A Sustainable Energy Management Framework for Smart Homes

Soteris Constantinou*, 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@ucy.ac.cy; costa.c@rinnoco.com; panos@cs.pitt.edu; com.ca@frederick.ac.cy; dzeina@ucy.ac.cy

***Abstract*—The escalating global energy crisis and the increasing CO_2 emissions have necessitated the optimization of energy efficiency. The proliferation of Internet of Things (IoT) devices, expected to reach 100 billion by 2030, contributed to this energy crisis and subsequently to the global CO_2 emissions increase. Concomitantly, climate and energy targets have paved the way for an escalating adoption of solar photovoltaic power generation in residences. The IoT integration into home energy management systems holds the potential to yield energy and peak demand savings. Optimizing device planning to mitigate CO_2 emissions poses significant challenges due to the complexity of user-defined preferences and consumption patterns. In this paper, we propose an innovative IoT data platform, coined *Sustainable Energy Management Framework (SEMF)*, which aims to balance the trade-off between the imported energy from the grid, users' comfort, and CO_2 emissions. *SEMF* incorporates a Green Planning evolutionary algorithm, coined *GreenCap*$^+$, to facilitate load shifting of IoT-enabled devices, taking into consideration the integration of renewable energy sources, multiple constraints, peak-demand times, and dynamic pricing. Based on our experimental evaluation utilizing real-world data, our prototype system has outperformed the state-of-the-art approach by up to ≈29% reduction in imported energy, ≈35% increase in self-consumption of renewable energy, and ≈34% decrease in CO_2 emissions, while maintaining a high level of user comfort ≈94%-99%.**

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

I. Introduction

According to the European Commission Green Deal[1], it was decided to reduce net greenhouse gas emissions by at least 55% by 2030, compared to 1990 levels, and become neutral by 2050. In [1], the survey covers the area of renewable planning highlighting various challenges that arise during the integration with the IoT infrastructure. Considering various energy sources utilized for power generation, including fossil fuels, renewable energy, and nuclear power, the environmental impact is commonly quantified in terms of kilograms of CO_2 emitted per kilowatt-hour (kWh) of energy produced. Home Energy Management Systems (HEMS) are instrumental in integrating

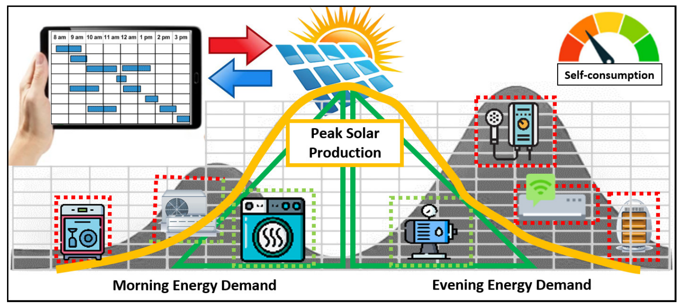

Fig. 1. An illustration of daily energy demand, solar energy production, and device usage. Red dashed lines represent grid-sourced consumption, while green dashed lines depict self-generated renewable energy utilization.

Renewable Energy Sources (RES) by enabling flexible energy demand, crucial for reducing CO_2 emissions, particularly in the context of distributed and weather-dependent RES (see Figure 1). The global count of residential IoT connected devices used in HEMS is anticipated to reach 30.9 billion units by the year 2025[2], and later to 100 billion by 2030 [2].

Green Planning encompasses computational methodologies that strive to enhance environmental quality by implementing load shifting strategies. An essential factor for managing energy consumption and mitigating CO_2 emissions lies in the widespread adoption of the IoT infrastructure utilizing open communication protocols [3]. Therefore, the convergence of energy usage and CO_2 emissions governed by IoT infrastructure can be achieved, by aligning both aspects within a unified framework. Further, the self-consumption of RES holds notable advantages over energy storage batteries, where approximately 17% of the energy is lost due to AC/DC conversion losses and heat dissipation [4]. It embodies a decentralized in-situ strategy that necessitates minimal infrastructure and predominantly relies on intelligent planning algorithms. Empirical evidence has demonstrated that this method yields more than a 70% reduction in energy consumption within domestic households [5], [1].

[1]European Green Deal, URL: https://tinyurl.com/3hbypfum

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

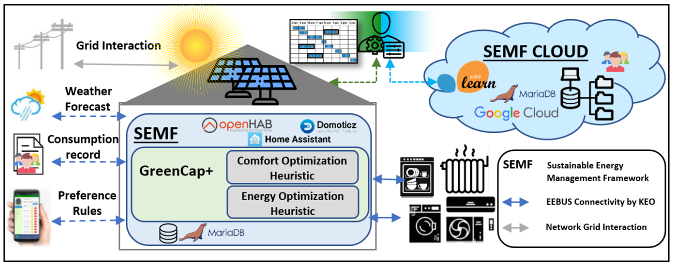

Fig. 2. Overview of the *Sustainable Energy Management Framework (SEMF)* demonstrating involved technologies and the related input data.

In our prior publications [6], [7], we have introduced *Energy Planner (EP)* and *Green Planner (GP)*, integrated in a system called $IMCF^+$. Both, EP and GP, effectively utilize off-the-shelf AI algorithms, namely hill climbing and simulated annealing, for their operations. $IMCF^+$ emphasis lies on "long-term" planning, enabling to compute comprehensive yearly plans by performing less intricate daily computations. Furthermore, a distinct algorithm has been developed called GreenCap[3] [5], [8], which refers to "daily" planning as it attempts to find the best combination for allocating appliances during the day by minimizing the imported energy from the grid. In this study, a new extended version of our developed algorithm is introduced, called *GreenCap*$^+$, incorporated with a new heuristic that considers peak demand and energy production times. *GreenCap*$^+$ is integrated into an IoT framework, coined *Sustainable Energy Management Framework (SEMF)*.

To exemplify the problem's complexity, let us demonstrate a practical example for clarity. Consider solar radiation for 10 hours on a given day, thus, 10 x 60 minutes = 600 time slots on the x-axis. A solar system on a household is usually about 10kWp at most, therefore, let us make the assumption that peak production occurs around noon, as illustrated in Figure 1. Observing the solar production curve, it can be approximated by two triangles of the following size: height = 10kWp / 1kW = 10 and base = 5x60 = 300 minutes, which forms a rectangle (height x base) of 3000 cells to plan each day. The primary challenge lies in populating these cells with device operations, while simultaneously adhering to their respective maximum energy bounds. For instance, the operation of a washing machine (i.e., 2-hour duration ≈1 kW) could be rescheduled in a high production period (i.e., reserving ≈120/3000 cells) by also avoiding peak demand times. The second challenge is the optimization of device planning, while considering user-defined preference rules. The majority of existing solutions encounter difficulties related to convergence, primarily stemming from their limitations in effectively managing a huge number of IoT devices and handling complex multi-objective problems [1].

The main objective of *SEMF* revolves around its core module, called *GreenCap*$^+$, which acts as an IoT data manager tasked with formulating a sustainable plan, while utilizing several input data as shown in Figure 2. The specific problem entails an adaptation of the NP-hard Bin Packing problem [9], known as the 2D packing. This classification implies the absence of a polynomial time algorithm capable of delivering a swift and efficient solution. In contrast, a Brute Force approach, involving back-tracking, has the capacity to compute the optimal solution, however, it demands a substantial amount of time and proves to be infeasible on low-end computing nodes (i.e., Raspberry Pi - 1.5GHz CPU). To tackle the user comfort objective, *SEMF* incorporates a cloud-hosted *AI appliance profiling module*. The *AI module* analyzes the household IoT appliances' energy consumption patterns and operational preferences, thereby generating a pool of recommendation rules, tailored to optimize device scheduling.

SEMF addresses the aforementioned challenges, with a primary focus on reducing CO_2 emissions and the reliance on grid-supplied energy. Given the complex nature of the decision space in the aforementioned problem, a Genetic Algorithm (GA) emerges as the most suitable approach for obtaining a sub-optimal solution. Employing an evolutionary algorithm enables to effectively harness bio-inspired operators, such as mutation, crossover, and selection. The integration of a GA with domain-specific local search heuristics culminates in the development of a *Memetic Algorithm (MA)*. This hybridization yields notable enhancements to user fitness and substantially augments convergence by mitigating the risk of becoming trapped in local optima. The *MA* proposed in this study, coined *GreenCap*$^+$, has been incorporated into our in-house developed pioneering *SEMF* platform, and further integrated with the openHAB framework. The experimental evaluation showcases that the proposed system achieves up to 54% self-consumption of RES and an impressive ≈94% user comfort level. Additionally, it successfully reduces ≈36% of the energy imported from the grid and curtails CO_2 emissions by ≈39%. In summary, the paper's key contributions follow:

- We present an upgraded version of [5] (i.e., utilizing a new heuristic), coined *GreenCap*$^+$, a *Memetic Algorithm* (MA) that can efficiently manage user comfort and reduce the imported energy from the grid, by considering CO_2 emissions, high production and peak demand periods.
- We propose a novel and comprehensive IoT platform, called *Sustainable Energy Management Framework (SEMF)*, designed and incorporated in openHAB system.
- We conducted an extensive experimental evaluation using real and synthetic IoT datasets, consisting of peak electricity demand and solar panel production measurements.
- We have developed a prototype system, illustrating the efficacy of the system in a real-world scenario.

The remainder of the article is organized as follows: Section II presents the related work and Section III the system model with the problem formulation. Section IV describes the proposed algorithms, where Section V outlines our complete system architecture and its internal components. Our experimental methodology and findings are presented in Section VI, and the article is concluded in Section VII.

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

II. Related Work

Home Energy Management Systems (HEMS) enable energy demand reduction by efficiently coordinating the operation of smart appliances, while also enhancing user comfort through energy management practices [10]. Through their sophisticated functionalities, HEMS actively contribute to the mitigation of climate change by supporting the efficient management and optimization of energy consumption within households [11]. The global market for HEMS has witnessed substantial growth, expanding from USD 864.2 million in 2015 to USD 3.15 billion by the year 2022[4].

Energy savings of up to 40% can be achieved using HVAC system incentives and home automation intelligent apps. A method introduced in [11], called Integer Linear Programming for Smart Scheduling (ILPSS), enhances the duty cycle of HVAC equipment, optimizing energy utilization while simultaneously adhering to users' comfort zone with regard to temperature. Moreover, the authors in [12] addressed the issue of chiller sequencing for minimizing HVAC electricity consumption in building operations. A data-driven methodology is proposed for runtime estimation of the chiller's coefficient of performance (COP), a computationally efficient COP prediction model, and an edge-based chiller sequencing framework. Chen L. et al. [13], introduced a model-based offline Reinforcement Learning (RL) algorithm tailored for personalized HVAC systems, adept at efficiently adapting to diverse occupants' thermal preferences with minimal feedback. Riekstin et al. [2], emphasized in mitigating residential electricity consumption and greenhouse gases by employing a time-series prediction model. VALOS [10], is an online scheduling algorithm for HEMS without reliance on predictive elements. It demonstrates a high probability of optimal purchasing timing with minimal computational costs.

In study [14], a symbolic aggregate approximation method and K-Means clustering were used on load data to characterize and estimate users' load patterns based on demographic and socioeconomic information. Additionally, a deep neural network is developed to better capture the correlation between users' consumption habits. The authors in [15], introduce a Day-Ahead Carbon Forecasting system (DACF) that utilizes machine learning to predict the carbon intensity of supplied electricity. DACF incorporates production forecasts for various electricity-generating sources and combines them with the carbon-emission rate of each source. An Economic Model Predictive Control (EMPC) framework is designed in [16] to facilitate demand response for enhancing power grid stability while ensuring occupants' thermal satisfaction in buildings. The controller addresses conflicting objectives by simultaneously optimizing grid stability, measured by grid costs tied to dynamic electricity prices, and occupants' thermal satisfaction, represented by a reference indoor temperature.

The authors in [17] implemented a system for monitoring individual photovoltaic (PV) modules using power line communication (PLC) compliant with HomePlug. The system enables users to access detailed information about the performance of their PV system, identifying abnormalities, and promoting effective energy management. A smart HEMS can be designed to optimize the use of home energy resources in environments with a high penetration of PV systems by utilizing a Natural Aggregation Algorithm (NAA) [18].

The aforementioned solutions face several computational challenges, including the complexity of solving multi-objective optimization problems, which demand significant computational resources for real-time decision-making. Additionally, the need for adaptability to accommodate users' preferences and environmental changes, while also considering RES integration, increases computational demands while necessitating advanced prediction and scheduling algorithms. Further, ensuring the scalability of computational models to suit diverse energy profiles across multiple residences is crucial.

III. System Model & Problem Formulation

In this section, the system model is defined, the problem formulation is articulated, and the main terminology adopted throughout this manuscript is introduced.

A. System Model

Let us assume a house with several residents equipped with a net-metering PV system. The analysis revolves around the household's numerous shiftable smart appliances denoted as D, including electric heaters, washing machine, air conditioners, lights, heat pump, etc. Certain appliances (e.g., refrigerator) are excluded from consideration in our analysis due to their high importance, as they necessitate continuous operation and should always remain turned on. Consequently, the occupants can utilize the PV power generated within the household (i.e., Energy Production Table *EPT*), thus, only drawing power from the grid when necessary, without storing any power surplus (i.e., this work does not consider energy storage technologies). We assume the building is equipped with a Home Energy Management System (HEMS), such as *SEMF*, to facilitate efficient energy management and distribution within the household. The system will efficiently process the relevant result-set obtained from the database (i.e., Energy Consumption Table *ECT* of IoT operations). This data serves as input to a planning algorithm, such as *GreenCap*$^+$, enabling the platform to intelligently schedule smart appliances at different times or distribute their operation over an extended period. *The primary objective of this study is to optimize an objective function that strikes a balanced trade-off among energy consumption, CO_2 emissions, and user comfort.* To achieve this, intelligent planning techniques are employed that strategically schedule the operation of appliances during high production and off-peak periods (i.e., Grid Demand Table *GDT*). The aim is to minimize energy usage and CO_2 emissions while ensuring user comfort remains paramount.

We consider a residence equipped with D smart devices that require sub-optimal planning. Let C represent the hourly energy consumption planning vector, where the elements $(C_d, d \in [1, D])$ denote the energy consumption of various

[4]MarketsandMarkets, URL: https://tinyurl.com/mmv28dzn

devices within the household. Further, let Z denote the hourly CO_2 emission, the elements of which $(Z_d, d \in [1, D])$ define the CO_2 emissions of various devices in the residential building. Each smart device in the system is characterized by its upper bound, denoted by U_d, and lower bound, denoted by L_d, which dictate the permissible power consumption levels for the respective devices.

The solar energy production at a certain time is defined with P^t. We also assume that a user has identified a set of preference rules PR_i^d for each device $d = 1, \dots, D$, and $N = |PR|$. N is recorded with a meta-service, like the *SEMF* platform we propose in this work, and stored in a database table. *GreenCap*$^+$ undertakes the periodic execution of these rules on IoT devices. Each PR in the database is contingent upon a designated input context, which encompasses critical factors such as location, peak-demand hours, and user-configured operation hours. By incorporating PR and considering factors such as energy costs, Residential Consumption Record (*RCR*), and CO_2 emissions, *SEMF* ensures the generation of sustainable energy management plans.

B. Problem Formulation

The efficacy of the proposed technique is gauged by evaluating two key metrics: (i) the *Imported Energy*, which quantifies the amount of energy drawn from the grid; and (ii) the *User Comfort*, which assesses the level of satisfaction and convenience experienced by the users in the context of energy consumption and appliance scheduling.

- **Imported Energy (IE):** *refers to the energy sourced from the grid to enable appliances D to fulfill the predetermined operational requirements set by the residents at a specific time-slot t. The equation is designed to determine a combination of IoT operations that require the minimum IE supplied from the grid. It can be computed as the difference between the energy consumption C_i and the power generation P during time-slot t, given by:*

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

- **User Comfort (UC)**: *is determined by the summation of all executed rules that have been configured by the user. The equation is designed to determine the best allocation of preferences that results in maximum UC. The complete set of preference rules is denoted as N, and each individual rule PR_i takes the value of 1 if it is successfully adapted and subsequently executed, otherwise, it is assigned a value of 0:*

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

The objective function is evaluated as a weighted sum function, where w_1 is associated to the IE objective, and w_2 to the UC objective. Both objectives contribute equally, as we assign equal distribution (50%/50%) to w_1 and w_2, reflecting their equivalent significance in achieving the overall aim. The sum of w_1 and w_2 equals 100%, indicating the trade-off between IE and UC in the optimization approach. These weights determine the relative importance of each objective in the overall optimization process, allowing us to strike an appropriate balance between reducing energy importation from the grid and maximizing user comfort.

$$Total = w_1 * IE + w_2 * UC \qquad (3)$$

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

- **Self-consumed Energy (SE):** pertains to the energy that a household consumes from its own renewable energy generation installations (e.g., PV panels or wind turbines).
- **CO_2 Emission ($Z_i(IE_i, k)$):** represents the CO_2 emission attributed to the operation of device d, contingent upon the imported energy consumption IE_i and the CO_2 emission intensity k characteristic of a specific country.
- **CPU Execution Time (F_t):** denotes the processing duration required by the system to execute the optimization fitness function and compute the desired output.

C. Baseline Approaches

In this section, we present an overview of the baseline methods employed for optimizing IE, UC, and F_t.

- **Standard Method:** consists of the execution phase, where the operational boundaries of devices are identified and recorded, laying the foundation for subsequent optimization tuning. This approach disregards the IE metric and instead prioritizes achieving maximum levels of UC.
- **Brute Force Method:** aims to discover an optimal solution with the minimum IE and least CO_2 emissions, hence, exploit SE. This approach employs an exhaustive search (i.e., Depth-First Search), to meticulously explore the best timing for devices' operation planning while adhering to the maximum consumption bounds of each device. The UC levels are relatively low, and the time consuming F_t makes the computationally intensive method impractical for real-time applications.
- **Random Method:** adopts a stochastic approach by randomly shifting the operation of devices throughout the day. The number of iterations performed in this random process can be specified as an input parameter. In a similar fashion to the previous case, both methods generate better results with respect to IE than the Standard approach, by sacrificing UC levels. However, Random F_t is considerably faster than Brute Force.
- **GreenCap:** is the prior version of our proposed algorithm, which represents a traditional GA algorithm.

IV. THE GREENCAP$^+$ ALGORITHM

This section provides a comprehensive overview of our algorithmic methodology, accompanied by the local search heuristics (i.e., Algorithms 1, 2) proposed in our research.

A. Overview

The research objective of this study is to devise an intelligent technique that empowers users to discover a sustainable allocation plan for operating a group of smart appliances, while considering a pool of preference rules and a tentative peak-demand history. The core aim is to reduce CO_2 emissions and the dependence on imported energy from the grid.

The *GreenCap*$^+$ algorithm is a novel combination of our developed Memetic Algorithm (MA) along with various local search heuristics. The extended version of our work (i.e., *GreenCap*$^+$) compared to its prior iteration [5], introduces a new developed heuristic named *Energy Optimization*, which collaboratively operates with *Comfort Optimization* to efficiently schedule the operation of appliances during off-peak hours and high production periods, boosting this way the system's performance. The MA builds upon the principles of a traditional genetic algorithm and incorporates a search technique aimed at enhancing user fitness while maintaining a diverse population to mitigate premature convergence. Several distinct approaches have been employed in prior research to address scheduling and planning challenges, such as Linear Programming, Machine Learning, and Dynamic Programming [19], [20], [21], [22]. However, these techniques encounter convergence difficulties and often struggle to handle a large number of devices while concurrently optimizing energy consumption, user comfort, costs, and CO_2 emissions. In contrast, MA typically outperforms traditional *GA* due to its hybridization with local search heuristics for additional optimization.

B. GreenCap$^+$ Memetic Algorithm (MA)

The *GreenCap*$^+$ MA adopts an optimization approach inspired by the natural genetic process observed in living organisms. At the beginning, a chromosome represents a residential energy consumption pattern including the status of the smart appliances (ON/OFF), each time-slot's energy consumption, and the length of chromosomes showing the total number of IoT appliances. Thereafter, a population is generated, which expresses a pool of possible solutions presenting each appliance's energy consumption state in a particular time-slot. For every possible solution, the fitness function is evaluated based on the problem's objective metrics, aiming to reduce IE and increase UC levels, while considering the Energy Consumption Table *(ECT)*, the Energy Production Table *(EPT)*, and the Grid Demand Table *(GDT)*. Consequently, this contributes to environmental sustainability by reducing CO_2 emissions and increasing the utilization of renewable energy.

During each iteration, the algorithm generates a new population by applying the genetic operators, crossover and mutation. Crossover involves combining two parent solutions (chromosomes) to create a new offspring O solution, based on a configured probability. By exchanging segments of information between the parents, the crossover operation generates diverse and potentially better solutions that inherit desirable traits from both parents. Mutation occurs to introduce randomness into the offspring population, which helps avoid repetition and promote diversity within the population. Next, the operations

Algorithm 1 $ComfortOptimization$: 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 $ECT*$

1: **COH**(ECT^{O1}, ECT^{O2}, RCR) ▷ **Comfort Optimization Heuristic**
2: **For each** (day in ECT) ▷ day: iterates daily through year
3: **While** ($h = 0; h < 24$) **do** ▷ h: iterates hourly through a day
4: $\vec{cd}[h] \leftarrow \vec{cd}[h] + consumptionPerDevice(h)$
5: $sp[h] \leftarrow sortHourlyProduction(h)$ ▷ sorts production
6: **EndWhile**
7: **If** ($cd < day_{RCR}$) **then** ▷ compares consumption plans
8: $a \leftarrow allocate(sp, cd, P_{max})$ ▷ allocates operations
9: **else**
10: $d \leftarrow deallocate(sp, cd, P_{max})$ ▷ deallocates operations
11: **return** (ECT^*) ▷ returns new energy consumption plan

of the two inspired local search functions follow, coined *Comfort Optimization* and *Energy Optimization* heuristics, which support the algorithm's precision and efficiency. Upon the completion of both crossover, mutation, and heuristic operations, the *GreenCap*$^+$ algorithm generates a new population of candidate solutions. The fitness of this new population is then compared and evaluated against the fitness of the previous population. The fitness evaluation process involves assessing each solution's quality based on the optimization objectives, which include reducing imported energy, increasing user comfort levels, and considering energy production and grid demand trends. By evaluating the fitness of the new population, the algorithm identifies potential improvements and determines whether the solutions have effectively evolved.

Furthermore, the *GreenCap*$^+$ algorithm incorporates users' preferences (i.e., user comfort UC) into the fitness function calculation. Users can configure their preference rules PR through the app or web portal of the proposed *SEMF* system, defining their desired IoT configurations. Each successfully adapted rule is regarded as a successfully executed action, while not adapted rules are assigned a proportional error cost based on the total set of *PR*.

C. Comfort Optimization Heuristic

The proposed local search heuristic, named Comfort Optimization, is designed to maintain the daily total energy consumption at its original level, utilizing the historical records of users' *RCR*. This approach addresses potential fluctuations that might arise as a result of the MA procedures. If the configured settings of PR conflict with the Residential Consumption History Record *(RCR)*, the system gives precedence to users' comfort by adjusting the corresponding preference rules. The algorithm computes the cumulative daily energy consumption for each individual IoT device and arranges the hours of energy production in order (see lines 4-5 of Algorithm 1). Then, the consumption of the generated plan is compared with the historical record *RCR* of the devices. When the consumption for the day is lower than the corresponding *RCR* level, an assignment of operations (activation) to the respective devices takes place. This procedure, shown in line 8, considers the upper U_d and lower L_d power load limits applicable to each device, as well as the hours of peak energy production.

Algorithm 2 *EnergyOptimization*: avoids planning during peak hours

Input: ECT: Energy Consumption Table; GDT: Grid Demand Table
Output: An energy plan solution ECT

```
1:  EOH(ECT^{O3}, GDT)                               ▷ Energy Optimization Heuristic
2:    For each (day in GDT)                          ▷ day: iterates daily through year
3:      While (h = 0; h < 24) do                     ▷ h: iterates hourly through a day
4:        ph[h] ← findPeakHours(h)                   ▷ n peak demand hours
5:        nph[h] ← findNonPeakHours(h)               ▷ n non-peak demand
6:        pd[h] ← findPeakProductionHours(h)         ▷ n production
7:      EndWhile
8:      If (pd ≠ ph) then                            ▷ If ph does not fall into pg
9:        ra ← reallocateAppliances(nph, pg)
10:     ia ← fitnessFunction(day_ECT)                ▷ calculates ia fitness
11:     na ← fitnessFunction(day_ra)                 ▷ calculates na fitness
12:     If (ia > na) then                            ▷ compares allocation results
13:       return (na)                                ▷ returns new allocation as planning solution
14:     else
15:       return (ia)                                ▷ returns initial allocation as planning solution
```

Conversely, if the calculated consumption surpasses the corresponding *RCR* value, the heuristic disengages (deactivates) the relevant devices (i.e., indicated in line 10). The objective of this function is to balance and harmonize energy consumption levels in scenarios where there is an excessive activation or deactivation of devices. Hence, this adaptation will maintain a high level of comfort for users.

D. Energy Optimization Heuristic

The second proposed local search heuristic, coined Energy Optimization, is responsible to shift devices' consumption, while considering peak production times, from a provided number of peak demand hours to non-peak demand hours, as calculated per day in the data flow of the total energy network dataset utilized. In case peak demand hours do not fall within production times, a reallocation of devices occurs, as indicated in lines 8 and 9 of Algorithm 2. Next, both results, from reallocation and initial allocation, are compared using the fitness function as shown in lines 10-12. If the result after reallocation shows that less imported energy is used then the algorithm keeps that solution, otherwise, it is discarded. The goal of this function is to exploit Demand Response, i.e. shifts energy consumption from peak hours where the cost of electricity imported from the power grid is higher compared to non-peak hours where the cost is much lower. In this manner, there is an opportunity for consumers to receive financial incentives when they reduce or shift energy usage during peak load times, and also minimize CO_2 emissions.

Case scenario: A user configures five comfort rules in the following simplified case scenario at the PR table for a house of three rooms. Various input information from the residence's sensors along with certain web services (e.g., peak-demand hours, high energy production times) are sent to *GreenCap*$^+$. The initial operation of *GreenCap*$^+$ is to convert PR to a binary vector, where each vector's position represents a preference rule in PR (see Figure 3). Then, it calculates the approximate daily consumption of each appliance based on the Residential Consumption History Record *(RCR)*. This calculation supports *Comfort Optimization* heuristic, which is incorporated in the system, to balance the energy consumption by avoiding turning on/off too many devices that could also affect the users' experience. A population function randomly generates a solution $s = < 0, 1, 1, 0, 1 >$, meaning that preference rules 2, 3, and 5 will be triggered at a specific time period, thus, 1 and 4 will be discarded. The solution then is evaluated using the fitness function with respect to the imported energy from the grid and user comfort. Further, a new solution is generated by the *Energy Optimization* heuristic $s* =< 0, 0, 1, 0, 1 >$, 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 procedure stops when the full cycle of generations is completed.

E. Performance Analysis

We analytically derive the performance of *GreenCap*$^+$ with respect to the estimated user comfort UC and CO_2 emission Z, which are correlated with the imported energy IE. We adopt a worst-case analysis as it provides a bound for all input. Our experiments in Section VI, show that under real datasets our approach performs more efficiently than the projected worst case. The analysis is based on our system model and ignores any energy not directly associated with the preference rules.

Lemma 1. GreenCap$^+$ *approach has a user comfort of* $F_{UC} = \frac{1}{n}\sum_{i=1}^{D}\sum_{j} UC_j(PR_i), j = 1, \dots, n$*, where* $n > 0$ *is the number of preference rules that will be executed.*

Proof. The algorithm will select at least $n > 0$ preference rules to be executed. In an unrealistic case scenario and for a user comfort equal to zero, *GreenCap*$^+$ will not execute any *PR*, meaning no device will be turned on, providing $F_{UC} = 0$. However, considering a realistic worst case scenario, our algorithm will perform like the *Random* approach since the notion of infinite time is not available, thus, some PR_i could be triggered by turning on a device with a minimum energy consumption. On the other hand, the *Brute-Force* approach by greedily executing all preference rules will offer a $F_{UC} = 1$.

Lemma 2. GreenCap$^+$ *approach has a* CO_2 *emission of* $F_Z = \frac{1}{n}\sum_{i=1}^{D}\sum_{j} Z_j(IE_j(PR_i), z), j = 1, \dots, n$*, where* $n \leq N$ *is the number of preference rules that will be executed, and* z *the* CO_2 *emission of a device.*

Proof. Similarly to *Lemma 1*, the algorithm will select at most $n \leq N$ preference rules to be executed. In the worst case scenario and assuming that all preference rules will be satisfied, *GreenCap*$^+$ will act as the *Standard* approach providing $F_Z = 1$. On the other hand in an unrealistic scenario, not executing any PR_i, meaning no device will be turned on, will provide $F_Z = 0$. However, in a best case scenario considering a realistic setting, our algorithm will act like the *Brute-Force* approach, since it will exhaustively search the entire space to find an optimal solution minimizing the imported energy, and consequently reducing CO_2 emissions.

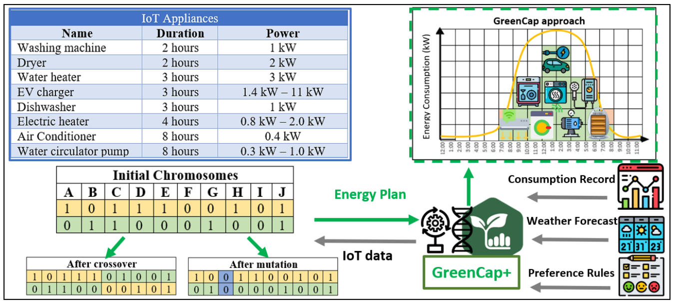

Fig. 3. The *GreenCap*$^{+}$ is liable to find a sustainable plan for the operation of IoT appliances by only utilizing a Preference Rules (PR) table, a Residential Consumption Record (RCR) history, and a weather forecast. Each IoT device is represented with a letter in the chromosomes stack of the proposed MA, and their state is denoted with 1 = ON or 0 = OFF.

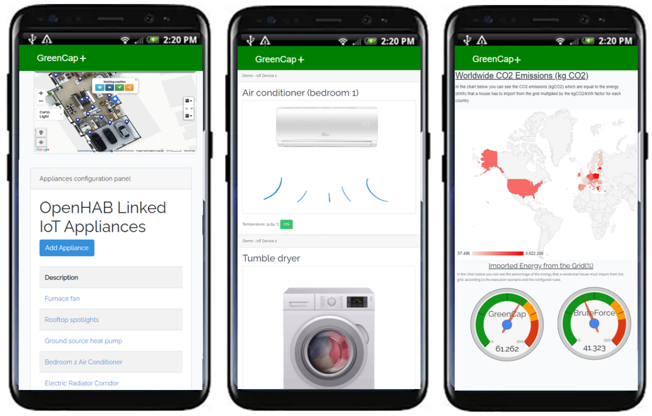

Fig. 4. *GreenCap*$^{+}$ mobile app: Interfaces displaying IoT appliances and their operational mode, PRs, energy consumption and performance results.

V. The Sustainable Energy Management Framework (SEMF)

This section provides an overview of the *SEMF* system architecture, which consists of four layers: (i) Storage Layer; (ii) Network Layer; (iii) Processing Layer; and (iv) Application Layer. The Storage Layer comprises several components, including a relational database (i.e., MariaDB), a file system, and a cloud storage such as those offered by Google or Azure. The Network Layer consists of a custom main *Control Unit (CU)* functioning as a smart residential management application, which allows the system to seamlessly integrate with either open Home Automation Bus (openHAB), Domoticz, or HomeAssistant. The IoT device connectivity is achieved through the industry-standard EEBUS, which offers a robust foundation for efficient communication and control. The Processing Layer is composed of the *GreenCap*$^{+}$ *Controller*, encompassing the entire energy management logic, and the *"AI appliance profiling"*, a module for analyzing energy consumption patterns and producing recommendation PR. The AI component is hosted in the cloud and employs a linear regression technique developed in Python. The Application Layer involved the utilization of the Laravel framework for the development of the Graphical User Interface (GUI) and the Application Programming Interface (API), in conjunction with the Linux crontab daemon. GUI is integrated into the web portal and mobile application of openHAB, enabling efficient control of IoT appliances and automated management of sustainability-aware preferences. The following paragraphs analyze the core elements of the *SEMF* system:

Control Unit (CU)**:** is a Java-based system installed on a device, such as a Raspberry Pi, functioning within a user's localized network. This design choice underscores the commitment to developing a system that is not only technologically advanced but also has low deployment cost due to low computational requirements (i.e., Raspberry Pi). To manage IoT devices based on user-configured preference rules, the CU will establish direct communication with them. Typically, once the users download the mobile application, they will gain interactive control over their appliances through the CU. Considering the design of the CU, one can extend frameworks like Domoticz, HomeAssistant, or openHAB, all of which are open-source home automation software platforms offering an extensive ecosystem of bridges. These bridges enable direct remote or local communication with devices. This approach offers the benefit of achieving compatibility with the IoT market, addressing the substantial challenge of IoT integration.

GreenCap^{+} **Controller:** serves as an augmentative application to the CU, devised to encompass the formulation of the memetic algorithm in conjunction with the GUI and essential storage mechanisms. Its purpose is to facilitate users in customizing their preferences PR, ultimately achieving an energy-aware planning solution. The user-defined settings, stored within a local relational MariaDB database, are passed as parameters in the *GreenCap*$^{+}$ algorithm, constructed as a JAVA module. Users input information into the database via the mobile application, which has been adjusted to integrate the configuration of PR through a web-based GUI.

AI appliance profiling module: constitutes a pivotal aspect of the system architecture, contributing to enhanced user comfort. Hosted within the cloud, the AI appliance profiling module is a sophisticated component that employs a linear regression technique [23]. Written in Python, this module scrutinizes residents' energy consumption patterns and operational preferences of their IoT appliances. It then generates a comprehensive array of recommendation rules, meticulously tailored to optimize energy usage according to each user's distinct comfort requirements. *Scikit-learn* has been utilized, a robust machine learning library, which provides a comprehensive suite of tools for data analysis. Through continual learning and adaptation, the AI component refines its recommendations over time, ensuring alignment with users' evolving needs.

The *AI appliance profiling* component resides on a cloud server side, while each CU (i.e., each residential house) acts as a client. When considering the technical intricacies, the AI-based module leverages consumption patterns (i.e., time series data) and IoT device information as input parameters. These data representing individual residential houses' consumption habits, are processed separately without sharing raw data centrally. Each CU client, collects and sends energy consumption behaviors to the cloud, where the processed data are then utilized to generate recommendation rules that optimize each client's comfort. The AI-generated *PRs* encapsulate valuable insights extracted from the aggregated data, securing sensitive information. Users can adapt the recommended preferences or discard them and add their own customized rules.

Graphical User Interface (GUI): developed in Laravel MVC framework, utilizing JavaScript, and HTML. The GUI is orchestrated by the NGINX web server, which is compatible with Raspberry Pi. The web-based interface is composed of the PR portal and the presentation of results stemming from the *GreenCap*$^+$ sustainability-aware process. The PR portal facilitates users in configuring their IoT preference settings for specific date-time slots (refer to Figure 4). Retrieving data concerning the status of openHAB IoT appliances is accomplished through the utilization of a RESTful API service. An Indoor Navigation Service has been also incorporated into our platform, called Anyplace[5], to enhance indoor mapping of IoT functionalities and venue construction.

Managerial Implications: *SEMF* presents substantial opportunities for public service organizations to enhance energy efficiency efforts. By disseminating information about its benefits and functionalities, alongside collaboration with stakeholders and offering incentives, such as grants or subsidies, adoption can be encouraged. Furthermore, its applicability extends to other entities like university campuses, municipal facilities, or factories, where energy-saving measures are crucial for sustainability goals. Moreover, the system could participate in demand response events initiated by utilities or grid operators. By intelligently managing connected devices, it could help reduce overall electricity demand during peak periods, contributing to grid stability and reliability.

VI. Experimental Methodology & Evaluation

This section provides an assessment of our proposed system. We commence with an outline of the experimental methodology and setup, subsequently detailing the series of experiments conducted to underscore the advantages of *GreenCap*$^+$.

A. Methodology

This section furnishes information considering the metrics utilized, the algorithms and datasets employed for the evaluation of the proposed methodology's performance.

Testbed: The evaluation process is conducted on our laboratory VMware private datacenter. The computational node employed is configured with a Ubuntu 18.04 server image, 4GB of RAM and powered by 4 virtual CPUs, operating at 2.40GHz. It leverages high-speed of 10K RPM RAID-5 LSILogic SCSI disks, formatted with VMFS 6.

Datasets: We have embraced a trace-driven experimental approach, characterized by the utilization of real datasets as inputs into our simulator. The first two datasets were retrieved by the Laboratory for Advanced System Software (LASS) at the University of Massachusetts Amherst, as part of the research project titled "Optimizing Energy Consumption in Smart Homes". Specifically, measurements were gathered for the energy consumption of diverse appliances within residential settings (such as ovens, heat pumps, washing machines, etc.), accompanied by data related to weather conditions and solar energy production. An additional dataset was employed to discern the hours of peak energy demand, sourced from the U.S. Energy Information Administration, which gauges the aggregate energy transmission directed towards the energy grid. Moreover, a thorough analysis was conducted based on the events and energy consumption patterns within these datasets to gain insights into user behavior, facilitating the understanding and generation of realistic preference rules.

- **Residential Energy-Consumption Dataset [FLAT]:** The 408MB dataset encompasses 527,040 data points per minute of a flat/apartment, spanning from January 1st, 2016 to December 31st, 2016. It consists of 20 columns, with the first column indicating the date and time, while the ensuing 19 columns encapsulate energy consumption measurements (in kilowatt-hours) for 19 distinct household appliances.
- **Energy-Production Dataset:** The dataset employed to model energy generation through a PV system encompasses 65,741 measurements per hour spanning from December 30th, 2010 to December 16th, 2017. It comprises two columns; the first one denotes the timestamp, while the second quantifies the energy production. The PV system's capacity is 5.5 kWp, thereby signifying its maximum hourly output potential as 5.5 kWh.
- **Peak-Demand Dataset:** The dataset employed to identify periods of peak energy consumption encompasses a volume of 63.1MB and 579,746 hourly measurements. The data was collected by numerous energy organizations across all states of the US from January 1st, 2016 to December 31st, 2016.

To assess the scalability of our propositions for buildings of different scales, we have generated two realistic datasets by expanding the above onto various residential building sizes. The resulting datasets are the following:

- **Residential Energy-Consumption Dataset [HOUSE]:** A dataset for a three-bedroom house was created by replicating and blending the readings, then scaling up the original dataset by a factor of four. The number of IoT devices, energy consumption, and preferences is proportionally increased following realistic patterns as we have a larger number of residents and requirements. The PV system's capacity for the house scenario is 10 kWp.
- **Residential Energy-Consumption Dataset [DORMS]:** A dataset for a university campus (dormitories) was also synthetically generated from the original datasets. It comprises 50 dormitory apartments, each containing two bedrooms. As the number of residents increases, the quantity of IoT devices, energy consumption, and preferences is correspondingly augmented. The PV system's capacity for the dorms scenario is 50 kWp.

[5]Anyplace, URL: https://anyplace.cs.ucy.ac.cy/

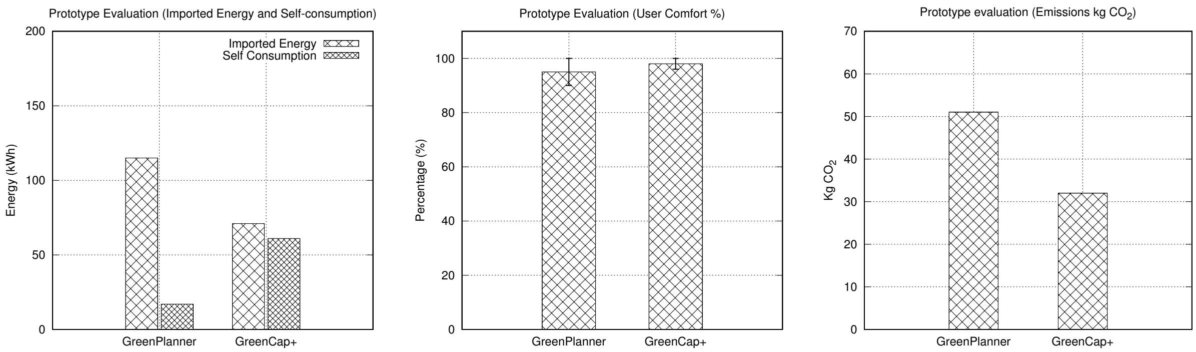

Fig. 5. **Prototype Evaluation:** Weekly system evaluation in terms of UC, SE and IE, based on the Standard method performance.

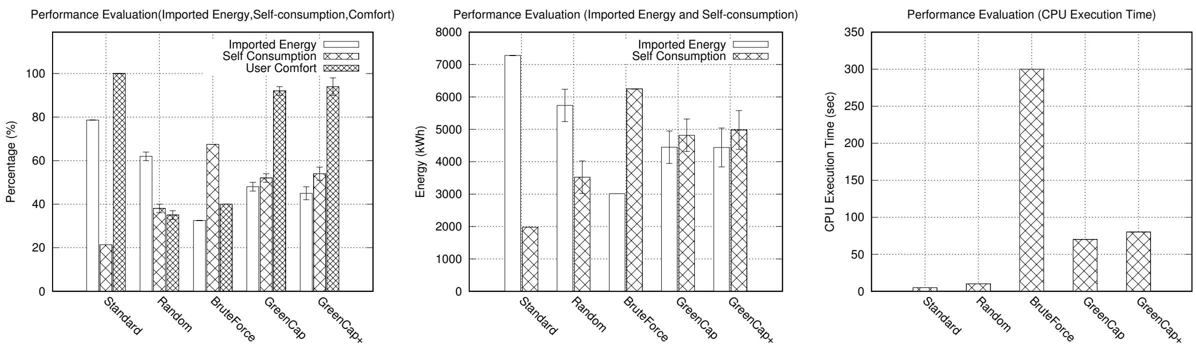

Fig. 6. **Performance Evaluation:** Evaluation in terms of IE, SE, UC, and F_t, based on the Standard method performance.

Metrics: The effectiveness of the methodology in attaining the previously introduced research objective is assessed through two key performance indicators, namely, *Imported Energy* and *User Comfort*, as expounded in Section III. The mean and standard deviation values derived from the results are depicted with error bars in all following experimental analyses, based on ten iterations for each scenario. Experimental series C, D, E, and F were conducted over the course of a year.

B. Prototype System Evaluation

In this series of experiments, we assess the performance of the proposed *GreenCap*$^+$ algorithm integrated into our *SEMF* prototype system in comparison to the *Green Planner* algorithm embedded in the $IMCF^+$ framework. According to our prior work [6], *Green Planner* outperformed $IFTTT$[6], which was state-of-the-art, by 18% increased user comfort, 30% less energy consumption, and 40% reduction in CO_2 emissions. We deployed a live instance of our real prototype system for a household of three individuals over the course of one week. Each user specified certain preference rules through a mobile app that interacts with the management system. We utilize data from the OpenWeatherMap[7], a service that provides real-time weather information for various locations around the world, to measure environmental parameters (i.e., sunlight, temperature). The evaluation is based on self-consumption, user comfort, and CO_2 emissions, as illustrated in Figure 5. The *Green Planner* algorithm achieved $\approx$ 96.5% user comfort, around 51 kg of CO_2 emissions, $\approx$ 17 kWh of self-consumption, and roughly 115 kWh imported from the grid. The *GreenCap*$^+$ algorithm yielded a user comfort rate of $\approx$ 99%, CO_2 emissions of about 32 kg, self-consumption totaling around 61 kWh, and an import of $\approx$ 71 kWh from the grid. Evidently, the *GreenCap*$^+$ algorithm delivers notably better results concerning self-consumption and CO_2 emissions. Moreover, the user comfort levels achieved by both methods are comparably close, although the *GreenCap*$^+$ approach demands slightly more execution time.

C. Performance Evaluation

In the subsequent set of experiments, we assess the efficacy performance of the *GreenCap*$^+$ algorithm in comparison to all other methods, with respect to imported energy, self-consumption of electricity, and user comfort levels, as indicated in Figure 6. The Standard approach provides a breakdown of the data extracted from the original datasets according to the metrics mentioned earlier. It reveals a 78% imported energy from the grid, a self-consumption of 21%, and the best level of user comfort achieved. The outcome of the Random method exhibits a relatively diminished user comfort level $\approx$35% and a self-consumption rate $\approx$38%, while revealing a higher imported energy rate from the grid at $\approx$61%. In terms of self-consumed energy, the best outcome was achieved by the Brute Force algorithm at approximately 67% ($\approx$ 6248 kWh), accompanied by a relatively low imported energy rate from the grid of about 32% ($\approx$ 3011 kWh), due to its capability

[6]IFTTT - Automate business & home, URL: https://ifttt.com/
[7]OpenWeatherMap, URL: https://openweathermap.org/

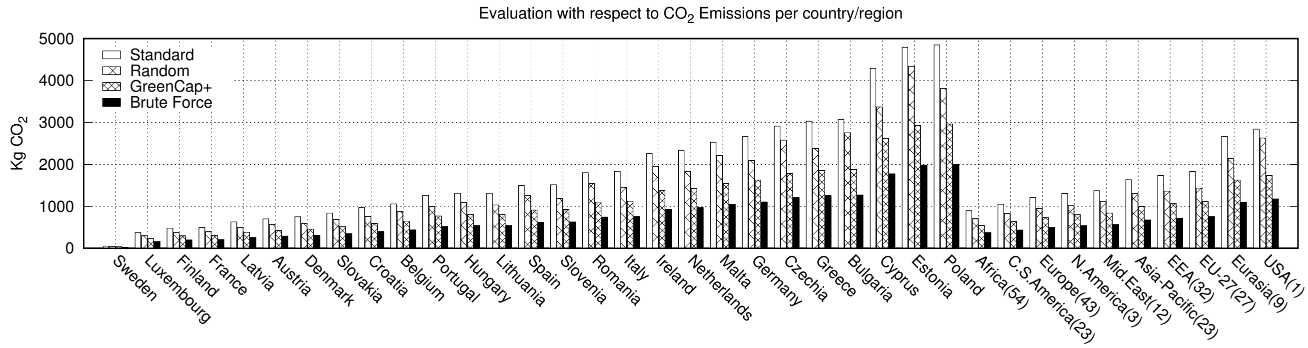

Fig. 7. **CO_2 Evaluation:** Evaluation with respect to CO_2 emissions in different countries/regions based on their kg CO_2 per kWh intensity factor. The figures next to the regions denote the number of countries within each continent.

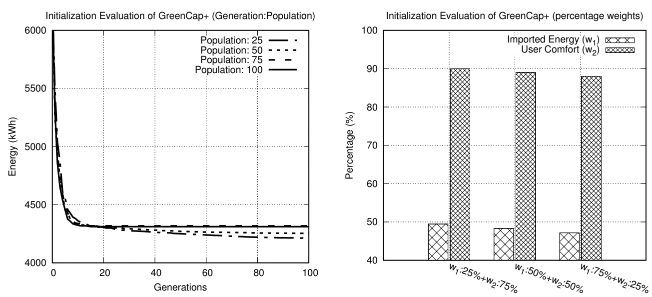

Fig. 8. **Initialization Evaluation:** Evaluation of the *GreenCap*$^+$ algorithm in terms of the IE (w_1) from the grid based and the UC (w_2) on various initialization techniques, based on the Standard method performance.

to provide an optimal planning solution. However, the user comfort obtained by Brute Force ranges at only $\approx$ 40%, being the second lowest among the other evaluated approaches. As evident from the results, the *GreenCap*$^+$ algorithm showcased the best overall performance and also outperformed its prior version [5], which represents a traditional GA algorithm, presenting an exceptional user comfort level of approximately 94%. It achieved a remarkable self-consumption rate of about 54% ($\approx$ 4980 kWh) and managed imported energy at around 43-47% ($\approx$ 4440 kWh).

The Standard approach demonstrates the fastest execution time due to its straightforward error calculation without considering energy production hours or peak demand times. Following closely is the Random approach, which features a relatively swift execution owing to its absence of time-consuming processes. The *GreenCap*$^+$ algorithm achieves a reasonable execution time while effectively balancing all the objectives. The Brute Force algorithm exhibits the worst execution time, as anticipated, due to its exhaustive search through all possible solution combinations.

D. CO_2 Evaluation

In the third phase of our experimental series, we assess the algorithms' performance in terms of CO_2 emissions. Recognizing that energy originates from diverse sources such as fossil fuels, renewables, and nuclear, the environmental impact is conventionally quantified in terms of kilograms of CO_2 emitted per kilowatt-hour (kWh) of energy produced. In countries characterized by a high kg CO_2 per kWh coefficient, this leads to a reduction in CO_2 emissions and contributes to the stabilization of the grid. The *GreenCap*$^+$ has been applied across various countries/regions around the globe considering a household scenario, displaying the CO_2 emission intensity (kg CO_2 per kWh) stemming from electricity generation, sourced from U.S. Energy Information Administration (EIA)[8], European Environment Agency (EEA)[9], and International Energy Agency (IEA)[10]. The intensity of CO_2 emissions is derived from the ratio of CO_2 emissions from public electricity production (relative to CO_2 emissions from public electricity and heat production) by the gross electricity production.

As depicted in Figure 7, the data clearly reveals that in regions characterized by high kg CO_2 per kWh factor, *GreenCap*$^+$ showcases the capability to decrease CO_2 up to 45% in comparison to the *Standard* approach. Notably, the Brute Force method outperforms other methods in terms of emission reduction, attributed to its exhaustive exploration of the solution space. The Random technique exhibits relatively higher emission levels compared to most approaches (i.e., second worst approach). On average, it is evident that a significant number of regions are still at a considerable distance from achieving CO_2 neutrality. This underscores the pressing need for innovative contributions in this domain, which presents an exciting opportunity to address the challenges associated with reducing carbon emissions and advancing sustainability.

E. Initialization Evaluation

In the fourth experiment, we evaluate the performance of the proposed *GreenCap*$^+$ algorithm using various population figures and different percentage weights, as described in Section III, with respect to the imported energy weight w_1 from the grid, and the user comfort weight w_2. After conducting multiple tests and tuning parameters, it has been determined that increasing the number of generations results in a longer execution time for the algorithm to complete. However, more generations do not necessarily mean a better performance outcome. As shown in the left plot of Figure 8, during the initial $\approx$10 generations the reduction of imported energy from the energy network is noticeable, where in the following

[8]U.S. Energy Information Admin., URL: https://tinyurl.com/3bzspb9c
[9]European Environment Agency, URL: https://tinyurl.com/46vh8tt2
[10]International Energy Agency, URL: https://tinyurl.com/3w48mm8v

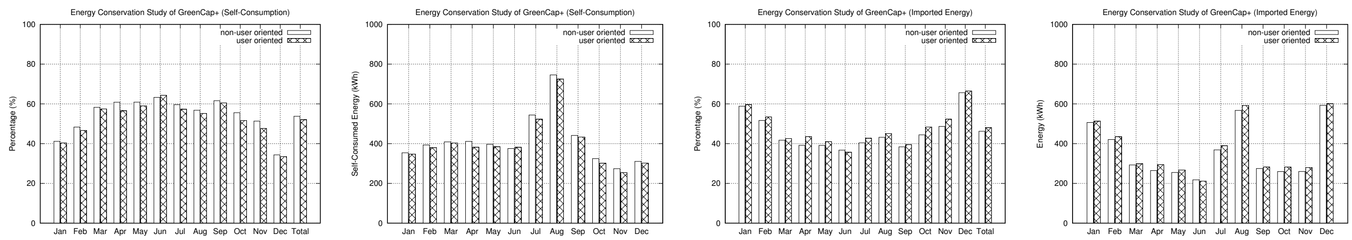

Fig. 9. **Energy Conservation Study:** Evaluation in terms of the SE and the IE, considering different UC values, based on the Standard method performance.

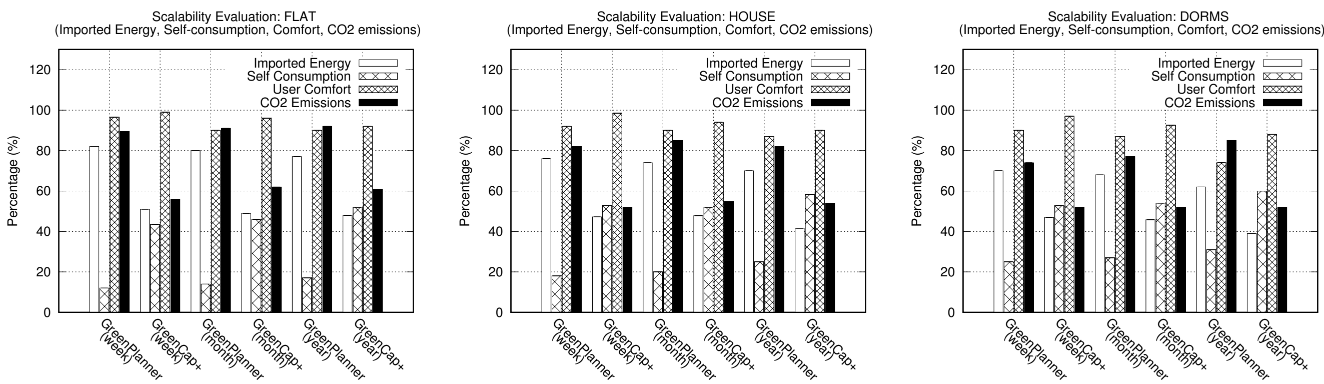

Fig. 10. **Scalability Evaluation:** Evaluation in terms of IE, SE, UC, and F_t, based on the Standard method performance, while considering different scale of residences over the course of a week, a month, and a year.

generations, the improvement becomes less significant. In a similar manner, the larger the population of the algorithm we select, the longer the execution time will last. Because of the randomness that exists in the genetic algorithm, it was chosen to use a population equal to 50, as it provides a fairly large range of random solutions. By examining the first plot of Figure 8, we identified a point where the population and generation settings led to minimal improvement. This point can be considered indicative of high convergence using those specific configuration parameters (i.e., population = 50, generation = 10). Based on our empirical findings, the upper bound limits for both parameters were set to 100, and are sufficient enough to allow the algorithm reach a solution.

To find the ideal balanced allocation between the two metrics we conducted several experiments adjusting the weights of the objective function and observed the overall performance, as shown in the right plot of Figure 8. The results in the middle plot are displayed with respect to the percentage of the total consumption (considered as 100%) against the w_1 and w_2. The trade-off is clearly presented through the various weight percentages utilized in the fitness function. According to the results obtained, the best allocation case scenario was obtained using w_1=75% for the imported energy and w_2=25% for the user comfort. Thus, the more we reduce the w_1 percentage, the more energy is imported from the grid, while the algorithm manages to maintain the user's comfort quite high (≈90%), even with just w_2=25% allocation.

F. Energy Conservation Study

In the fifth experimental series, we evaluate the monthly performance of the proposed *GreenCap*$^+$ algorithm, in terms of user comfort. In Figure 9, we observe the algorithm's performance and trade-off for each month with respect to self-consumption and imported energy, while comparing two different cases. In the first case, the *GreenCap*$^+$ completely ignores the user comfort (non-user oriented), meaning that none of the preference rules configured by users are considered, where in the second case all preference rules are considered accordingly (user oriented).

In Figure 9, the algorithm manages to consume more solar energy produced by the PV system when user comfort is not considered. Specifically, the *GreenCap*$^+$ manages to self-consume 54% of the total consumption when preference rules are completely ignored, while the self-consumption when user comfort is taken into consideration is about 51%. We also observe that the IE from the grid for each month of the year is slightly less when the algorithm does not consider UC, which is expected. More specifically, the total input energy when preference rules are not considered is about 46%, while when the algorithm takes into account UC is about 49%.

G. Scalability Evaluation

In the last experimental series, we assess the scalability performance of the *GreenCap*$^+$ algorithm in comparison to state-of-the-art *Green Planner*, with respect to imported energy, self-consumption of electricity, user comfort levels, and CO_2 emissions, as indicated in Figure 10. Specifically, we tested the system on realistic large-scale scenarios (utilizing real and synthetic data as described in Section VI-A) involving a flat apartment, a house, and dormitories over the course of a week, a month, and a year. Figure 10 demonstrates that *GreenCap*$^+$ outperforms *Green Planner* in all cases, while it shows a balanced performance considering the increase of users, IoT devices, preferences, and energy consumption. In the flat case *GreenCap*$^+$ outperforms *Green Planner* by $IE{\approx}29\%$, $UC{\approx}2.5\%$, $SE{\approx}35\%$, and $Z{\approx}34\%$. According to the house scenario *GreenCap*$^+$ achieved better results than *Green Planner* by $IE{\approx}28\%$, $UC{\approx}6\%$, $SE{\approx}32\%$, and $Z{\approx}30\%$. In the last case of dorms, better performance is achieved from *GreenCap*$^+$ by $IE{\approx}23\%$, $UC{\approx}14\%$, $SE{\approx}28\%$, and $Z{\approx}27\%$. In summary, our system proficiently performs under different scales and time frames, highlighting its robustness and effectiveness in diverse settings.

VII. Conclusion & Future Work

The majority of the works in the existing literature established an energy management model to either cater to user comfort levels or electricity costs. However, none of them simultaneously cater to CO_2 emissions, integration of RES, multiple constraints, peak demand times, and user comfort to fully utilize the innovative smart metering infrastructure. In this work, an intelligent evolutionary algorithm is proposed, called *GreenCap*$^+$, integrated into a framework, coined *SEMF*, that enables users to find an energy efficient allocation plan for the operation of a set of IoT devices along with a pool of preference rules, while considering peak-demand and high production periods. The system's sophisticated heuristics are adept at analyzing historical energy consumption patterns, considering weather conditions, and operational preferences through a centralized *CU* to curate a personalized energy management experience for an easier and greener life. According to our experimental evaluation utilizing real-world data, *SEMF* has outperformed the state-of-the-art approach by up to ≈29% reduction in imported energy from the grid, ≈35% increase in self-consumption of renewable energy, and ≈34% decrease in CO_2 emissions, while maintaining a high level of user comfort ≈94%-99%. In the future, we intend to expand our research on Green Planning solutions, tackling challenges, such as interoperability, scalability, and fluctuations in power supply, and detection of unusual energy-pattern behavior.

References

[1] S. Constantinou, A. Konstantinidis, and D. Zeinalipour-Yazti, "Green planning systems for self-consumption of renewable energy," *IEEE Internet Computing (IC'23)*, vol. 27, no. 1, pp. 34–42, 2023.

[2] A. Riekstin, A. Langevin, T. Dandres, G. Gagnon, and M. Cheriet, "Time series-based ghg emissions prediction for smart homes," *IEEE Transactions on Sustainable Computing (TSUSC'20)*, vol. 5, no. 01, pp. 134–146, jan 2020.

[3] M. Poess, R. Nambiar, K. Kulkarni, C. Narasimhadevara, T. Rabl, and H. Jacobsen, "Analysis of tpcx-iot: The first industry standard benchmark for iot gateway systems," in *34th IEEE International Conference on Data Engineering (ICDE'18)*, pp. 1519–1530, 2018.

[4] R. Schrage, P. H. Tiemann, and A. Niesse, "A multi-criteria metaheuristic algorithm for distributed optimization of electric energy storage," *ACM SIGEnergy Energy Informatics Review (SIGEnergy'22)*, vol. 2, no. 4, p. 44–59, feb 2023.

[5] 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'23)*, pp. 65–72, 2023.

[6] S. Constantinou, A. Konstantinidis, P. K. Chrysanthis, and D. Zeinalipour-Yazti, "Green planning of iot home automation workflows in smart buildings," *ACM Transactions on Internet of Things (TIOT'22)*, vol. 3, no. 4, pp. 1–30, sep 2022.

[7] S. Constantinou, A. Konstantinidis, D. Zeinalipour-Yazti, and P. K. Chrysanthis, "The iot meta-control firewall," in *37th IEEE International Conference on Data Engineering (ICDE'21)*, pp. 2523–2534, 2021.

[8] S. Constantinou, N. Polycarpou, C. Costa, A. Konstantinidis, P. K. Chrysanthis, and D. Zeinalipour-Yazti, "Greencap: A platform for solar self-consumption using iot data," in *24th IEEE International Conference on Mobile Data Management (MDM'23)*, pp. 176–179, 2023.

[9] D. S. Abdul-Minaam, W. M. Al-Mutairi, M. A. Awad, and W. H. El-Ashmawi, "An adaptive fitness-dependent optimizer for the one-dimensional bin packing problem," *IEEE Access*, vol. 8, pp. 97 959–97 974, 2020.

[10] C. Xia, W. Li, X. Chang, T. Zhao, and A. Y. Zomaya, "Lightweight online scheduling for home energy management systems under uncertainty," *IEEE Transactions on Sustainable Computing (TSUSC'22)*, vol. 7, pp. 887–898, oct 2022.

[11] D. Petrov, R. Alseghayer, P. K. Chrysanthis, and D. Mosse, "Smart room-by-room hvac scheduling for residential savings and comfort," in *10th IEEE International Green and Sustainable Computing Conference (IGSC'19)*, pp. 1–7, 2019.

[12] Z. Zheng, Q. Chen, C. Fan, N. Guan, A. Vishwanath, D. Wang, and F. Liu, "An edge based data-driven chiller sequencing framework for hvac electricity consumption reduction in commercial buildings," *IEEE Transactions on Sustainable Computing (TSUSC'22)*, vol. 7, no. 03, pp. 487–498, jul 2022.

[13] L. Chen, F. Meng, and Y. Zhang, "Fast human-in-the-loop control for hvac systems via meta-learning and model-based offline reinforcement learning," *IEEE Transactions on Sustainable Computing (TSUSC'23)*, vol. 8, no. 03, pp. 504–521, 2023.

[14] Z. Wei and H. Wang, "Characterizing residential load patterns by household demographic and socioeconomic factors," in *12th ACM International Conference on Future Energy Systems (e-Energy'21)*, pp. 244–248, 2021.

[15] D. Maji, R. K. Sitaraman, and P. Shenoy, "Dacf: Day-ahead carbon intensity forecasting of power grids using machine learning," in *13th ACM International Conference on Future Energy Systems (e-Energy'22)*, pp. 188–192, 2022.

[16] M. Frahm, P. Zwickel, J. Wachter, F. Langner, P. Strauch, J. Matthes, and V. Hagenmeyer, "Occupant-oriented economic model predictive control for demand response in buildings," in *13th ACM International Conference on Future Energy Systems (e-Energy'22)*, pp. 354–360, 2022.

[17] J. Han, C. Choi, W. Park, I. Lee, and S. Kim, "Plc-based photovoltaic system management for smart home energy management system," *IEEE Transactions on Consumer Electronics (TCE'14)*, vol. 60, no. 2, pp. 184–189, DOI: 10.1109/TCE.2014.6851992, 2014.

[18] F. Luo, G. Ranzi, C. Wan, Z. Xu, and Z. Y. Dong, "A multistage home energy management system with residential photovoltaic penetration," *IEEE Transactions on Industrial Informatics (TII'19)*, vol. 15, no. 1, pp. 116–126, 2019.

[19] M. da Silva, A. Gamatie, G. Sassatelli, M. Poss, and M. Robert, "Optimization of data and energy migrations in mini data centers for carbon-neutral computing," *IEEE Transactions on Sustainable Computing (TSUSC'23)*, vol. 8, no. 01, pp. 68–81, jan 2023.

[20] G. Xie, X. Xiao, H. Peng, R. Li, and K. Li, "A survey of low-energy parallel scheduling algorithms," *IEEE Transactions on Sustainable Computing (TSUSC'22)*, vol. 7, no. 01, pp. 27–46, jan 2022.

[21] A. Pahlevan, M. Zapater, A. K. Coskun, and D. Atienza, "Ecogreen: Electricity cost optimization for green datacenters in emerging power markets," *IEEE Transactions on Sustainable Computing (TSUSC'21)*, vol. 6, no. 02, pp. 289–305, 2021.

[22] R. Chen and X. Wang, "Situation-aware orchestration of resource allocation and task scheduling for collaborative rendering in iot visualization," *IEEE Transactions on Sustainable Computing (TSUSC'22)*, vol. 7, no. 4, pp. 935–949, 2022.

[23] C. Cervellera and D. Macciò, "Local linear regression for function learning: An analysis based on sample discrepancy," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS'14)*, vol. 25, no. 11, pp. 2086–2098, 2014.