftarrow fittest(F_1, F_2)$ ▷ O_3: fittest offspring between O_1, O_2
13: $F_3 \leftarrow evaluate(O_3, EPT, ev)$ ▷ F_3: fitness of O_3
14: $O \leftarrow fittest(F_1, F_2, F_3)$ ▷ O: fittest offspring between O_1, O_2, O_3
15: $populate(O)$ ▷ populate fittest offspring to least fit in $popul$
16: $i++$ ▷ increase iterations
17: **EndWhile**
18: $g++$ ▷ increase generations
19: **EndWhile**
20: $ECT^* \leftarrow fittest(popul)$; ▷ ECT^*: fittest chromosome in population
21: **return** (ECT^*)

B. GreenCap Memetic Algorithm (MA)

The proposed GreenCap MA, adapts an optimization approach based on a living organism's natural genetic procedure, where each iteration is dealing with various possible solutions. Initially, as shown in Algorithm 1, a chromosome is 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 (see Figure 4). Afterwards, in line 3, a population is generated, which expresses a pool of possible solutions presenting each device's energy consumption state in a specific time-slot. For every possible solution, the fitness function is compared based on the problem's objective metrics, as indicated in line 4, aiming to reduce imported energy and increase user comfort, while considering ECT, EPT, and GDT. Consequently, this will facilitate the reduction of CO_2 emissions, electricity costs, and increase the self-consumption.

In each iteration, the algorithm generates a new population and applies the natural genetic process, crossover and mutation, as shown in lines 8-9. The crossover operates based on a configured probability, thus it is responsible to crossover two chromosome strings and produce a new offspring O, which differ from its parents. The GreenCap mutates the results in order to cause some randomness in the offspring, thus the population's repetition is avoided. The mutation process is based on a very low probability, and is responsible to change one or more chromosome genes from the initial state. Right after, there is an inspired local search function introduced, and well explained in the following sub-section, coined $ComfortOptimization$ heuristic (line 10), which supports the algorithm's efficiency on retrieving better results. When crossover, mutation, and heuristic operations are completed, a new population is produced, where its fitness is compared and evaluated with the previous population (lines 13-15). Moreover, the users' preferences are taken into consideration during the calculation of the fitness function. The PR can be configured through the app or web portal of the proposed GreenCap system accordingly. Each rule adapted is considered as successfully executed, otherwise it is charged with a proportional error cost based on the total set of PR.

C. Comfort Optimization Heuristic

The proposed local search heuristic, coined Comfort Optimization, aims to retain the daily total consumption to its original state, based on the users RCR history, due to fluctuations that may occur from the MA procedures. 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. The total daily consumption per device is calculated, along with the sorted energy production hours, as indicated in lines 4 and 5 of Algorithm 2. The consumption of the generated plan is then compared with the RCR energy consumption history of the devices. In cases where the consumption during the day is less than the RCR state, an allocation (turns ON) of operation in corresponding devices follows, considering the power load bounds (i.e., U_d and L_d) per device and the highest production hours (line 8). Otherwise, the heuristic deallocates (turns OFF) corresponding devices accordingly, as shown in line 10. The goal of this function is to retain and balance the energy consumption levels in case too many devices are turned on or off. Consequently, this adjustment will keep users' comfort at high levels.

IV. THE GREENCAP SYSTEM

In this section, a description of our implemented prototype system is presented, named GreenCap. The GreenCap has been developed using the Laravel MVC framework along with the Linux crontab daemon, and can be easily linked with either open Home Automation Bus (openHAB[2]) or Domoticz[3].

[2]openHAB, URL: https://www.openhab.org/
[3]Domoticz, URL: https://www.domoticz.com/

Algorithm 2 $ComfortOptimization$: preserves consumption to its original state

Input: ECT: Energy Consumption Table (O_1 & O_2); RCR: Residential Consumption History Record; P_{max}: Max power load (max bound) per appliance
Output: An energy plan solution $ECT*$

COH($ECT_{O_1}, ECT_{O_2}, RCR$) ▷ **Comfort Optimization Heuristic**
  For each (day in ECT) ▷ day: iterates daily through year
    While ($h = 0; h < 24$) **do** ▷ h: iterates hourly through a day
      $\vec{cd}[h] \leftarrow \vec{cd}[h] + consumptionPerDevice(h)$
      $sp[h] \leftarrow sortHourlyProduction(h)$ ▷ sorts production
    EndWhile
    If ($cd < day_{RCR}$) **then** ▷ compares consumption plans
      $a \leftarrow allocate(sp, cd, P_{max})$ ▷ allocates operations
    else
      $d \leftarrow deallocate(sp, cd, P_{max})$ ▷ deallocates operations
  return (ECT^*) ▷ returns new energy consumption plan

The GUI is incorporated directly into openHAB's web portal and mobile application, liable for proficient control of IoT appliances and automated management of sustainability-aware Preference Rules (PR) utilization.

The system architecture is composed of the following elements: (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 framework 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, 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 appliances through CU. For the design of the CU, one can extend Domoticz or openHAB framework, which are open source automation software packages for smart residences offering a vast ecosystem of bridges that allow users to directly communicate remotely or locally with IoT appliances. The advantage of this is that we can obtain the greatest level of IoT market compatibility, as IoT integration is a big challenge. For example, consider a user in his residence 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.

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 preference rules 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 database through the mobile application, which has been regulated to smoothly incorporate the definition of PR via a web GUI portal (see Figure 3).

Graphical User Interface (GUI): is constructed using Laravel MVC framework, HTML, and JavaScript, composed of 8000 lines-of-code. 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 4). Information in regards to the state of openHAB IoT appliances is obtained through a Rest API service.

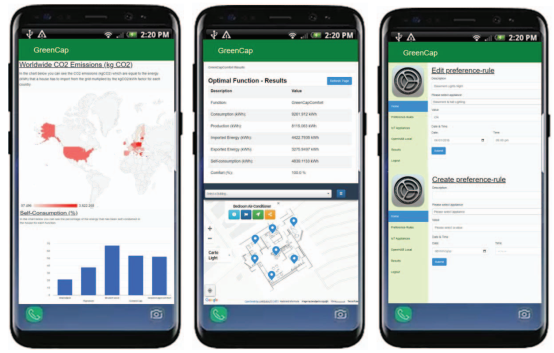

Fig. 3. GreenCap mobile application: Interfaces displaying consumption results, algorithms' performance, and create/edit portals for preference rules.

V. Experimental Methodology & Evaluation

This section presents an experimental evaluation of our proposed system. We start-out with the experimental methodology and setup, followed by various experiments conducted that expose the core benefits of our GreenCap system.

A. Methodology

This section provides details regarding the algorithms, metrics and datasets used for evaluating the performance of the proposed approach.

Testbed: Our evaluation is carried out on our laboratory VMware private datacenter. Our computing node comprises of a Ubuntu 18.04 server image, featuring 6GB of RAM with 4 virtual CPUs (@ 2.40GHz). The image utilizes fast local 10K RPM RAID-5 LSILogic SCSI disks, formatted with VMFS 6 (1MB block size).

Datasets: We have adopted a trace-driven experimental methodology in which three real datasets are fed into our simulator executed on the testbed. The first two datasets were collected by the Laboratory for Advance System Software in the University of Massachusetts Amherst. Particularly, measurements were taken for the energy consumption of various smart devices in different real residences, weather conditions in the places where the houses are located, as well as measurements of solar energy production from various photovoltaic systems. Further, we utilized another set of data to identify peak demand hours collected from the United States Energy Information Administration, which measures the total energy flow transmitted to the energy grid to meet the energy needs of the country.

- **Home Power Usage Dataset:** The 408MB dataset consists of 527,040 measurements per minute from 01/01/2016 until 31/12/2016. The set consists of 20 columns, where the first is the date and time and the

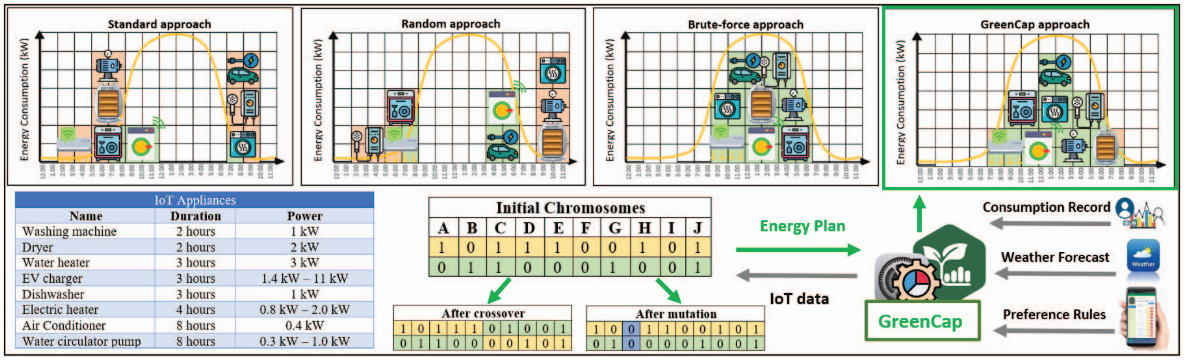

Fig. 4. A daily planning representation of the standard, random, brute-force, and GreenCap methods. The GreenCap is liable to find a sustainable plan for the operation of IoT appliances by only utilizing a Preference Rules (PR) table, a Residential Consumption Record (RCR) history, and a weather forecast. Each IoT device is represented with a letter in the chromosomes stack of the memetic algorithm, and their state is indicated with 1 = ON or 0 = OFF.

remaining 19 columns are energy consumption measurements from 19 different home appliances in kWh.

- **Energy Production Dataset:** The dataset used for the production of energy by a photovoltaic system consists of 65741 measurements per hour from 12/30/2010 to 30/16/2017. It consists of 2 columns, where the first is the timestamp and the second column is the energy production. The utilized PV system is 5.5 kWp, i.e., thus the maximum output it can achieve per hour is 5.5 kWh.
- **Peak Demand Dataset:** The dataset utilized to find peak hours of energy consumption in the US is 63.1MB in size and consists of 579746 measurements per hour, collected and combined by various energy organizations in all US states from 01/01/2016 until 31/12/2016. For our experimental series we preserved essential information that consist the name of the organization, the timestamp, and the total power consumption in kW.

Metrics: The efficiency of the proposed technique to achieve the research goal introduced earlier, is measured by the *Imported Energy* and *User Comfort*, as described in Section II. The mean and standard deviation of the results is shown with error bars in the experiments, based on ten repetitions. The entire experimental series was conducted on an annual basis. During execution, the algorithms consider various preference rules configured by real users.

Baseline Approaches: Here we provide a concise overview of the baseline methods optimizing IE, UC, and F_t.

- **Standard Method:** During the execution, the operational bounds of each device are identified, which will be considered later for optimization tuning. This method ignores IE and provides maximum UC levels.
- **Brute Force Method:** aims to find an optimal solution with the least IE from the grid and CO_2 emissions, and therefore to exploit as much SE as possible. Particularly, it performs an in-depth search (Depth-First Search) to find the best timing for the devices' operation planning, respecting the maximum consumption bounds of each device. However, the UC levels are low and the execution F_t is time consuming.
- **Random Method:** randomly shifts the operation of devices during the day, where the number of performed iterations can be provided as input parameter. Similarly to the previous method, both approaches provide better IE than the Standard method by sacrificing UC, however Random execution F_t is much faster than Brute Force.

B. Performance Evaluation

In this experimental series, we evaluate the performance of the proposed GreenCap algorithm against the baseline methods, with respect to imported energy, self-consumption of electricity and user comfort levels, as shown in Figure 5. The Standard method shows a breakdown of the data as retrieved by the original datasets based on the aforementioned metrics. The initial results based on our baseline approach and before adjusting any smart planning, is 78% for the imported energy, with the worst case of self-consumption at 21% and the best user comfort levels. The results of the Random approach seem to be low in terms of user comfort ($\approx$ 35%) and self-consumption ($\approx$ 38%), and high in regards to the imported energy from the grid ($\approx$ 61%). Considering the self-consumed energy, the best result was obtained by the Brute Force algorithm at around 67% ($\approx$ 6248 kWh) and the imported energy from the grid being only at $\approx$ 32% ($\approx$ 3011 kWh), since it provides an optimal planning solution. However, the user comfort achieved by Brute Force approach ranges at only $\approx$ 40%, being the second worst among the other methods. As observed, the best overall performance was obtained by the GreenCap algorithm with a very high user comfort level at $\approx$ 92%, an impressive self-consumption at around 52% ($\approx$ 4818 kWh), and an imported energy at $\approx$ 48% ($\approx$ 4447 kWh).

The fastest execution time is achieved by Standard method since it simply executes an error calculation ignoring peak demand and energy production hours. The Random approach comes second, as there is not any time consuming process during its execution. An impressive execution time is achieved by GreenCap algorithm as manages to maintain the balance between high user comfort, reasonable levels of self-consumption and imported energy from the grid, while avoiding peak demand hours. The worst period of execution is achieved by Brute Force function, which is to be expected

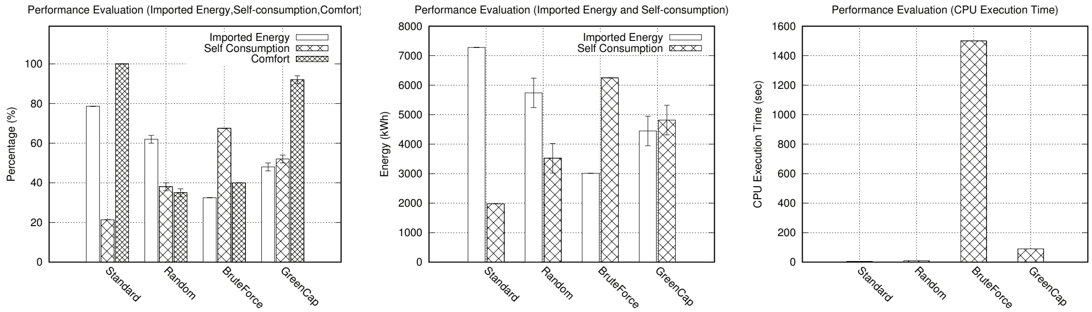

Fig. 5. **Performance Evaluation:** Evaluation in terms of Imported Energy, Self-consumed Energy, User Comfort, and CPU Execution Time.

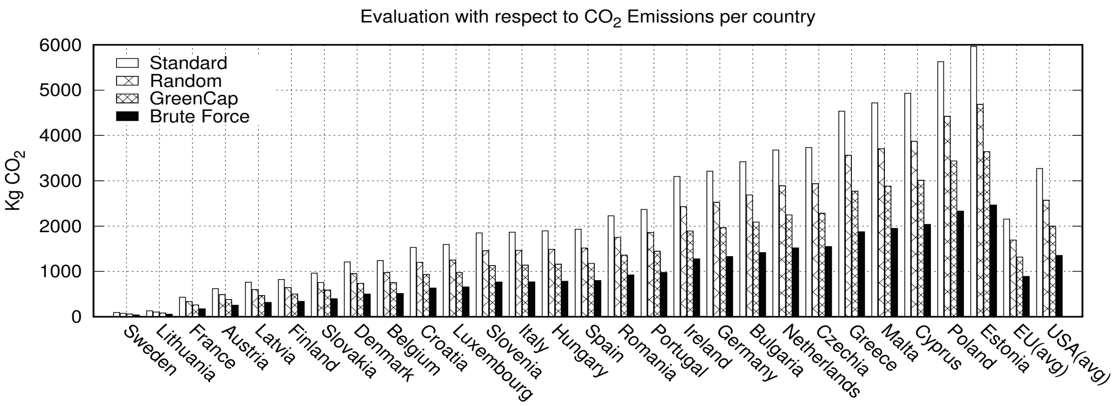

Fig. 6. **CO_2 Evaluation:** Evaluation with respect to CO_2 emissions in different countries based on their kg CO_2 kWh factor.

as it takes much longer to find an optimal solution since it runs through every possible combination. Brute Force is not ideal for this type of operations since it cannot concurrently manage as efficient as GreenCap the problem's decision space, with respect to load shifting, user comfort, peak-demand periods, electricity costs, and CO_2 reduction.

C. CO_2 Evaluation

In the second experimental series, we evaluate the performance of the algorithms with respect to CO_2 emissions. Given that energy is produced in a variety of manners (fossil, renewable, nuclear, etc.), the impact on the environment is typically measured in kg CO_2 emitted per kWh of energy produced [4]. In countries with a high kg CO_2 per kWh factor, this effectively reduces CO_2 pollution but also contributes to the stabilization of the energy grid. The CO_2 emission intensity (kg CO_2) is calculated as the ratio of CO_2 emissions from public electricity production (as a share of CO_2 emissions from public electricity and heat production related to electricity production), and gross electricity production. In Figure 6, it is clearly indicated that in countries with high kg CO_2 per kWh factor, the GreenCap algorithm can reduce up to $\approx 40\%$ of the carbon dioxide emissions. It seems that the Brute Force technique obtains better results, and this is because it exhaustively searches space for an optimal solution. The Random method has the second worst emission levels out of all approaches. On average, we see that most countries have still a long way for becoming CO_2 neutral and that this is an exciting problem space to seek for novel contributions.

[4]In this work, we denote the more typical metric of kg CO_2-eq(uivalent) with only kg CO_2.

VI. Related Work

A comprehensive overview of energy monitoring and prediction for smart homes is provided in this part. Gemello [14], is a system responsible for estimating a home's energy breakdown by utilizing a mechanism to compare similar households with a hardware-based disaggregation approach. A deep latent model for energy disaggregation adapted on variational recurrent neural networks [15], is accountable to predict energy consumption of residential devices that consume less power and have no discernible repeating pattern. The latent variable abstractions assist in great prediction performance on previously unexplored data.

Storing huge amount of IoT data is challenging for efficient execution of the corresponding smart home applications to meet real-time demands, as a substantial amount of the data produced may be unimportant. GradeSense [16], implements a grading mechanism based on multimodal data fusion, integrated with an independent storage module that leverages the grading scheme for efficient storing achieving up to 87% data reduction on average in faster storage tier. Anomalies and data errors are pervasive in time series data, such as IoT sensor readings. The authors in [17], exploit active learning algorithms to detect both temporal errors and events in a single solution that aims at minimizing user interaction and also repair data where possible.

To lessen the cost of collecting and processing large scale of energy data through smart IoT meters to perform basic analytic tasks (i.e., time-of-use pricing in residences), various data storage approaches are supported within distributed computing, which are meant to ease the real-time data analysis process. A smart meter data analysis system was developed in [18], utilizing PostgreSQL and the MADLib machine learning toolkit. Efficiently performing beneficial energy saving computations still remains as an interesting challenge. A real-time stream processing engine, named SPEAR [19], has been developed for spatial-temporal data based on modern big data platforms, which can fully take advantage of the big data ecosystem and IoT cloud computing. It achieves high scalability with dynamic Geo-Hash based spatial partitioning and high throughput with in-memory based processing, requiring minimum latency with the ability to seamlessly handle changing query states.

In terms of home automation strategies, smart thermostats can significantly reduce consumers' energy usage. The Integer Linear Programming for Smart Scheduling (ILPSS) approach improves the HVAC equipment duty cycle and optimizes energy utilization, while maintaining the temperature based on users' comfort zone [20]. In [21], a model was developed to keep the total consumption of each device under a configured threshold with maximum possible benefit, while trying to optimize each scheduled hour of a day. The ant colony was utilized as an optimization technique to solve multiple knapsack problems, enabling smart appliance scheduling. An evolutionary algorithm is proposed in [22] for optimizing the integrated usage of multiple residential energy resources considering stationary storage systems, while focusing on the minimization of energy cost and the end-user's dissatisfaction. In contrast, our work considers peak-demand periods and emphasizes on CO_2 reduction through self-consumption, as we do not take into account storage systems for energy saving.

VII. CONCLUSION

In this work, we propose an innovative IoT data system, coined GreenCap, which utilizes a Green Planning evolutionary algorithm for load shifting of IoT-enabled devices, considering the integration of renewable energy sources, multiple constraints, peak-demand times, and dynamic pricing. Our prototype system serves as proof of concept as it efficiently generates a sustainability-aware plan, in a reasonable response time, obtaining high levels of user comfort 92-99% along with $\approx$52% of self-consumption, while reducing $\approx$35% of the imported energy from the grid and $\approx$40% of CO_2 emissions. In contrast with other developments, the proposed approach does not face convergence difficulties and can efficiently manage a large number of smart devices, while maintaining electricity costs and user comfort, as shown in the experimental series. In the future we plan to extend our research based on Green Planning solutions exposing different essential challenges that need to be addressed, like security and privacy, scalability, interoperability, power fluctuations, and interdisciplinarity.

REFERENCES

[1] S. Althaher, P. Mancarella, and J. Mutale, "Automated demand response from home energy management system under dynamic pricing and power and comfort constraints," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1874–1883, 2015.

[2] N. G. Paterakis, O. Erdinç, I. N. Pappi, A. G. Bakirtzis, and J. P. S. Catalão, "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.

[3] "Home energy management system market by hardware, by communication technology and by software and service - global forecast to 2022," *http://www.marketsandmarkets.com.*, 2016.

[4] 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.

[5] 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 ICDE*, 2018.

[6] T. Huld, H. Ruf, and G. Heilscher, "Self-consumption of electricity by households, effects of pv system size and battery storage," Proceedings of the 29th EUPVSC, Amsterdam, 2014.

[7] S. Charadi, Y. Chaibi, A. Redouane, A. Allouhi, A. El Hasnaoui, and H. Mahmoudi, "Efficiency and energy-loss analysis for hybrid ac/dc distribution systems and microgrids: A review," *International Transactions on Electrical Energy Systems*, vol. 31, no. 12, 2021.

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

[9] M. Dash, G. G. Chua, H. L. Nguyen, G. E. Yap, C. Hong, X. Li, S. P. Krishnaswamy, J. Decraene, and A. S. Nash, "An interactive analytics tool for understanding location semantics and mobility of users using mobile network data," in *IEEE 15th MDM*, vol. 1, 2014, pp. 345–348.

[10] 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.

[11] 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 pages.

[12] V. Portella and H. Shen, "An improved online multidimensional bin packing algorithm," in *20th International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2019, p. 479.

[13] M. Parvania, M. Fotuhi-Firuzabad, and M. Shahidehpour, "Optimal demand response aggregation in wholesale electricity markets," *IEEE Transactions on Smart Grid*, vol. 4, no. 4, pp. 1957–1965, 2013.

[14] N. Batra, A. Singh, and K. Whitehouse, "Gemello: Creating a detailed energy breakdown from just the monthly electricity bill," in *Proceedings of the 22nd ACM SIGKDD*, ser. KDD '16, NY, USA, 2016, p. 431–440.

[15] G. Bejarano, D. DeFazio, and A. Ramesh, "Deep latent generative models for energy disaggregation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 850–857, Jul. 2019.

[16] M. Mallick, N. Ganguly, and S. Bhattacharya, "Gradesense: Gradation aware storage for robust activity recognition in a multimodal smarthome," in *2020 21st IEEE MDM*, 2020, pp. 119–124.

[17] K. H. Le and P. Papotti, "User-driven error detection for time series with events," in *2020 IEEE 36th ICDE*, 2020, pp. 745–757.

[18] X. Liu, L. Golab, and I. F. Ilyas, "Smas: A smart meter data analytics system," in *2015 IEEE 31st ICDE*, 2015, pp. 1476–1479.

[19] F. Baig, D. Teng, J. Kong, and F. Wang, "Spear: Dynamic spatio-temporal query processing over high velocity data streams," in *2021 IEEE 37th ICDE*, 2021, pp. 2279–2284.

[20] D. Petrov, R. Alseghayer, P. K. Chrysanthis, and D. Mosse, "Smart room-by-room hvac scheduling for residential savings and comfort," in *The 10th IEEE International GSCC*, pp. 1–7, 2019.

[21] S. Rahim, S. A. Khan, N. Javaid, N. Shaheen, Z. Iqbal, and G. Rehman, "Towards multiple knapsack problem approach for home energy management in smart grid," in *2015 IEEE 18th International Conference on Network-Based Information Systems*, 2015, pp. 48–52.

[22] A. Soares, l. Gomes, C. H. Antunes, and C. Oliveira, "A customized evolutionary algorithm for multiobjective management of residential energy resources," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 492–501, 2017.