**REPUBLIC OF TURKEY**
**YILDIZ TECHNICAL UNIVERSITY**
**GRADUATE SCHOOL OF SCIENCE AND ENGINEERING**

# PREDICTION OF REGIONAL USAGE HABITS PARAMETERS BY GENETIC ALGORITHM FOR DOMESTIC ELECTRICAL ENERGY DEMAND AND PRODUCTION OF DAILY LOAD PROFILES

**Ferhat YILMAZ**

MASTER OF SCIENCE THESIS
Department of Control and Automation Engineering
Program of Control and Automation Engineering

Supervisor
Assoc. Prof. Dr. Yavuz EREN

July, 2023

**REPUBLIC OF TURKEY**
**YILDIZ TECHNICAL UNIVERSITY**
**GRADUATE SCHOOL OF SCIENCE AND ENGINEERING**

# PREDICTION OF REGIONAL USAGE HABITS PARAMETERS BY GENETIC ALGORITHM FOR DOMESTIC ELECTRICAL ENERGY DEMAND AND PRODUCTION OF DAILY LOAD PROFILES

A thesis submitted by Ferhat YILMAZ in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 21.07.2023 in Department of Control and Automation Engineering, Program of Control and Automation Engineering.

Assoc. Prof. Dr. Yavuz EREN
Yıldız Technical University
Supervisor

**Approved By the Examining Committee**

Assoc. Prof. Dr. Yavuz EREN, Supervisor
Yıldız Technical University _____

Asst. Prof. Dr. Uğur YILDIRAN, Member
Yıldız Technical University _____

Assoc. Prof. Dr. Tufan KUMBASAR, Member
İstanbul Technical University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Prediction of Regional Usage Habits Parameters by Genetic Algorithm for Domestic Electrical Energy Demand and Production of Daily Load Profiles supervised by my supervisor, Assoc. Prof. Dr. Yavuz EREN. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Ferhat YILMAZ

Signature

*Dedicated to Islam...*

# ACKNOWLEDGEMENTS

I would like to express my gratitude to my esteemed advisor, Yavuz Eren, who helped me at every stage of the realization of this study and offered solutions to the problems I encountered. In addition, I would like to thank my esteemed professors Uğur Yıldıran and Tufan Kumbasar, who were in the thesis final jury, for their contributions to the improvement of my thesis.

Finally, I would like to express my gratitude to my one and only wife Şûrâ Yılmaz, who never lost faith in me and provided all kinds of support.

Ferhat YILMAZ

# TABLE OF CONTENTS

**LIST OF SYMBOLS** vii

**LIST OF ABBREVIATIONS** ix

**LIST OF FIGURES** x

**LIST OF TABLES** xi

**ABSTRACT** xii

**ÖZET** xiii

**1 INTRODUCTION** 1

1.1 Motivation . . . . . 1

1.2 Literature Survey and Background . . . . . 2

1.3 Contribution . . . . . 5

**2 METHODOLOGY** 8

2.1 Overview . . . . . 8

2.2 Data Quality Criteria . . . . . 9

2.3 Mathematical Background . . . . . 10

2.4 Parameter Estimation with Genetic Algorithm . . . . . 13

2.5 Data Generation . . . . . 20

2.6 Verification . . . . . 22

**3 IMPLEMENTATION** 23

3.1 Development and Simulation Environment . . . . . 23

3.2 Configuration . . . . . 23

3.3 Coding . . . . . 28

**4 SIMULATION STUDIES** 35

4.1 Scope and Objective . . . . . 35

4.2 Household Types . . . . . 36

4.3 Appliances and EVs . . . . . 36

4.4 Simulation Parameters . . . . . 38

4.5 Results . . . . . 40

**5 CONCLUSION** **46**

**REFERENCES** **48**

**PUBLICATIONS FROM THE THESIS** **53**

# LIST OF SYMBOLS

| | |
|---|---|
| $a$ | Appliances. |
| $g(u_t, a_j)$ | Cost function for $u_t$-$a_j$. |
| $c$ | CPDF for the chromosomes. |
| $D_{a_j}(k)$ | Daily temporal total assignment count for $a_j$ up to the sector $k$. |
| $K_{ud_t,a_j}$ | Estimated daily parameter representing the usage habit of $u_t - a_j$. |
| $K_{u_t,a_j}$ | Estimated general parameter representing the usage habit of $u_t - a_j$. |
| $K_{u_t-a_j}$ | Estimation vector $\left(\begin{bmatrix} K_{u_t,a_j} & K_{ud_t,a_j} & K_{uw_t,a_j} \end{bmatrix}\right)$ |
| $K_{uw_t,a_j}$ | Estimated weekly parameter representing the usage habit of $u_t - a_j$. |
| $l_m$ | Fitness function for RWSM. |
| $l$ | Fitness function values for each chromosome. |
| $K_G(t)$ | Gene value. |
| $j, m, t$ | Index variables for $a$, $l$ and $u$, respectively. |
| $l'_m$ | Modified $l_m$. |
| $P'_m$ | Modified $P_m$. |
| $MR_C$ | Mutation rate of each chromosome. |
| $MR_G$ | Mutation rate of each gene. |
| $P_{M,C}(\lambda)$ | Mutation probability of each chromosome. |
| $P_{M,G}(\lambda)$ | Mutation probability of each gene. |
| $u_t$-$a_j$ | Operation mode of $a_j$ corresponding to $u_t$. |
| $f(u_t, a_j)$ | Operation status function for $u_t$-$a_j$. |

| | |
|---|---|
| $P_m$ | Probability function for RWSM. |
| $n$ | Population number for each generation. |
| $RAF(u_t, a_j, i, k)$ | RAF for $u_t$-$a_j$. |
| $r$ | Random real number from the set of $[0, 1]$. |
| $U(\gamma)$ | Random sign function. |
| $RWO(u_t, a_j)$ | RWO for $u_t$-$a_j$ |
| $h_{st,a_j}(i, k)$ | Start time function of $a_j$ at the sector $i$ in the day of $k$. |
| $S$ | Sum of each element in$l$. |
| $SWO(a_j)$ | SWO for $a_j$. |
| $T_{p,pro}(u_t, k)$ | Temporal TP in $u_t$. |
| $i$ | The sectors in a day. |
| $k$ | The sectors in a week. |
| $J$ | Total number of appliances. |
| $T$ | Total number of user types. |
| $TP(u_t)$ | TP in $u_t$. |
| $TWO(u_t, a_j)$ | TWO for $u_t$-$a_j$. |
| $u$ | User types. |
| $W_{a_j}(k)$ | Weekly temporal total assignment count for $a_j$ up to the sector $k$. |

# LIST OF ABBREVIATIONS

| | |
|---|---|
| APM | Automatic power meter. |
| BU | Bottom-up |
| CPDF | Cumulative probability distribution function. |
| DR | Demand response |
| EV | Electrical vehicle. |
| GA | Genetic algorithm. |
| JSON | JavaScript Object Notation |
| MD | Measured data |
| OECS | The Organisation for Economic Co-operation and Development |
| PRNG | Pseudorandom number generator |
| RAF | Random assignment function. |
| RMS | Root mean square. |
| RWO | Reference weekly total operation. |
| RWSM | Roulette-wheel selection method. |
| SD | Survey data |
| STD | Statistical data |
| SWO | Personal survey data on weekly operation time. |
| TD | Top-down |
| TP | Total number of persons. |
| TRNG | True random number generator |
| TWO | Total weekly operation time. |

# LIST OF FIGURES

**Figure 2.1** Overall architecture . . . . . 8
**Figure 2.2** The generated load profile quality criteria . . . . . 9
**Figure 2.3** Generic algorithm operator types . . . . . 14
**Figure 2.4** Fundamental principle of generic algorithm . . . . . 15
**Figure 2.5** CPDF applied $P'$ set . . . . . 18
**Figure 3.1** Load profile generation process . . . . . 29
**Figure 4.1** Daily average load profile for each household type . . . . . 41
**Figure 4.2** Daily average load profile for each household type without EV . 42
**Figure 4.3** Overall average of the user type profiles . . . . . 43
**Figure 4.4** RMS value of the user type profiles . . . . . 43
**Figure 4.5** Comparison of the user load profiles for each day . . . . . 44
**Figure 4.6** 2000 profiles corresponding to the user types . . . . . 45

# LIST OF TABLES

**Table 1.1** Review of Some Remarkable Studies on the Profile Generation for Household Loads . . . . . 6
**Table 3.1** Contents of the configuration files . . . . . 24
**Table 4.1** Household Types . . . . . 36
**Table 4.2** Daily Operation of Appliances[59–62] . . . . . 37
**Table 4.3** Electric vehicle characteristics . . . . . 38
**Table 4.4** Genetic algorithm paramaters . . . . . 39
**Table 4.5** Simulation parameters . . . . . 40

# ABSTRACT

## Prediction of Regional Usage Habits Parameters by Genetic Algorithm for Domestic Electrical Energy Demand and Production of Daily Load Profiles

Ferhat YILMAZ

Department of Control and Automation Engineering
Master of Science Thesis

Supervisor: Assoc. Prof. Dr. Yavuz EREN

Considering millions of electricity consumers worldwide, significant energy savings can be achieved through smart pricing or incentives for load shifting. Therefore, predicting user load profiles has become crucial for electricity distribution companies. However, obtaining the data used for load profile estimation requires expensive measurement devices. In this study, a customized genetic algorithm is proposed to estimate load profile types without the need for measurement devices. Real-time scenarios for residential consumers are considered using regional surveys or statistical data to generate load profiles. Energy demands are calculated based on usage habit parameters, considering statistical energy consumption of user types. Genetic algorithms are employed to predict the usage habit parameters, aiming to generate load profiles with minimal error. The obtained results are tested and demonstrated to be usable in MATLAB®.

**Keywords:** Genetic algorithm, load profile generation, roulette wheel selection method, usage habit parameters.

**YILDIZ TECHNICAL UNIVERSITY**
**GRADUATE SCHOOL OF SCIENCE AND ENGINEERING**

# ÖZET

## Evsel Elektriksel Enerji Talebi için Genetik Algoritma ile Bölgesel Kullanım Alışkanlıkları Parametrelerinin Kestirilmesi ve Günlük Yük Profillerinin Üretilmesi

Ferhat YILMAZ

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç. Dr. Yavuz EREN

Dünya özelinde milyonlarca elektrik tüketicisi göz önüne alındığında akıllı fiyatlandırma veya yük kaydırmaya yöenlik teşvikler ile enerji noktasında büyük tasarruflar elde edilebilir. Bu nedenle, elektrik enerjisi dağıtımı şirketleri için kullanıcıların yük profillerini kestirme problemi önem kazanmıştır. Ancak, yük profillerinin kestiriminde yararlanılan veriyi elde etmek pahalı ölçüm cihazları gerektirmektedir. Bu çalışmada ölçüm cihazlarına ihtiyaç duymadan yük profil tiplerinin kestirimi için özelleştirilmiş bir genetik algoritma ile yeni bir prosedür sunulmuştur. Yük profilleri oluşturmak için bölgesel anket veya istatistiksel verileri kullanarak, konut tipi tüketiciler açısından gerçek zamanlı senaryolar ele alınmıştır. Kullanıcı tiplerinin istatistiksel enerji tüketimi dikkate alınarak kullanım alışkanlık parametreleri üzerinden enerji talepleri hesaplanmıştır. Kullanım alışkanlık parametreleri genetik algoritma tarafından tahmin edilip, yük profillerinin en az hata ile üretilmesi hedeflenmiştir. Elde edilen sonuçların MATLAB® üzerinde test edilip kullanılabilir olduğu gösterilmiştir.

**Anahtar Kelimeler:** Genetik algoritma, yük profili üretimi, rulet çarkı seçimi yöntemi, kullanıcı alışkanlık parametreleri.

**YILDIZ TEKNİK ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**

# 1
# INTRODUCTION

## 1.1 Motivation

The study of residential load profile generation presents a unique and compelling opportunity to probe deeply into the intricate dynamics of electricity consumption patterns within households. Hence, it contributes a distinct and transformative impact on the efficient and sustainable management of energy resources. This perspective discovers the subtle yet critical patterns of electricity usage during diverse time frames, thus enabling the design and implementation of intelligent strategies that optimize energy distribution, ensuring a resilient and stable power supply. An exhaustive and comprehensive exploration of this domain necessitates a distinct and interdisciplinary knowledge base, encompassing fields such as data analytics, machine learning, power engineering, and renewable energy integration. As experts in the field of residential load profile generation, researchers and practitioners contribute significantly to the development of smart grids, establishing smooth adaptability to the evolving energy demands, while fostering the integration of consumers and providers in the growing digital era.

The fascinating aspect of this field is its ability to advance sustainability on multiple subjects. By acquiring expertise in load profile generation, scholars and professionals can empower homeowners and consumers with valuable insights into their energy consumption habits, inspiring a culture of energy conservation and efficiency that resonates through communities and transcends national boundaries. Additionally, their research activities have the capacity to accelerate the advancement of innovative technologies, such as demand response systems and virtual power plants, thus catalyzing a swift transition towards a future where renewable energy sources flourish, leading to a substantial reduction in our reliance on fossil fuels.

In summary, residential load profile generation has significant importance in terms of energy management and planning, demand response programs, energy efficiency

and conservation, renewable energy integration, infrastructure development, grid stability and reliability and billing and tariff design.

Considering energy management and planning, to understand the consumption patterns of residential electricity users is crucial for energy providers and policymakers to effectively manage and plan the energy grid. Load profiles help identify peak demand periods, which allows for better resource allocation and ensures a stable and reliable power supply. For practical applications, load profiles are essential in the success of the demand response (DR) programs. These programs aim to adjust electricity usage during peak times, reducing strain on the grid and preventing blackouts. Accurate load profiles enable the implementation of targeted demand response strategies, incentivizing users to shift their consumption to off-peak hours. Moreover, energy efficiency and conservation issue is linked by analyzing load profiles. Researchers can easily identify energy-wasting patterns and develop strategies to promote energy efficiency and conservation. This information can be used to design energy-saving initiatives and encourage consumers to adopt energy-efficient practices. From a broader perspective, With the increasing integration of renewable energy sources like solar and wind, load profiles become essential in matching electricity generation to demand. Understanding peak demand periods helps ensure that renewable energy sources are effectively utilized and integrated into the grid. Also, load profiles help us to infrastructure development activities. Knowing the load patterns of residential users helps utilities determine the capacity requirements and plan infrastructure upgrades accordingly. Finally, it can be stated that the accurate load profiling deeply help us for grid stability and reliability as well as billing and tariff design. Utilities can anticipate fluctuations in demand and take proactive measures to avoid overloading or under utilizing the grid, minimizing the risk of power outages. Moreover, load profiles play a role in designing electricity tariffs and billing systems. Time-of-use tariffs, for instance, are based on load profiles, where electricity rates vary depending on peak and off-peak hours.

## 1.2 Literature Survey and Background

The power system is under strict requirements to achieve a balance between electricity supply and consumption, even under challenging conditions. Efforts to explore the rich potential of renewables have led to a diverse supply side operation in the power system. The European Commission aims to cover at least 27% of final energy consumption from renewables by 2030 [1], making the intermittent nature of renewable sources a significant factor in the energy balance on the supply side.

Over the years, there has been a shift in the share of electrical consumption among different sectors in The Organisation for Economic Co-operation and Development (OECD) countries. The share of commercial public services and residential sectors grew from 48.4% in 1974 to 62.5% in 2017, while the industry's power demand decreased from 48.7% to 32.2% during the same period [2]. Meanwhile, the share of electricity demand from the residential sector remained relatively stable, increasing from 28.7% to 30.8%. However, residential load demand exhibits higher temporal variability compared to other sectors due to sudden and unpredictable changes in human activities and the stochastic nature of appliance usage patterns.

Understanding the temporal variability of residential load consumption can be utilized to compensate for the intermittent nature of renewables through incentive DR applications. Obtaining realistic and accurate predictions of residential load profiles is vital for developing energy-based policies for both supply and demand sides. Load profiling, with more integrated solutions, allows for clustering analysis to provide characteristic actions and a responsive framework for adaptive energy balance policies.

Load profiles serve as a fundamental tool for customer classification; however, their utility extends beyond that to encompass various other applications. These include devising strategies for supply and demand, designing pricing or tariff schemes, financial planning and reporting, integrating renewable energy sources, and enhancing energy dispatching [3]. Load profiles can be influenced by several factors, such as temporal demand patterns, the diversity of appliances used, user preferences, and more. Consequently, employing detail-oriented algorithms capable of capturing the temporal dynamics of these factors leads to more dynamic and realistic class representations, ensuring better alignment with real-world usage scenarios.

The realism of load profiles necessitates addressing issues of accuracy, randomness, and diversity. To achieve this, various basic models have been explored in the literature. Studies on load profile generation mainly classify methods into two categories: top-down (TD) and bottom-up (BU) models. The TD technique focuses on macro scale aspects to extract the load template, emphasizing the most influential characteristics for building the load profile [4]. On the other hand, BU approaches for generating aggregated profiles require detailed data on various factors such as physical specifications of houses, appliance ownership, usage periods, frequency of appliance usage, number of occupants, and predicted activities [4]. In summary, these techniques often face challenges of either high complexity or low accuracy [5].

The literature offers a diverse array of techniques, including clustering, fuzzy inference modeling, neural networks, and stochastic control, among others. Automatic Power Meter (APM) based load profile generation has been explored as a means to achieve faster peak time analysis and more accurate consumption portraits [6]. However, while this approach enhances accuracy, it poses a drawback in terms of cost due to the installation of metering equipment. Various studies have utilized APM devices to identify representative demand profiles and show the diversity of electricity usage patterns [7] [8]. A stream structure of data is employed in another study, featuring an adaptive and recursive clustering method for updating load profiles with online data feedback [9]. Furthermore, a hybrid approach combining k-means clustering and generative adversarial networks is used to generate load profiles with dynamic and stochastic behavior [5]. A combination of neural networks and fuzzy inference systems is developed to approximate the nonlinear nature of the load [10].

In an innovative approach, generative adversarial networks are employed as a black box approach to generating load profiles, eliminating the need for assumptions on building types and occupant activity profiles [3]. Additionally, bottom-up approaches have been explored, including one that establishes a model for classifying load demands of residential buildings in Singapore [11]. Gaussian model-based load profiling techniques are introduced to capture complex behaviors exhibited by residential load demands [12]. Markov processes are used in various studies to produce realistic load profiles, considering daily load volatility and time-dependency for power demands [13] [14].

Moreover, load profile prediction methods without metering data have also been investigated. For example, a tool based on Markov chains is formulated to obtain residential load profiles considering seasonal effects [15]. In another study, survey data on load demand is utilized, and statistical methods are employed to classify typical load patterns of customers [16].

Table 1.1 presents a compilation of significant studies on household load profile generation, categorized by published year, adopted model, data types used for validation or model training, origin country of the data, and proposed techniques. The models are classified into two main forms: BU and TD models. Data types are categorized into three cases: Measured data (MD), survey data (SD), and statistical data (STD). MD can be obtained through APMs or other measuring devices. SD is collected through specialized surveys that capture individual preferences in electrical appliance usage patterns. On the other hand, STD is derived from public reports or sectoral analyses that observe historical customer behavior

trends. Generating household profiles based on STD requires more specific analysis compared to SD. The proposed techniques for the listed studies are detailed in the last column of Table 1.1.

Considering Table-1, load profile generation studies have been conducted, especially in USA-centered research, using different data sources and regression analysis method [17] [18] [19] [20]. Regression analysis is primarily a statistical method, which has been replaced by meta-heuristic methods in recent years. However, statistical approaches are not limited to regression analysis. Studies conducted in Australia [21], UK [22] [23], Spain [24], and Bulgaria [25] have also used statistical methods in different years. Many of the studies that prefer statistical methods utilize MD data. In fact, this method was commonly used before the widespread adoption of meta-heuristic algorithms. Nowadays, statistical methods are frequently employed to make sense of data pieces that constitute a whole. The selection of machine learning and meta-heuristic methods in studies often indicates the suitability of the chosen approach and data type in our study with respect to the literature. For instance, in studies conducted in UK [26] [27] [28], Taiwan [16], and USA [29], this observation can be noticed.

After reviewing the studies presented in Table 1.1, certain observations and comments can be made. The load generation process necessitates high-resolution and credible data to accurately reflect the volatility and random characteristics of load profiles. As a result, the trend in applied techniques has shifted towards learning and meta-heuristic approaches to predict the intricate details of demand profiles. In this context, evolutionary algorithms fall under the umbrella of meta-heuristic techniques, known for being naturally inspired and possessing the capability to produce realistic random load profiles. Consequently, our proposed approach is a BU model employing a customized genetic algorithm (GA) technique to generate random load profiles based on SD. Further elaboration on the techniques will be provided in the subsequent sections.

## 1.3 Contribution

Reviewing the available literature on residential load profile generation, the primary objective is to present more realistic and cost-effective methods. Traditionally, achieving higher realism in load profiles has been associated with increased costs due to the need for additional APM devices. However, this paper introduces a novel approach for load profile generation that solely relies on regional survey or sampled data regarding usage habits. This innovative method enables the acquisition of

**Table 1.1** Review of Some Remarkable Studies on the Profile Generation for Household Loads

| Year | Reference | Model (BU-TD) | Data Type (MD-SD-StD) & Origin Country | Proposed Technique to Generate Household Load Profiles |
|---|---|---|---|---|
| 1980 | [17] | BU | SD/USA | Regression analysis |
| 1984 | [18] | BU | MD/USA | Regression analysis |
| 1985 | [30] | BU | MD-SD/USA | Probability analysis |
| 1991 | [21] | BU | MD/Australia | Statistical analysis |
| 1994 | [31] | BU | MD-SD/Italy | Probability analysis with Monte carlo extraction process |
| 2000 | [19] | BU | SD/USA | Regression analysis |
| 2000 | [32] | BU | MD/Australia | Optimal estimation |
| 2002 | [33] | TD | MD/Greece | Neural networks |
| 2004 | [34] | BU | SD/Japan | Occupant activity based simulation |
| 2005 | [26] | BU | SD/UK | Cluster analysis based on the scenarios with occupancy |
| 2006 | [35] | BU | MD-STD/Finland | Probability analysis |
| 2007 | [16] | BU | SD/Taiwan | Probabilistic neural network approach |
| 2009 | [36] | TD | MD-SD/Sweden | Occupant activity based analysis |
| 2009 | [37] | BU | MD-SD/Canada | Scenario based statistical analysis |
| 2010 | [38] | BU | MD-SD/Sweden | Activity based Markov chain |
| 2010 | [22] | BU | MD-SD/UK | Statistical analysis |
| 2012 | [39] | BU | SD/Australia | Physics-based analysis |
| 2012 | [23] | TD | MD/UK | Statistical analysis |
| 2013 | [15] | TD | MD/Belgium | Markov model based clustering |
| 2013 | [20] | BU | SD/USA | Regression analysis |
| 2013 | [29] | BU | SD/USA | Heterogeneous Markov chain |
| 2013 | [27] | BU | SD/UK | Markov chain |
| 2014 | [40] | BU | MD/Germany | Non-homogeneous Markov chain |
| 2014 | [28] | BU | SD/UK | Markov chain by Monte Carlo approach |
| 2014 | [24] | BU | MD-SD/Spain | Statistical analysis |
| 2014 | [11] | BU | MD-STD/Singapour | Probability analysis |
| 2014 | [41] | BU | SD/USA | Occupant behaviour analysis |
| 2015 | [42] | BU | STD/Germany | Probability analysis |
| 2016 | [4] | TD | SD/UK | Multiple Gaussian functions |
| 2016 | [9] | TD | MD/Austria | Probabilistic analysis |
| 2016 | [43] | BU | MD-SD/Sweden | Probabilistic&Empirical analysis |
| 2016 | [44] | BU | MD-SD/Netherland | Markov Chain model by Monte Carlo approach |
| 2017 | [45] | BU | MD-SD/UK | Probabilistic analysis |
| 2017 | [13] | TD | MD/USA | Markov process |
| 2018 | [46] | BU | MD/China | Probabilisitc analysis with occupant behaviours |
| 2018 | [47] | BU | MD/Iran | Probabilistic analysis |
| 2018 | [48] | TD | MD/Bulgaria | Statistical analysis |
| 2019 | [25] | BU | SD/Denmark | Occupant activity based analysis |
| 2019 | [5] | TD | MD/Ireland | Generative adversarial network |

credible load profiling without the necessity of installing expensive equipment in each residence for demand response applications, clustering, and other purposes. Therefore, the study's main contributions can be summarized as follows:

- A pioneering metaheuristic approach utilizing GA is introduced to generate load profiles solely based on regional statistical data. This innovative method presents a fresh perspective as the customized GA ensures the coherence of the generated profiles with the underlying data. Furthermore, the adopted methodology adheres to the fundamental principles of synthetic load profiling, including accuracy, randomness, and diversity. Additionally, by adjusting the stopping criteria of GA, the correlation between the referenced data and the generated profiles can be effectively controlled.
- As an innovative approach, the allocation of appliances to the profiles is carried out through a random function, significantly impacting the probability

subroutines during the assignment process. This function incorporates variables such as the number of occupants in households, previous appliance assignments, and usage habit parameters. The overall procedure comprises two main parts: (i) the estimation of usage habit parameters using regional statistics, and (ii) utilizing these parameters to generate any desired number of load profiles.

- The customized GA for load profile generation has been tailored to meet the specific requirements of the problem. To enhance the selection operator, we have adapted the roulette-wheel selection method (RWSM), modifying it to provide a cumulative distribution property. This modification assigns greater weight to lower values in the cost function, resulting in a relatively larger sector in the wheel for chromosomes with lower fitness values. Additionally, a hetero-probabilistic mutation operator has been introduced, dynamically influencing the frequency of mutation cases and rearranging the weight of the parameters based on the mutation frequency. Notably, we have utilized the recombination operator method [49] in the crossover operator for the first time in such a stochastic problem, enabling us to observe its performance in this context.

# 2
# METHODOLOGY

This section will discuss the criteria on which the load profile generation process is based, the mathematical model of the system, how consumer habit parameters are obtained using a genetic algorithm, the data generation process using the obtained parameters, and the methods used for verifying the generated data.

## 2.1 Overview

The load profile generation process is essentially divided into two stages: The first stage involves determining the parameters that represent consumer habits, while the second stage involves generating load profiles using these parameters.

Fig.2.1 illustrates overall architecture of the load profile generation process. In the first stage, survey data containing usage habits related to the region for which load profiles are desired serves as the input to the system. At this point, the employed genetic algorithm estimates the appropriate parameters for predetermined consumer types while staying within the specified error range. The resulting parameters serve as an input to the algorithm operating in the second stage.

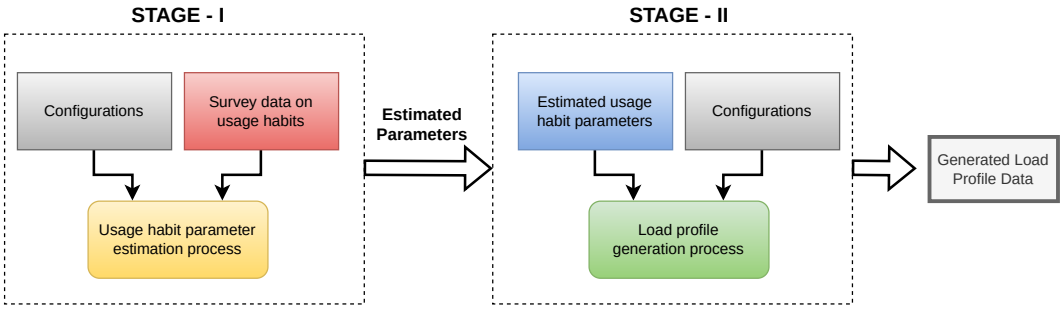



**Figure 2.1** Overall architecture

The mentioned general flow is applicable when we do not have parameters representing consumer habits. If the first stage has been previously executed and the necessary parameters have been generated, then using the same parameters,

different sets of load profiles can be generated solely through the second stage. This is one of the advantages of the presented study.

## 2.2 Data Quality Criteria

Ensuring the quality of the data used in algorithms developed for maintaining supply-demand balance, dynamic pricing, load shifting applications, or peak shaving operations in electric grids is of critical importance for the success of such studies. In this work, three criteria have been employed, as shown in Fig.2.2, to characterize the quality of the generated load profile data: Accuracy, randomness, and diversity. The remaining part of this section will discuss how these criteria are achieved.

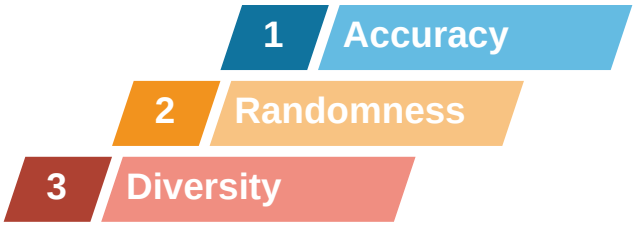



**Figure 2.2** The generated load profile quality criteria

**Accuracy**

The first characteristic to be sought in the generated load profile data should be accuracy. This is because if a load profile is generated for a specific region, the algorithms to be applied will be developed entirely based on this data. Therefore, any inaccuracy in the generated data would lead to the algorithms producing incorrect results and losing their connection to reality. In this study, we ensure the accuracy of the generated data statistically. During the stage of generating parameters representing usage habits, the specific genetic algorithm we have developed has a "fitness function" component that aims to approximate the average usage of each device in the generated data to the survey values provided as input to the algorithm. Thus, even though individual usages may vary, the overall usage consistently approaches the actual results. This study ensures accuracy in this way. The topic of verification will be discussed again in the subsequent sections under the heading of verification.

**Randomness**

The data generated as a result of this study will be created in the specified types, with the specified number of users, and each user having the specified devices. This stage

involves a significant amount of randomness in the generated data. For example, the answers to many questions, such as how many users will be in each type, which devices and electric vehicle models the users will have, and at what hours the users will use their devices, are all the same: "random". Our study offers two options for randomness: One is the pseudorandom number generator (PNRG) algorithm, which can be used in all computer environments and can actually exhibit repetition within certain limits. The other is the true random number generator (TRNG) algorithm, which is generated entirely based on physical observations and does not exhibit repetition no matter how many values are generated. In the implemented system, a third-party service was used to utilize the TRNG algorithm [50]. However, since this service is paid, it has been used only where its feasibility has been proven, and for the simulations presented in the subsequent sections, the PRNG has been preferred. As long as the number of users in the generated data does not reach excessive levels, no negative impact of the PRNG algorithm has been observed.

**Diversity**

The principle of diversity is closely related to the principle of randomness. This principle emphasizes that even if the same usage habit parameters are used and the simulation settings are kept exactly the same, each set of generated load profiles will be different from one another. The only common point among these generated sets is that they statistically yield the same results. We can refer to this situation as the characteristic model of the desired region's load profiles produced with survey data. While individual data may vary, the characteristic of the data set will remain the same as long as the survey data remains unchanged. This principle is of crucial importance, especially for the development of algorithms such as regional load shifting, dynamic pricing, or peak time estimation. By having different data sets with the same characteristic, the developed algorithm will converge towards success with each iteration and can be objectively tested.

## 2.3 Mathematical Background

In this study, the main point we considered while generating consumer load profiles is the compatibility of the generated data with reality. This compatibility was achieved using genetic algorithms. In order to incorporate genetic algorithms into our study, it was necessary to mathematically represent the defined consumer profiles. Only in this way could we utilize the operators of the genetic algorithm and achieve satisfactory results. The realism of the mathematical model is also of critical importance. In this section, we will step-by-step demonstrate the mathematical

background of the study, how we modeled user profiles, and at which point we integrated the genetic algorithm.

Firstly, we need to classify the consumers in the region where load profiles will be generated as $u = \{u_t | t \in \{1, \ldots, T\}\}$, where $T$ is any integer. Similarly, let's represent the appliances included in the study as $a = \{a_j | j \in \{1, \ldots, J\}\}$, where $J$ is any integer. In the surveys used in this study, the usage duration of each appliance was taken as the weekly hours per person, and based on this, the total weekly duration of a household was calculated. Therefore, the $TP(u_t)$ function represents the total number of individuals in a household of type $u_t$, the $SWO(a_j)$ function represents the weekly operation duration obtained from the survey for appliance $a_j$, and the $RWO(u_t, a_j)$ function indicates how many hours per week a household of type $u_t$ operates appliance $a_j$ in reality. Based on this, the following Eq.2.1 can be written:

$$RWO(u_t, a_j) = TP(u_t)SWO(a_j) \tag{2.1}$$

The goal in the generated load profile data is to converge to the actual value, namely the $RWO$ value for each $(u_t, a_j)$ combination. Therefore, we can express the cost function to be used for the genetic algorithm as Eq.2.2.

$$g(u_t, a_j) = \left| \frac{TWO(u_t, a_j) - RWO(u_t, a_j)}{RWO(u_t, a_j)} \times 100 \right| \tag{2.2}$$

The $TWO(u_t, a_j)$ function represents the weekly total operation duration obtained from the generated data for the given user type $u_t$ and appliance $a_j$. The convergence of the given cost function $g(u_t, a_j)$ to zero implies that the generated data approaches reality. This will also be the objective within the genetic algorithm.

As expressed in Section 2.2, the time at which assigned devices are operated by respective users is randomly determined. However, this randomness does not occur uniformly. The most critical point of the mathematical model presented in this study is the creation of the "random assignment function ($RAF$)". The value obtained from this function influences whether the corresponding device will be used by the respective user at that particular moment. While always maintaining the randomness factor, we eliminate the uniformity of this randomness in order to generate more realistic data. As stated in Eq.2.3, whether a device is used at a given time by the specific user-type is modeled mathematically.

$$RAF(u_t, a_j, i, k) = T_{p,pro}(u_t, k)K_{u_t,a_j} - D_{a_j}(k)K_{ud_t,a_j} - W_{a_j}(k)K_{uw_t,a_j} \quad (2.3)$$

The indices $i$ and $k$ correspond to the sectors within a day and week, respectively. The function $T_{p,pro}(u_t, k)$ provides the number of individuals present in the household within the given time sector. In other words, if we consider all consumers as residential, it gives the number of individuals present in the house at the specified time. The function $D_{a_j}(k)$ indicates how many times device $a_j$ has been used within the same day in the simulation until the specified time point. Similarly, the function $W_{a_j}(k)$ indicates how many times device $a_j$ has been used within the same week in the simulation up to the current point. The parameter $K_{u_t,a_j}$ represents the usage habit of user $u_t$ towards appliance $a_j$ and has a positive effect on the likelihood of the appliance being used at that moment. The parameter $K_{ud_t,a_j}$ is a parameter determined based on the number of times appliance $a_j$ is used within the same day and has a negative effect on the likelihood of the appliance being used at that moment. The parameter $K_{uw_t,a_j}$ is a parameter determined based on the number of times appliance $a_j$ is used within a week and also has a negative effect on the likelihood of the appliance being used at that moment. The summary of Eq.2.3 is as follows: The probability of a device being used at a selected time for a given type of household increases as the number of people present in the house at that moment increases, decreases as the number of times the device has been used within the same day increases, and decreases as the number of times the device has been used within the week increases. In this study, the likelihood of an appliance being used or not at any given moment is modeled in this way. This is one of the important contributions of our study.

In light of the above explanation, the remaining points in Eq.2.3 are the parameters $K_{u_t,a_j}$, $K_{ud_t,a_j}$, and $K_{uw_t,a_j}$. At this stage, our solution involves the integration of a genetic algorithm. If we consider these parameters as set $K_{u_t-a_j}$, a chromosome to be used in the genetic algorithm is formulated as in Eq.2.4:

$$K_{u_t-a_j} = [K_{u_t,a_j}, K_{ud_t,a_j}, K_{uw_t,a_j}] \quad (2.4)$$

Another issue in Eq.2.3 is the $D_{a_j}(k)$ and $W_{a_j}(k)$ functions. In this study, when assigning the operating hours of devices, we divided a day into four parts, each consisting of six-hour intervals. After determining whether an appliance will operate during a specific hour of the day, the question arose regarding how many

times we would perform a similar operation within the same day. At this point, we decided to divide the day into four parts and apply a probabilistic operation process for each appliance four times a day. These four parts were determined experimentally. Dividing the day into more parts increases the computational burden of the genetic algorithm and takes more time to reach a solution. The results of the simulations will be discussed in subsequent sections.

The $D_{a_j}(k)$ function is reset to zero for each device when a new day begins. Similarly, the $W_{a_j}(k)$ function is reset to zero for each device when a new week starts. These functions can be mathematically expressed as Eq.2.5 and Eq.2.6, respectively.

$$D_{a_j}(k) = \{\forall k \in \{1, 2, \ldots, 7\} : \sum_{i=1}^{4} h_{st,a_j}(i, k)\} \tag{2.5}$$

$$W_{a_j}(k) = \sum_{k=1}^{k_0} \sum_{i=1}^{4} h_{st,a_j}(i, k) \tag{2.6}$$

where $h_{st,a_j}(i, k)$ is the starting-time function of the appliances at the corresponding day and $k_0$ is the current day in the simulation.

If the corresponding appliance is operational during the given time intervals $h_{st,a_j}(i, k) = 1$; otherwise, $h_{st,a_j}(i, k) = 0$.

In summary, as the value of the $RAF$ function increases, the probability of the device being operational at that moment also increases. Due to the difficulty of predicting the range of the resulting value, we attempted to control the output of the function within a certain range by manipulating the values of the $K_{u_t - a_j}$ set given in Eq.2.4. After calculating the $RAF$ function, a random number is selected from the interval $[0, 100]$. If the selected value is less than or equal to the result of the $RAF$ function, the device is operated; otherwise, it is not. In this way, the genetic algorithm calculates the $K_{u_t - a_j}$ parameters in a way that keeps the result of the $RAF$ function within the range of $[0, 100]$. The presented idea is one of the important breakthroughs of this study.

## 2.4 Parameter Estimation with Genetic Algorithm

Genetic algorithm is an intuitive optimization and search algorithm inspired by natural selection. It is one of the subcategories of evolutionary algorithms. It

incorporates operators with the same names, such as crossover, mutation, and selection, inspired by certain biological processes. Based on the algorithms, many variations of genetic algorithms have been developed. However, they all converge at the same fundamental operating principle. Fig.2.3 illustrates some variations of genetic algorithm operators [51].

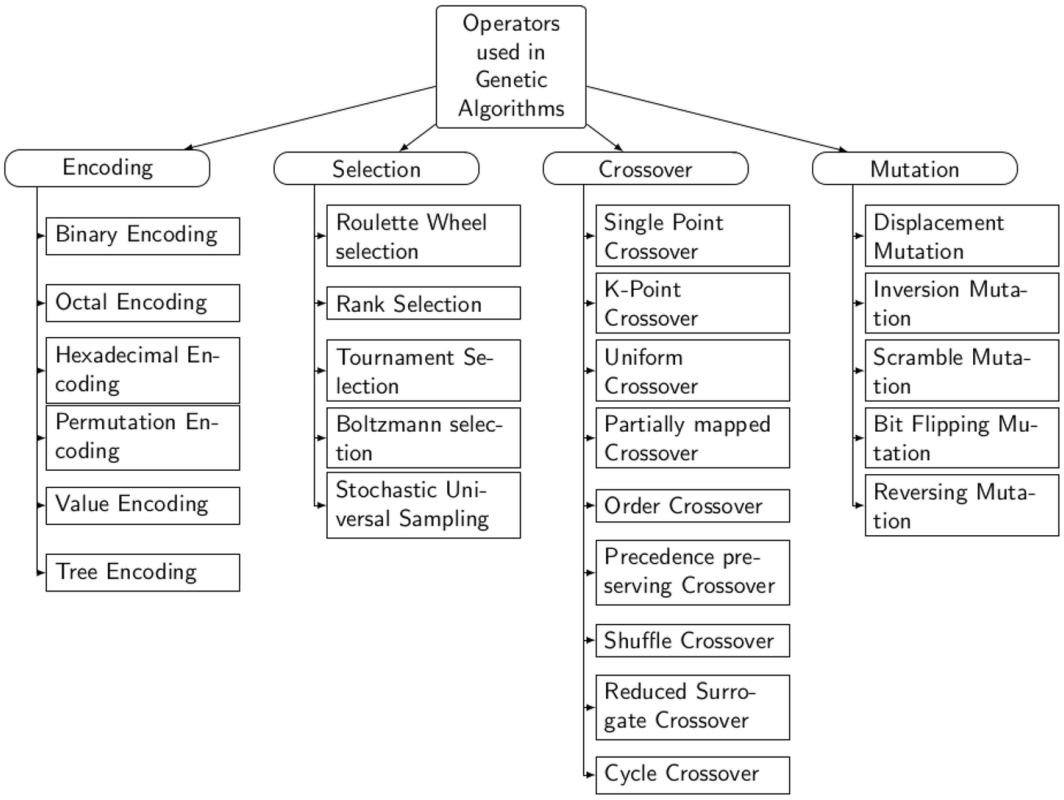



**Figure 2.3** Generic algorithm operator types

The genetic algorithm to be applied should be carefully chosen according to the specific problem at hand. Otherwise, issues may arise such as the algorithm getting stuck at local minima or local maxima points, or continuously searching for the optimal point without focusing on a particular point.

In the load profile generation problem, we knew that we couldn't benefit from existing variations of genetic algorithms for estimating the parameters representing usage habits. This is because our specific problem did not conform to the patterns found in the literature. Before discussing the features of the genetic algorithm developed within the scope of this study, it is worth addressing the problem at hand.

The main difference of the usage habit parameter estimation problem from traditional optimization problems is its stochastic nature. In other words, the parameters defined in Eq.2.3, as described in Section 2.3, do not yield the same result in every usage. The characteristic of the problem being stochastic necessitates the solution to be in the form of a range rather than a single point. In this study, an

error margin of $15\%$ is set. This means that the convergence rate of the generated load profile data with the survey data should be at least $85\%$ in terms of proximity. The simulation parameters in this study will be discussed in subsequent sections, but the point being made here is that we do not have a deterministic problem. As a result, the genetic algorithms found in the literature do not fully adapt to this problem.

Fig.2.4 illustrates the basic flow of the genetic algorithm. The genetic algorithm presented in this study operates on the same principle. The difference lies in the algorithms contained within the operators shown in the figure. In the following subsections, we will discuss the genetic algorithm operators that we developed to find the usage habit parameters.

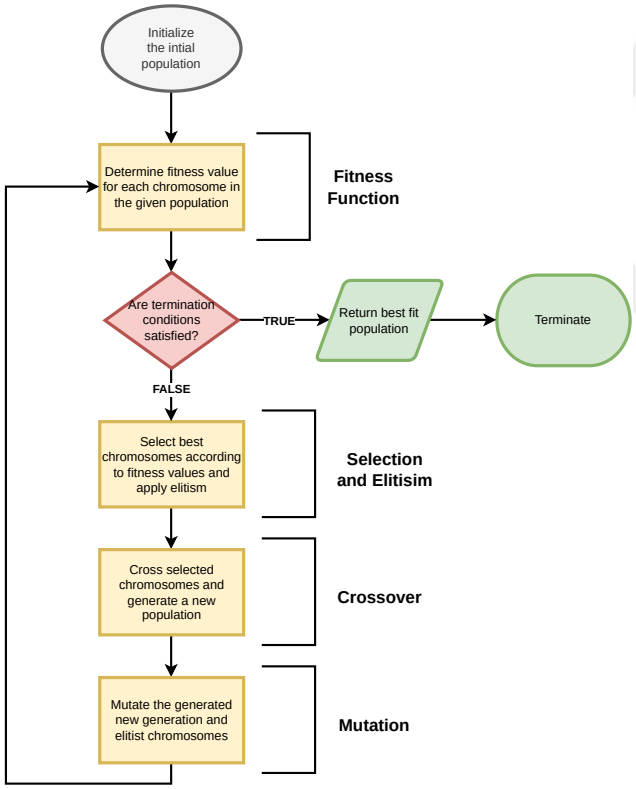



**Figure 2.4** Fundamental principle of generic algorithm

### Fitness Function

The fitness function operator is a function where sets of parameters, called chromosomes, are tested and scored based on how close they are to the desired outcome. In this way, it performs the same task as the "cost function" in optimization problems.

In this study, the fitness function specifically designed for the problem tests the

parameter set $K_{u_t-a_j}$ given in Eq.2.4. Within the genetic algorithm, each element of the $K_{u_t-a_j}$ set is referred to as a "gene," while the set itself is referred to as a "chromosome".

Within the fitness function, a part of Stage-II shown in Fig.2.1 is executed. The function receives the user type parameter and the device assignment parameter as inputs. Then, a weekly usage data is generated for the assigned device based on the chromosome (usage habits parameter) to be tested. The resulting data is evaluated using Eq.2.2. The success value of the given chromosome is determined based on this evaluation. Subsequently, the algorithm proceeds to the next stages.

**Selection and Elitism**

The selection process is of critical importance for the successful progression of the genetic algorithm. Chromosomes that are evaluated and receive specific scores within the fitness function undergo a selection process at this stage. There are two important considerations at this stage. The first is to ensure that chromosomes with higher scores are not lost and allow them to dominate in the subsequent generation. The other important consideration is the possibility that these high-scoring chromosomes may represent local optima, and therefore, it is necessary to carry over chromosomes with lower scores to the next generation to explore alternative solutions. In addition to the selection process, there is another operation that serves as insurance during the genetic algorithm process: Elitism. This operation aims to transfer a small number of the most successful chromosomes to the next generation without subjecting them to any selection process, as a safeguard. In this study, we also applied elitism. Since the focus of this study is not solely on the genetic algorithm, a comparison between the application and non-application of elitism was not made.

To perform the selection process in an objective manner, maintain randomness in the genetic algorithm to prevent it from getting stuck at a particular point, and at the same time increase the probability of selecting more successful chromosomes, the cumulative probability distribution function (CPDF) has been used. However, the CPDF in its pure form posed certain challenges. We will examine these challenges and how we addressed them.

Let $F = \{f_1, \ldots, f_n\}$ represent the set of fitness values for the current generation. In this form, the selection probabilities $P$ should be as given in Eq.2.7:

$$P = \left\{ p_i \middle| p_i = \frac{f_i}{\sum_{k=1}^{n} f_k} \right\}, i = 1, \ldots, n \tag{2.7}$$

With $S = \sum_{k=1}^{n} f_k$, as stated in Eq.2.8, the sum of the elements in the P set will be equal to 1:

$$\sum_{i=1}^{n} p_i = \frac{f_1}{S} + \frac{f_2}{S} + \cdots + \frac{f_n}{S} = 1 \tag{2.8}$$

A significant problem arises when considering Eq.2.7. If $f_i = 0$ for any $i$, then $p_i$ will also be zero. However, in the context of the prepared fitness function in Eq.2.2, the case of $f_i = 0$ represents the most successful chromosome. Consequently, the most successful chromosome would never be selected in this way. If we take the reciprocal of the obtained $p_i$ values according to the multiplication operation, we encounter divided-by-zero situations, which prevents us from obtaining any probabilistic expressions in this manner.

To solve this problem, the following method has been applied. Let's consider Eq.2.9 with $A > 0$ and $A + n \neq 0$:

$$\frac{A}{A} = \frac{A+n}{A+n} = 1 \tag{2.9}$$

The numerator and denominator values in Eq.2.9 have changed, but their ratios remain the same. Thus, even in the case where $A = 0$, a probabilistic value is obtained. It should be noted that we utilized the equality $A = A$ here. In the case of the contrary, this operation would alter the proportionality.

Therefore, based on this, we can rewrite Eq.2.8 as follows:

$$\sum_{i=1}^{n} p_i = \frac{S}{S} = \frac{S+n}{S+n} = \frac{f_1+1}{S+n} + \cdots + \frac{f_n+1}{S+n} = 1 \tag{2.10}$$

Starting from Eq.2.10, $P'$ can be expressed as Eq.2.11:

$$P' = \left\{ p_i{}' \middle| p_i{}' = \frac{f_i+1}{S+n} \right\}, i = 1, \ldots, n \tag{2.11}$$

When the elements of $P$ and $P'$ sets are compared, it can be observed that the

values are different from each other. However, this is a negligible difference. As the value added to each element increases, especially in the case of fitness values close to zero, the negative difference also increases. Experimentally, adding 1 to each numerator element was found to be sufficient. Therefore, the loss incurred can be considered negligible in terms of resolving the problem described above.

After organizing the fitness values as shown in Eq.2.11, the reciprocal of each element is taken with respect to the multiplication operation and divided by the total values so that both elements close to zero have a higher probability range and the normalization of values is achieved.

From this point on, the fitness values set $P'$ can undergo the CPDF operation and as a result, the table shown in Fig.2.5 is obtained.

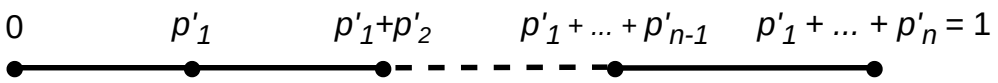



**Figure 2.5** CPDF applied $P'$ set

After applying the CPDF operation, as indicated in Fig.2.5, a region is formed for each chromosome's fitness value. Subsequently, a random real number is selected from the interval $[0, 1]$. The selected number determines which region it falls into, and the chromosome associated with that region is chosen. Thus, while preserving randomness, a higher likelihood of selection is given to more successful chromosomes. This contribution of the presented study to the literature lies in the genetic algorithm selection operator.

**Crossover**

The crossover operator is where the genetic algorithm demonstrates its prowess. With this operator, the genetic algorithm moves towards optimal values in a vast search space with each new generation.

The method for performing the crossover operation depends directly on the problem at hand. Fig.2.3 presents some of the crossover methods found in the literature. While some of these crossover methods can operate on real numbers and integers, others may work exclusively with binary representation. Additionally, each method may have a different spread radius.

In this study, we relied on the article by Stjepan Picek, Domagoj Jakobovic, and Marin Golub to determine the crossover method to be used [49]. In their

paper, they combined known crossover methods for "real-coded" genetic algorithm problems, and achieved more successful results compared to individual methods that could be used separately. Based on this, we combined 12 different crossover methods in a randomized selection manner for each crossover operation, and obtained our results accordingly. Some of the crossover methods used are as follows: discrete crossover [52], simple arithmetic crossover [52], single arithmetic crossover [52], whole arithmetic crossover [52], local crossover[53], SBX crossover [54], BLX-alpha crossover [55], heuristic crossover[56], BLX-alpha-beta crossover [57], and one-point crossover [58].

**Mutation**

The last operator of the genetic algorithm is the mutation operator. The mutation operator performs a function equivalent to its biological counterpart. The selected chromosomes have undergone the crossover operator, resulting in new chromosomes. Now we have the chromosomes generated through crossover and the chromosomes selected through elitism during the selection process. The power of a genetic algorithm lies in its nature of searching unpredictable parameter spaces. The mutation operator plays a critical role in this regard. If there are still chromosomes that remain stuck at local optimal points after selection and crossover, some of them are randomly altered within a predetermined range of changes. This may disrupt the optimal point, but the genetic algorithm will address such issues in the next generation. We can liken the genetic algorithm to a soldier shooting continuously into a specific area. Sooner or later, the soldier will "accidentally" hit the target. While the working principle of the genetic algorithm is similar to this analogy, it progresses in a more planned manner than chance.

In this study, a dynamically changing mutation rate was applied to the chromosomes. This approach both limits the number of chromosomes undergoing mutation and allows for the possibility of chromosomes not undergoing mutation at all. As a result, the mutation operator exhibits greater similarity to its biological counterpart.

In this study, we have proposed hetero-probabilistic mutation operator by presenting dynamical influencing factor on the repeating cases of the mutation. The operator readapts the weight of the parameters based on the mutation incidence frequency. For this aim, we define the variables of the mutation probability of each chromosome ($P_{M,C}(\lambda)$) and each gene ($P_{M,G}(\lambda)$) where $\lambda$ is local variable keeps the iteration number. Also, we define the parameters of $MR_C$ and $MR_G$ which are the mutation rate to each chromosome and gene, respectively. The

chromosomes are sorted randomly and a random number $\alpha \in [0, 100]$ is assigned to the each chromosome. The chromosomes in the list are compared with $P_{M,C}(\lambda)$, respectively. If $\alpha < P_{M,C}(\lambda)$ satisfies, then the corresponding chromosome is selected for mutation. Immediately, the genes are sorted again randomly and a random number $\beta \in [0, 100]$ is assigned to the each genes. For each genes, the comparison of $\beta < P_{M,G}(\lambda)$ is tested and whenever it is satisfied, the regarding gene value ($K_G(\lambda)$) is modified as follows

$$K_G(\lambda + 1) = K_G(\lambda) + U(\gamma) \times K_G(\lambda) \times (MR_G/100) \tag{2.12}$$

where $U(\gamma)$ is a random sign function that is $+1$ whenever the random binary digit $\gamma = 0$, $-1$, otherwise. Then, the $P_{M,G}(\lambda)$ and $P_{M,C}(\lambda)$ are updated as

$$P_{M,G}(\lambda + 1) = P_{M,G}(\lambda) - \zeta_1 \times MR_G \tag{2.13}$$

$$P_{M,C}(\lambda + 1) = P_{M,C}(\lambda) - \zeta_2 \times MR_C \tag{2.14}$$

where $\zeta_1$ and $\zeta_2$ are the counter of mutant gene and chromosome numbers, respectively.

## 2.5 Data Generation

The generation of load profile data is the second stage of the entire process, as shown in Fig.2.1. During data generation, the first stage is to randomly create the desired number of users based on the configured user types. The specified devices are also randomly assigned to these created users according to the configuration stage. Whether or not electric vehicle allocation is performed is determined in the configuration where user types are specified. As mentioned in Sec.2.2, TRNG can be used during both user type assignment and device allocation. Our study makes certain assumptions. The purpose of these assumptions is to reduce the complexity of the system by sacrificing a negligible amount of data. For example, each device can only be assigned to a user once. Additionally, since survey data is person-based, when calculating the consumption of a household, we assume that each person contributes equally to certain appliances. For shared appliances, we assume an approximate contribution and calculate the total consumption. Moreover, sleep hours during the night or working hours of employed household members are approximately given. These values are based on our personal observations. Although we have such assumptions, the aim of our study is not to accurately generate load profiles of a specific region; rather, the main objective is to generate accurate data based on the provided survey data and entered initial parameters.

Therefore, the accuracy of the input data is entirely the responsibility of the user.

After the assignment of users and their corresponding appliances is completed, the operating times of these assigned appliances are determined. For this purpose, as mentioned in Sec.2.3, each day is divided into four equal sectors, and the working times of the assigned appliances in each sector are randomly determined based on the probabilities obtained from Eq.2.3, which includes the habits of whether all assigned appliances are being used or not. At this point, it is worth mentioning the types of appliances used in our study. We divided the appliance types into three categories:

1. *Non-periodic appliances:* For these appliances, once the start time is determined, the end time is also known since their operating durations are fixed. Examples of such appliances include washing machines, dryers, water heaters, etc.

2. *Periodic and non-continuous appliances:* For these appliances, once the start times are determined, their operating durations are randomly assigned within the limits specified in the configuration. Examples of such appliances include TVs, PCs, etc.

3. *Periodic and continuous appliances:* For these types of appliances, specific start and end times are not determined. Instead, they may have options such as periodic operation with intervals of work and rest or continuous operation without stopping. Examples of such appliances include refrigerators, air conditioners, modems, etc.

Determining the operating times of assigned devices is one of the most complex parts of the algorithm. Unless otherwise restricted, devices can operate at any time during the day. There may also be constraints such as ensuring that different devices do not operate simultaneously. Such flexibilities help enhance the realism of the data. Once the operating hours are assigned, the energy consumed by the devices is allocated to the corresponding operating hours. There are also options available for determining the energy values. After the energy allocations are completed, the usage of all devices is aggregated to form the load profile of the corresponding user for the desired period.

## 2.6 Verification

One crucial aspect, as important as load profile generation, is the validation of the generated data. The fundamental aim of our study is to produce individual load profile data that is consistent with real statistical data. The validation of this process is performed within the established architecture during the course of the study. To validate the data, the parameters representing user habits need to be confirmed. Additionally, the data generated using these parameters and Eq.2.3 should also be validated. In other words, both the parameters and the Eq.2.3 presented should be part of the validation process.

As described in Section 2.4, the genetic algorithm, specifically tailored for our problem, predicts the consumption habits represented by parameters. In doing so, at each new parameter generation, a one-week data is generated for the corresponding household and appliance using these parameters. After the data is generated, Eq.2.2 is utilized to calculate the fitness value of the parameters, in other words, to determine how close the generated data is to reality. If this value falls below the pre-defined 85% compatibility threshold, the process continues with another set of parameters. Once the desired success rate is achieved, the genetic algorithm strives to approach the 100% full compatibility figure. This process continues until it reaches the generation limit, which is determined experimentally.

When considering the process of predicting usage patterns and generating load profiles using these parameters, the data verification process is performed within the process itself. Of course, data verification could have been performed after all the load profiles were generated. However, such a scenario would impose a significant computational burden, and in return, there would not be a substantial difference.

Another factor that affects the accuracy of the data is how many sectors a day is divided into. This issue was discussed in detail in Section 2.3. In our simulation study presented in this work, we divided a day into 4 sectors. However, by increasing this number to, for example, 8 sectors, we may achieve slightly better results. Nevertheless, this will also lead to a higher computational load. Many of the parameters in our simulation study are experimentally optimized.

# 3
# IMPLEMENTATION

## 3.1 Development and Simulation Environment

The simulation of the proposed load profile generation is implemented in MATLAB® using an Intel® Core™ i3-3110M processor. The implementation process was carried out using MATLAB® core functions, and no additional toolboxes were included. From this perspective, our work is platform-independent. The same study can be implemented using any open-source structured programming language. Functional programming technique was used instead of object-oriented technique. This allows the impact of each function on the data to be easily observed. The structure format was frequently used to simplify the representation of the data.

## 3.2 Configuration

The configuration phase of the developed software is of critical importance for the accuracy and reliability of the data to be obtained at the end. The configuration is entirely done using text-based JavaScript Object Notation (JSON) files. In other words, the configuration process can be carried out using a simple text editor. Table 3.1 presents the configuration files used in our study and their respective responsibilities.

Configuration process shall be done with consideration of the region which is target of the survey data. It is not enough only acquire usage data for each appliances. Besides, it is necessary to know basic types of household profiles and appliance types which they own. The user types to be created during the configuration phase are of critical importance for the reliability of the generated load profile data. However, the aim of our study is to produce load profiles that are compatible with the given configuration and survey data. Therefore, the accuracy of the input data is the responsibility of the user.

The flexibility achieved by the presented study to cover various scenarios has led to

**Table 3.1** Contents of the configuration files

| Configuration File Name | Content |
|---|---|
| Appliances Data | All configuration related to household appliances is kept in this file. Information such as the types of appliances, their energy consumption, time restrictions, real usage data, etc., are included in this configuration file. |
| Residental Types | The configuration parameters for the specified household profiles are included in this file. Parameters such as household population, number of employed individuals, working days, working hours, sleep hours, electric vehicle ownership status, appliances that need to be assigned and not assigned, are set through this file. There is no restriction on the number of household profiles. |
| Electric Vehicles | Unlike household appliances, parameters related to electric vehicles are set through this file. It includes the names of vehicle models, battery capacities, pre-defined battery charge levels for software usage, and charging time restrictions. |
| Genetic Algorithm | All configuration parameters related to the genetic algorithm used to predict usage parameters are included here. The genetic algorithm also utilizes configuration files such as appliances data or residential types. With this configuration file, various parameters can be adjusted, such as the number of chromosomes in each population, the number of genes in each chromosome, information used in the selection operator, mutation parameters, and termination conditions. These settings directly affect the performance of the genetic algorithm. |
| Initial Conditions | This configuration file, used by both the genetic algorithm and the data generation algorithm, contains fundamental parameters such as sample time, the number of households for which load profiles will be generated, the number of weeks for which load profiles will be generated, and the division of the day into sectors. |
| Run Probability Parameters | This configuration file contains parameters that are not manually created by the user but are automatically generated through the genetic algorithm process. The parameters representing consumption habits are included in this file. |

the complexity of the configuration files. Especially in the configuration

**Listing 3.1** Instance of an appliance configuration

```
"washingMachine": {
    "type": 0,
    "needOperator": 1,
    "power": {
      "format": "choice",
      "value": [
        600,
        900,
        1200]},
    "operation": {
      "mode": "non-periodic",
      "continuity": 0,
      "runDuration": [1, 20]},
    "weeklyRunInReal": {
      "case": 1,
      "personCount": 1,
      "value": [0,213],
      "additionEachPerson": [0, 25]},
    "dependency": {
      "case": 0},
    "constraints": {
      "workTimeConstraint": {
        "case": 0},
      "applianceConflictionConstraint": {
        "case": 0},
      "evConflictionConstraint": {
        "case": 0}}}
```

files where household appliances and user types are entered, there are many options. Listing 3.1 shows the configuration for the washing machine. If we examine the parameters, the "type" parameter determines whether the entered device is a consumer or a producer.

The "needOperator" parameter indicates whether the household appliance requires an operator for operation. The "power" parameter specifies the power consumption of the household appliance. The variables can be written in two different formats: When the "choice" format is used, a random selection is made among the given

values to determine the power consumption of the household appliance when it is used. The other format, "interval", allows the power consumption value to be randomly selected from a range of two given values. These two formats provide significant flexibility in determining the power consumption of household appliances.

The "operation" parameter determines the mode of operation of the household appliance, which was discussed in Section 2.5. The "runDuration" parameter determines the operating duration of the household appliance, and of course, this duration is interpreted differently depending on the operating mode. The time format used throughout the configuration is "[hour, minute]".

The "weeklyRunInReal" parameter represents the interpreted form of the relevant survey data for the respective household appliance. The "case" parameter, present here and in subsequent sections, serves as a flag to indicate whether the relevant configuration is present. This facilitates querying within the software.

The "dependency" parameter is used during the assignment of household appliances. To assign the relevant household appliance, the household appliance specified by this parameter must also be assigned to the respective user.

The "constraints" parameter contains various restrictions. The "workTimeConstraint" parameter, as the name implies, defines constraints on the operating duration of the household appliance. The "applianceConflictionConstraint" parameter lists household appliances that cannot operate together simultaneously, and the "evConflictionConstraint" parameter indicates that the relevant household appliance cannot operate together with the specified electric vehicle.

As mentioned above, these parameters add significant flexibility to the study. However, as a negative effect, they introduce a degree of complexity during the configuration stage.

Listing 3.2 shows an example of the configuration for household types. In this example, parameters specific to Type-4 residential type are presented. Similar to the configuration for household appliances, there are many flexibilities in this configuration as well.

The "employedCount" parameter indicates the number of employed individuals within the residence, while the "nonemployedCount" parameter specifies the number of non-employed individuals. Thus, the total number of residents in the

house is obtained by adding these two parameters.

**Listing 3.2** Instance of a household type

```
"type4": {
    "employedCount": 2,
    "nonemployedCount": 2,
    "alwaysAppliances": [
      "washingMachine",
      "dishwashing",
      "lamp",
      "tv",
      "fridge"],
    "neverAppliances": [
      "windTurbine",
      "solarPanel"],
    "haveEV": 2,
    "jobSchedule": {
      "case": 1,
      "workDays": [
        1,
        2,
        3,
        4,
        5],
      "lowerTime": [8,0],
      "upperTime": [19,0]},
    "constraints": {
      "constraintsCount": 0},
    "sleepTime": {
      "case": 1,
      "lowerTime": [23,0],
      "upperTime": [7,0]}}
```

The "alwaysAppliances" parameter lists household appliances that must be assigned to the respective residential type without exception. Similarly, the "neverAppliances" parameter specifies household appliances that must not be assigned to the respective residential type. These two parameters play a significant role in the assignment of household appliances.

The "haveEV" parameter indicates whether the respective user type will be assigned

an electric vehicle or not. There are three options here: situations where electrical vehicle (EV) assignment is absolutely not made, situations where EV assignment is absolutely made, and situations where EV assignment is made randomly.

The "jobSchedule" parameter specifies the workdays and working hours for the respective residential type. The "lowerTime" parameter indicates the start time of the working hours, while the "upperTime" parameter shows the end time.

The "constraints" parameter contains restrictions on the use of certain items. However, such a restriction is not present in the current residential type.

The "sleepTime" parameter indicates the sleep habits of the respective residential type. For example, in this example, we can say that residents of this type sleep between 23:00 and 07:00. Therefore, during these hours, no household appliance requiring an operator should operate. However, there are no such restrictions for household appliances that do not require an operator.

Other configuration files also have a similar structure. All stakeholders of the generated data have different flexibilities in this way. The configurations given as examples above were used in our simulation study.

## 3.3 Coding

The code implementation of the presented study is fundamentally divided into three parts: configuration, genetic algorithm, and data generation. Additionally, there is a section dedicated to plotting to analyze the generated data. The implementation process is entirely completed through multiple functions communicating via an interface and main script files that call these functions. In this section, pseudocodes of some algorithms involved in the data generation process will be discussed.

The general flow of the process is provided in Fig.3.1 with all its aspects. This flow is a more detailed version of the previously presented Fig.2.1. The expressions related to the methodology described in Chapter 2 are also included in Fig.3.1.

If we start the data generation process by predicting the parameters representing usage habits using the genetic algorithm, Algorithm 5 should be expressed first. This algorithm represents the general process of the designed genetic algorithm. We can see the sequence of operations and the tasks performed by these operators with this algorithm. The given algorithm is a detailed version of the genetic algorithm flowchart provided in Section 2.4.

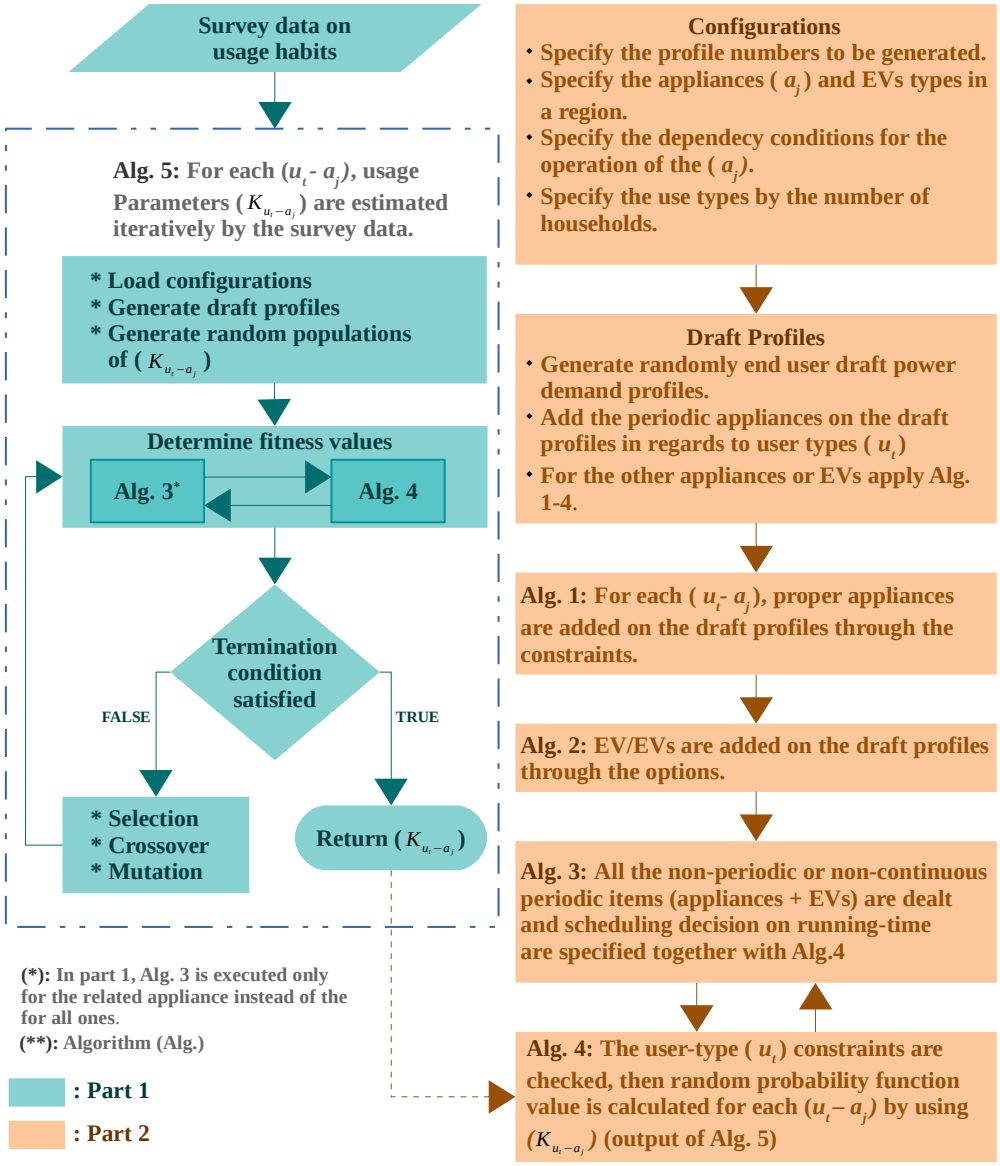



**Figure 3.1** Load profile generation process

**Algorithm 1:** Inserting appliances to the end-user profiles

```
foreach endUser(u_t) do
    foreach appliance(a_j) do
        /* Pick an appliance randomly and evaluate the options of
           exact, never and inexact for the assignment with
           constraints:                                          */
        if a_j in available list then
            assign a_j to endUser profile;
        else if a_j in unavaliable list then
            continue;
        else
            /* Define binary-digit variable(randomBit) and assign a
               random value.                                     */
            randomBit = rand([0,1]);
            if randomBit == 1 then
                /* Check the correlation constraint.            */
                if correlations satisfied then
                    /* Conditions are satisfied, hence assign the new
                       correlated appliance.                     */
                    assign a_j to endUser;
                else
                    continue;
                end if
            else
                continue;
            end if
        end if
    end foreach
end foreach
```

Within Algorithm 5, two different algorithms are used, which are also employed during the data generation process. One of these is Algorithm 3, which determines the operation times of the assigned appliances to households and is relatively more complex. This algorithm applies a working test for all assigned appliances in all sectors within a day. During this test, it calculates the probabilities of appliance operation. The calculated value is then compared with a randomly selected value, and accordingly, the appliance's operation time is assigned.

Another process within the scope is represented by Algorithm 4. The main purpose of Algorithm 4 is to calculate the probability of the appliance's operation as given in Equation 2.3. As mentioned earlier, this algorithm is used both in the genetic algorithm process and in the data generation process. It is worth mentioning here that the verification method explained in Section 2.6 emphasizes this point. The accuracy of the generated parameters is reevaluated multiple times during this process, so the presented study performs its own validation internally. Therefore, there is no need for an additional verification method.

After the completion of the genetic algorithm process, the data generation process

**Algorithm 2:** Inserting the electric vehicles profiles.

```
foreach endUser(u_t) do
    /* Get the list of electric vehicles.                          */
    electricVehicleList = getListEVs();
    /* Pick randomly a draft profile generated by Algorithm 1 and
       evaluate the options of none(0), inexact(1) and exact(2)
       for the assignment of the EV to the end-user type.          */
    switch option do
        case 0 do
            continue;
        case 1 do
            /* Define binary-digit variable(randomBit) and assign a
               random value.                                       */
            randomBit =rand([0,1]);
            if randomBit == 1 then
                /* Select randomly an EV from the list and add to the
                   profile.                                        */
                selectedEV = randomSelect(electricVehicleList);
                assign selectedEV to endUser profile;
            else
                continue;
            end if
        case 2 do
            /* Select an EV and assign to the profile.             */
            selectedEV = randomSelect(electricVehicleList);
            assign selectedEV to endUser;
    end switch
end foreach
```

begins as shown in Fig.3.1. At this point, the first task is to create the desired number of household examples for each designated household profile type. Subsequently, following the established configuration, the allocation of appliances to these generated households is performed. Algorithm 1 is designed to carry out this process. Except for the specifications explicitly defined in the configuration, all other allocations are entirely random. One of the significant achievements of our study is that despite the distinct characteristics of all these appliances and entirely random allocations, the system converges close to the given survey data by only manipulating the usage times.

Algorithm 2 demonstrates the allocation of electric vehicles. The process here is carried out in accordance with the established configuration. The options for allocating electric vehicles are explained in Section 4.2.

As mentioned above, Algorithm 3 and Algorithm 4 are parts of both the genetic algorithm process and the data generation process. Therefore, these algorithms are executed again during the data generation stage.

To summarize once again, while predicting the parameters representing usage

**Algorithm 3:** Providing probabilistic scheduling decision on the running-time of the each "non-periodic" and "non-continuous periodic" items (appliances and electric vehicles)

```
/* This algorithm is valid for only "non-periodic" and
   "non-continuous periodic" items.  Alg.3 uses the draft
   profiles generated by Alg.1&2.                          */
foreach endUser(u_t) do
    /* Randomly assign days of week to 'dayList'.          */
    dayList = randomSortDays();
    foreach day in dayList do
        /* Assign randomly the items to 'itemsList'.       */
        itemsList = randomSortItems();
        foreach items in itemsList do
            /* Divide randomly a day into parts.           */
            dayParts = randomSortParts();
            foreach part in dayParts do
                /* Select a random start time.             */
                startSample = randomSelect from part;
                /* Check constraints and specify random assignment
                   function(RAF) value for the sample time determining by
                   Alg.4.                                  */
                runProbability = Algorithm4(startTime);
                /* Select random number between (0,100).   */
                randomNumber = rand([0,100]);
                /* The higher the RAF, the greater the chance of being
                   activated for the item.  Compare 'randomNumber' and
                   'runProbability'.                       */
                if runProbability ≥ randomNumber then
                    /* Check the constraints and decide to increase or not
                       D_{a_j}(k) and W_{a_j}(k).          */
                    if conflictionConstraintsSatisfied then
                        assignWorktime();
                        D_{a_j}(k) = D_{a_j}(k) + 1;   W_{a_j}(k) = W_{a_j}(k) + 1;
                    else
                        continue;
                    end if
                else
                    continue;
                end if
            end foreach
        end foreach
        /* Reset daily usage counter.                      */
        D_{a_j}(k) = 0;
    end foreach
    /* Reset weekly usage counter.                         */
    W_{a_j}(k) = 0;
end foreach
```

habits, a process very similar to the data generation process is applied. The compatibility of the generated data with the survey data is checked. If there is not enough compatibility, the same process is repeated with another set of parameters. Therefore, there are many algorithms shared between these two stages.

**Algorithm 4:** Check the constraints and calculate the random assignment function(RAF) value for each item just scheduling in Algorithm 3

```
/* For each items, random run-time is specified by Algorithm 3,
   then Algorithm 4 is called.  Firstly, the user-type
   constraints are checked for any contemporisation with sleep
   time, operator need etc.  Then, the RAF value of u_t − a_j is
   assigned.                                                    */
condition-1 = (0/1);/* Operator need?                          */
condition-2 = (0/1);/* Job schedule of End-user?               */
condition-3 = (0/1);/* Run-time in job schedule?               */
/* Check the conditions for run-time assignment.               */
if runtime in sleepTime then
    if condition-1 then
        RPF = 0;
        return;
    end if
end if
if condition-2 then
    condition-3 = (0/1);
else
    condition-3 = 0;
end if
/* Get parameters generated by Algorithm 5 (GA).              */
get (K_{u_t−a_j} = [K_{u_t,a_j}, K_{ud_t,a_j}, K_{uw_t,a_j}]);
/* By the conditions, specify T_{p,pro}(u_t, k) and calculate the RAF
   value with Equ.(3).                                        */
if condition1 AND condition2 AND condition3 then
    T_{p,pro}(u_t, k) = (NonEmployed)Persons;
else
    T_{p,pro}(u_t, k) = (Employed+NonEmployed)Persons;
    /* Calculate Equ.(3):                                     */
    RAF(u_t, a_j, k) = T_{p,pro}(u_t, k) × K_{u_t,a_j} − D_{a_j}(k) × K_{ud_t,aj}  −W_{a_j}(k)  ×K_{uw_t,aj};
end if
```

**Algorithm 5:** A novel GA to predict usage habit parameters

```
/* Regard the statistical data for usage habits                    */
foreach userType(u_t) do
    foreach appliances(a_j) do
        /* Produce randomly the initial population                 */
        population(K_{u_t,a_j}, K_{ud_t,a_j}, K_{uw_t,a_j}) =
          generateRandomPopulation(TRNG);
        /* Calculate the fitness values                            */
        fitnessFunction:
        /* Assign weekly usage for each of u_t-a_j:                */
        call Algorithms 3&4;
        assignWeeklyUsage(u_t - a_j));
        calculate(TWO(a_t, aj), RWO(a_t, aj);
        f(t, j) = (TWO(u_t,aj) - RWO(u_t,aj)) / RWO(u_t,aj)) × 100;
        while f(t, j) does not satisfy termination conditions do
            Selection: Customized Roulette-Wheel;
            compute(CDPF); select parents;
            Elitism: Specify and Seperate the chromosomes;
            Crossover: Recombination operator;
            while offspring count ≠ specified number of chromosome do
                offspring = crossover(parents);
                offspring add to offsprings;
            end while
            Mutation: Particular scattering operator
            /* Get the mutation parameters P_{M,C}(λ), P_{M,G}(λ), MR_C, MR_G.
               */
            /* Define mutant gene/chromosome counter, local variable */
            ζ1 = 0, ζ2 = 0; λ = 0;
            foreach chromosome do
                P_{M,C}(λ + 1) = P_{M,C}(λ) − ζ2 × MR_C; α = rand(0, 100);
                if α ≤ P_{M,C}(λ + 1) ⇒ ζ2 ++ then
                    foreach gene in the chromosome do
                        P_{M,G}(λ + 1) = P_{M,G}(λ) − ζ1 × MR_G; β = rand(0, 100);
                        if β ≤ P_{M,G}(λ + 1) ⇒ ζ1 ++ then
                            K_G(λ + 1) = S(λ) × K_G(λ) + MR_G(λ + 1); /* Mutation occurs
                               */
                        end if
                    end foreach
                end if
            end foreach
            /* Assign the offsprings as new population:            */
            offsprings ⇒ population
            /* Recompute the cost function for the new generation  */
            fitnessFunction /* Given in the lines 6-13.            */
        end while
    end foreach
end foreach
```

# 4
# SIMULATION STUDIES

## 4.1 Scope and Objective

After describing the methodology in Chapter 2 and the implementation in Chapter 3, Chapter 4 focuses on our simulation study. The scope of the simulation includes only residential consumers. In these households, besides household appliances, electric vehicles can also be used. Although the software prepared for this study technically includes "producers" as stakeholders, the simulation study only involves "consumer" type parties.

From the beginning of our study, our goal, which we believe we have achieved, is to generate individual load profile data that is at least 85% compatible with the given statistical electricity consumption survey data for a specific region. This is the purpose of our simulation study. The adapted form of the survey data we used for our study can be seen in Table 4.2.

Within the scope of our simulation study, 5 household profiles have been identified, and a weekly load profile for a total of 2000 households belonging to these types has been generated. Each household profile has distinct characteristics to observe how the generated data changes with different profiles. Detailed explanations of both household profiles and the specified household appliances will be presented in the following sections.

The simulations of Part 1&2 with the stopping criteria of at least 85% accuracy with respect to the reference data is carried out for $7\,[\mathrm{h}] : 23\,[\mathrm{min}]$, respectively. Various graphs focused on different aspects have been plotted to enable the evaluation of the generated load profile data from multiple perspectives, and all this data will be presented at the end of this section.

## 4.2 Household Types

In the conducted simulation study, five types of household load profiles were determined. Some attributes of these profiles can be seen in Table 1. The most critical point in household profiles is the population of the household. As stated in Section 2.3, the energy consumption of appliances is directly proportional to the population of the house. After population, the second critical point is the number of employed individuals in the household. Because Eq.2.3, which calculates the probability of an appliance running at any given hour, takes the number of individuals in the household as input for the selected hour. This situation requires dynamically monitoring the household population continuously. Therefore, during working hours, if there is an employed person in the household, the household population will be lower, and the probability of the appliance running will decrease accordingly. Whether the household profile owns an electric vehicle or not is also one of the significant factors that affect the load profile. To observe this, electric vehicles are definitively assigned to Type-4 and Type-5 profiles, while in other profiles, assignments are made randomly. Regarding the allocation of household appliances, additional configurations are available in household profiles. For instance, we can specify certain household appliances that must be assigned or must not be assigned. Since this aspect is not relevant to the main purpose of our simulation study, the related configurations are not provided in Table 4.1.

**Table 4.1** Household Types

| | **Type 1** | **Type 2** | **Type 3** | **Type 4** | **Type 5** |
|---|---|---|---|---|---|
| **Number of Employed Persons** | 0 | 1 | 1 | 2 | 2 |
| **Number of Unemployed Persons** | 2 | 1 | 2 | 2 | 3 |
| **Ownership Status of EV** | Random | Random | Random | Certain | Certain |
| **Sleeping Period** | 00:00<br>08:00 | 00:00<br>06:00 | 23:00<br>06:00 | 23:00<br>07:00 | 23:00<br>07:00 |

## 4.3 Appliances and EVs

The household appliances we selected to use in our simulation study and the collected survey data related to them are given in Table 4.2. While some household appliances' daily usage is provided in "hours", others are given in "cycles." Such appliances have a specified starting time, as mentioned in Section 2.5, and they operate for a fixed duration before completing their operation.

The features of the appliances are not limited to Table 4.2. In order to increase the system's flexibility and the realism of the generated data, additional constraint

parameters can be introduced for the usage of household appliances. For instance, a time interval can be defined during which an appliance can operate, or the simultaneous operation of multiple appliances can be restricted. Additionally, during the random assignment of appliances to households, ensuring that any appliance is not assigned to households with specific profiles can be implemented. Another characteristic of appliances is whether they require human intervention during their operation. With this configuration, it becomes possible for the appliances to operate even during the nighttime sleep period.

**Table 4.2** Daily Operation of Appliances[59–62]

| **Appliances** | **Daily Operation** |
|---|---|
| Washing Machine and Tumble Dryer | $[0-2.2]$ persons $\rightarrow 0.38$ cycle<br>$[2.3-2.7]$ persons $\rightarrow 0.45$ cycle<br>$[>2.2]$ persons $\rightarrow 0.48$ cycle |
| Dishwaher | Per person $\rightarrow 1.3$ cycle |
| Television | Per person $\rightarrow 4.23$ hours |
| Lamps | Per two persons $\rightarrow 3(+1)^* \times 12$ hours |
| EV | 8 hours |
| Fan | Per person $\rightarrow 2.3$ hours |
| Toastar | Per person $\rightarrow 1$ hour |
| Microwave | Per two person $\rightarrow 4(+1)^*$ hours |
| Kettle | Per person $\rightarrow 0.5$ hour |
| Oven | Per person $\rightarrow 1$ hour |
| Computer | Per person $\rightarrow 5$ hours |
| Air Conditioner | 10 hours (Only summer) |
| Hair Dryer | Per person $\rightarrow 10$ minutes |
| Water Heater | Per person $\rightarrow 30$ minutes |
| Fridge/Deep Freezer | Non-stop |

(*): For each extra person

We configured electric vehicles separately from household appliances. Charging electric vehicles occurs under much more stable conditions. Based on our observations, we determined the charging times of electric vehicles to be outside working hours, including weekends. Additionally, since the battery status of an electric vehicle creates variations when charging starts, we selected predetermined values for this aspect. Table 4.3 shows the electric vehicle category and its characteristics used in the simulation. The charging process takes place randomly

during a time slot outside working hours. However, the critical point here is that the charging process continues until the battery is fully charged. Therefore, the randomly selected charging time will be constrained based on the current battery status. For instance, if the battery is 25% charged at that moment, the charging process should start earlier to ensure that the battery is fully charged by the end of the time slot. The initial battery charge levels are shown in Table 2. In our simulation study, we used only one electric vehicle model. However, our software is capable of accommodating more models. When multiple vehicles are added, their assignment will be random, selecting one from the available options.

**Table 4.3** Electric vehicle characteristics

| | **Battery Capacity (kWh)** | **Charger Power (kW)** | **Possible Charge Levels (%)** |
|---|---|---|---|
| **Chevy Volt** | 16 | 3.3 | 25, 50, 75, 100 |

## 4.4 Simulation Parameters

In the performed simulation, the configuration given in Table 4.4 was used to predict usage habits using genetic algorithms. The number of chromosomes was selected as 16 for each population. As stated in Eq.2.4, we already know that there are 3 genes in each chromosome. The range of initial values for the genes is an important point. However, even if this range is incomplete, the genetic algorithm will converge to the correct point by compensating for the deficiency. For instance, it was observed from the simulation results that some of the predicted parameters were negative. When examining the configuration for the selection stage, it is stated that 8 chromosomes will be selected from the entire population of obtained fitness values, and 2 of them will be selected through elitism. The reason for selecting 8 chromosomes is that many crossover operators accept two chromosomes as input and produce only 2 offspring as output. This way, the population reaches the number of 16 again. The importance of elitism was mentioned in Section 2.4. In the conducted simulation, elitism was used in line with this principle. The parameters given for another crucial aspect of the genetic algorithm, mutation, are fully consistent with Eq.2.13. Accordingly, the probability of chromosomes undergoing mutation is set to a maximum of 50%. Each chromosome that undergoes mutation reduces this probability by 10% for the next chromosome. When the given values are examined with Eq.2.13, the following result emerges: in each population, a maximum of 5 chromosomes can undergo mutation. Additionally, considering our mutation algorithm, there is also a possibility that no chromosome will undergo mutation. For

each gene of the mutated chromosomes, a similar dynamic probability calculation is performed. However, these values are not included in the table as they are deemed unnecessary. The last parameter in the mutation section indicates to what extent each mutated gene can change. According to the given configuration parameter, each gene will change at a maximum rate of $\pm 5\%$. The exact value will be randomly selected within the given interval. The last section of Table 4.4 is dedicated to the completion conditions of the genetic algorithm. The maximum generation count is a precaution to prevent the algorithm from getting stuck. If the desired success conditions are not achieved, the algorithm will make a maximum of 30 generations of attempts, and then the most successful of the obtained chromosomes will be provided as the output. The target score and target chromosome count represent the success rate we aim for. In other words, if 12 out of the 16 chromosomes have fitness values of 15% or lower for the relevant device, the genetic algorithm process can be completed. In this case, the most successful 12 chromosomes will be provided as the output.

**Table 4.4** Genetic algorithm paramaters

| **Parameter** | **Value** |
|---|---|
| ***Population*** | |
| Chromosome Count | 16 |
| Gene Count | 3 |
| Initial Gen Interval | [0, 200] |
| ***Selection*** | |
| Chosen Count | 8 |
| Elite Count | 2 |
| ***Mutation*** | |
| Max. Mutation Percentage | 50 (%) |
| Each Chromosome Effect | 10 (%) |
| Mutation Rate | 5 (%) |
| ***Termination Conditions*** | |
| Max. Generation Count | 30 |
| Target Score | 15 (%) |
| Target Chromosome Count | 12 |

The parameters related to the data generated in the simulation are provided in Table 4.5. The number of households is set to 2000. However, we can see from Table 4.1 that there are 5 household profile types and their characteristics. The 2000 households to be generated are distributed randomly according to the given 5 profile types. The number of weeks is set to 1, meaning that we will have one week's worth of data in the end. As mentioned in Section 2.3, the day sector parameter is set to

4. To reduce data size and increase applicability, the sample time is chosen as 5 minutes. Finally, as discussed in Section 2.2, the simulation is performed using the PRNG (Pseudorandom Number Generator) method.

**Table 4.5** Simulation parameters

| Parameter | Value |
|---|---|
| Household Count | 2000 |
| Week Count | 1 |
| Day Sector Count | 4 |
| Sample Time | 5min |
| Randomization Method | PRNG |

## 4.5 Results

The results of the simulation conducted with the specified parameters and survey data are presented in this section. Fig.4.1 shows the daily average data for all household types. When looking at this table, it is evident that Type-4 and Type-5 have significantly higher average consumption compared to other types. The main reason for this can be found in Table- 4.1. As indicated in the table, Type-4 and Type-5 have larger populations compared to other consumer profiles, which leads to higher consumption levels. Another noticeable point in Fig.4.1 is the distinct breakpoints, especially around 07:00. The reason for these breakpoints is the EV factor. The charging time for EV parameters is limited between 18:00 and 07:00, resulting in breakpoints occurring at those times. To fully understand this phenomenon, the EV loads were removed from the generated data, and the same graph was redrawn. Fig.4.2 shows the daily average data for all user types without the EV consumption data. It can be observed that there are no sharp breakpoints as in Fig.4.1, and the average consumption further decreases.

Fig.4.3 presents the overall average consumption for all household types. The EV related breakpoints mentioned above are also evident in this table. When comparing household profiles, equivalent results are observed corresponding to the applied configuration.

Fig.4.4 shows the RMS graph of the generated data. This graph clearly indicates the usage difference between different household types. Another observation from the graph is that the consumption on Saturdays and Sundays is higher than on weekdays. The main reason for this is the presence of working individuals in the configured households. Since these individuals spend less time at home on

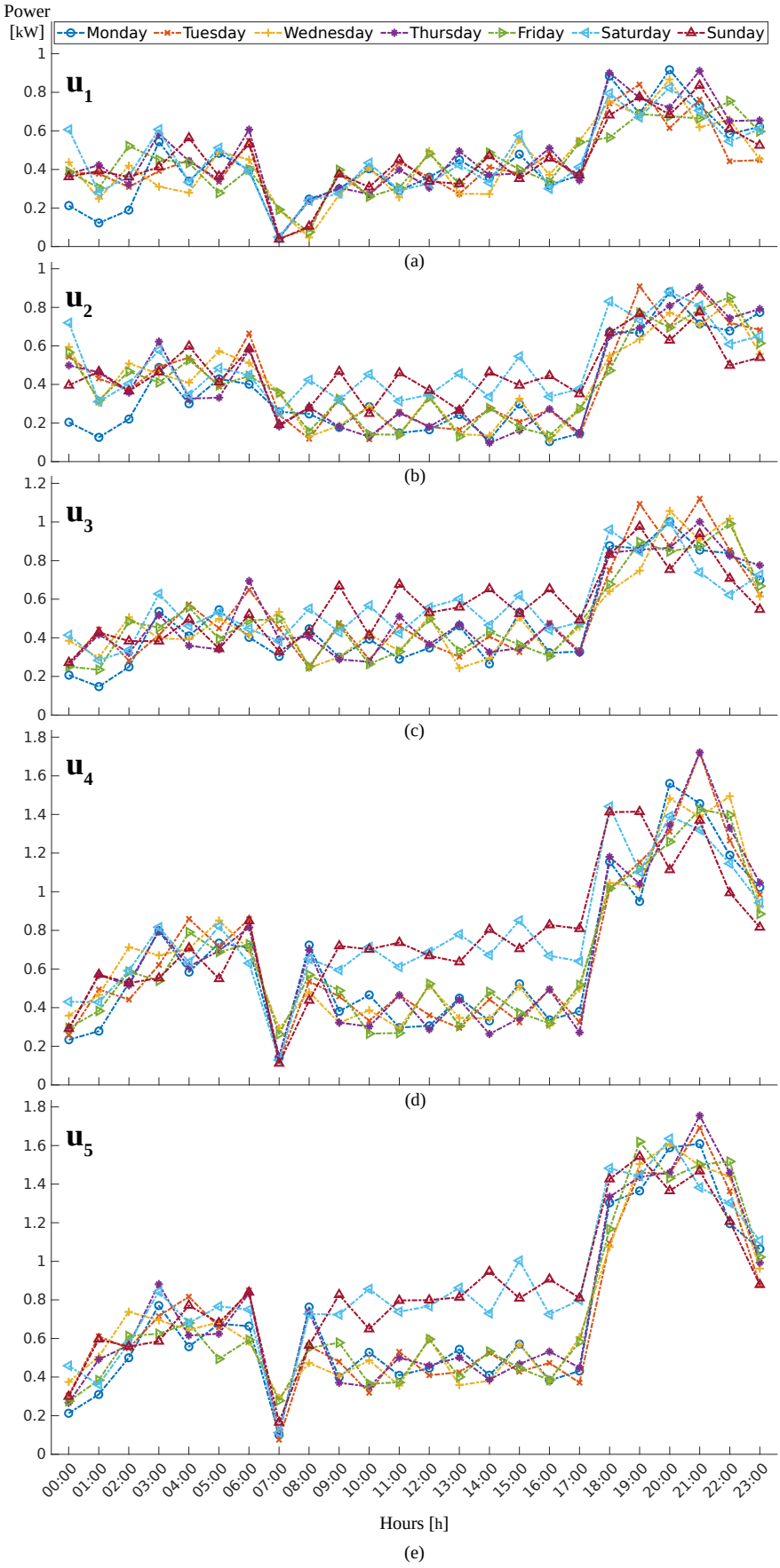



**Figure 4.1** Daily average load profile for each household type

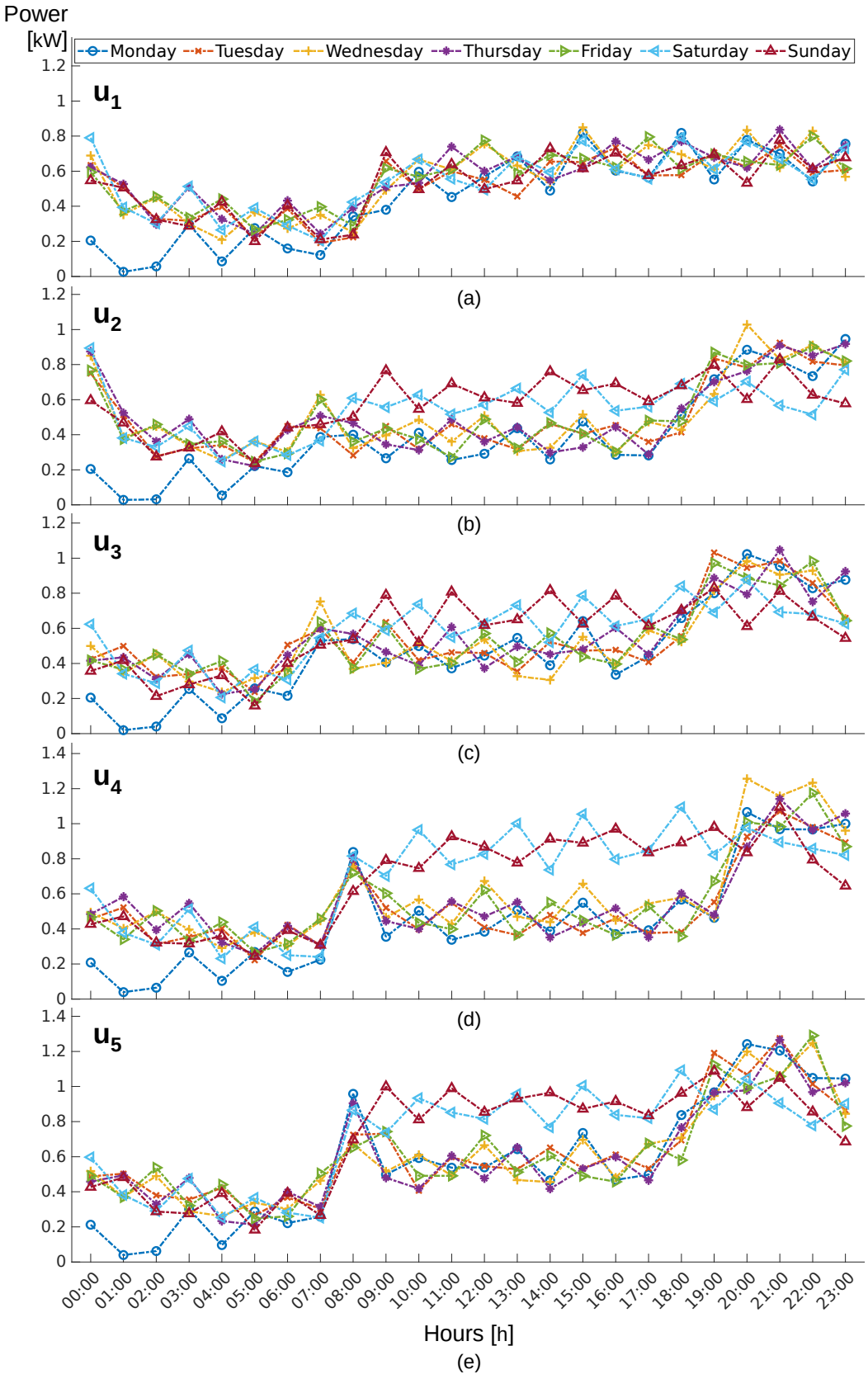



**Figure 4.2** Daily average load profile for each household type without EV

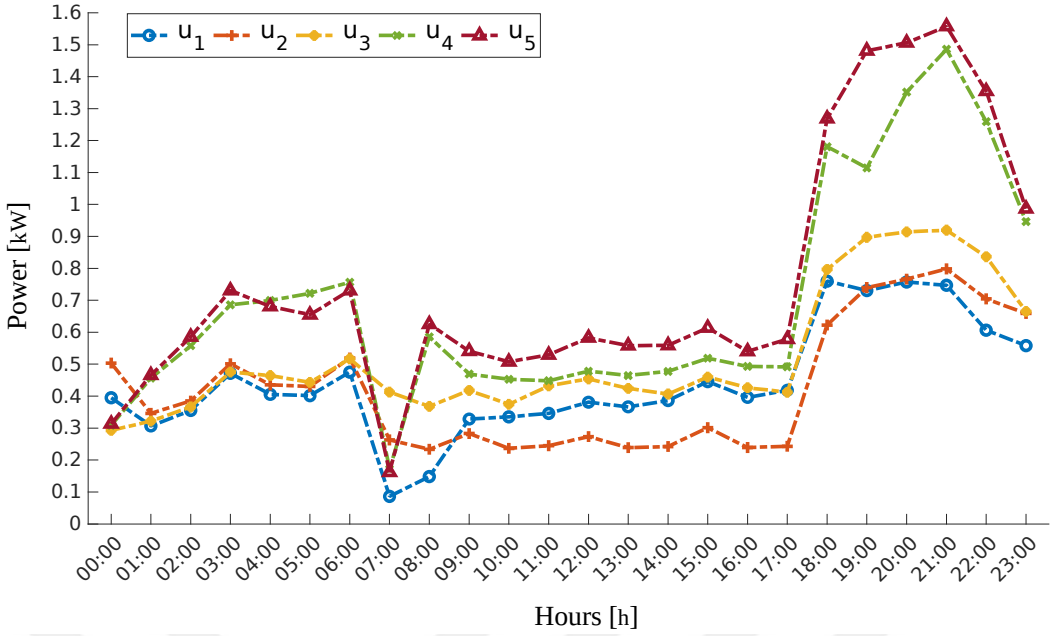



**Figure 4.3** Overall average of the user type profiles

weekdays, the consumption naturally decreases. This table clearly demonstrates how well the generated data aligns with the applied parameters.

In Fig.4.5, the average usages of household types are compared for each day of the week. Particularly, the difference between Type-4 and Type-5 households and other types is more pronounced on Saturdays and Sundays. Since we defined weekdays as working days in the simulation parameters, it is expected to have such differences on weekends. This graph also indicates that the shape of consumption during the night is influenced by EV charging.

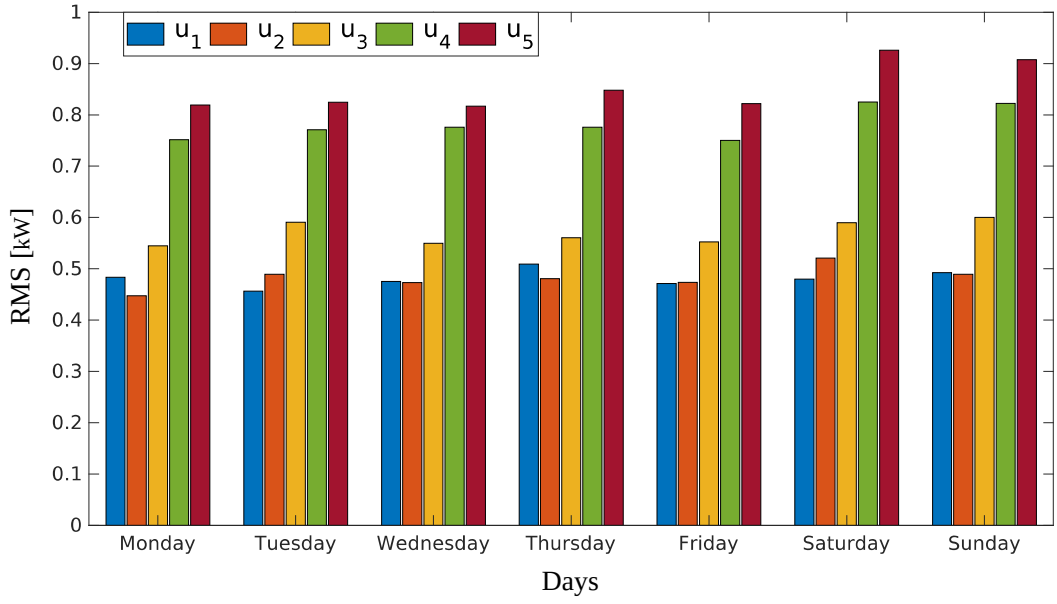



**Figure 4.4** RMS value of the user type profiles

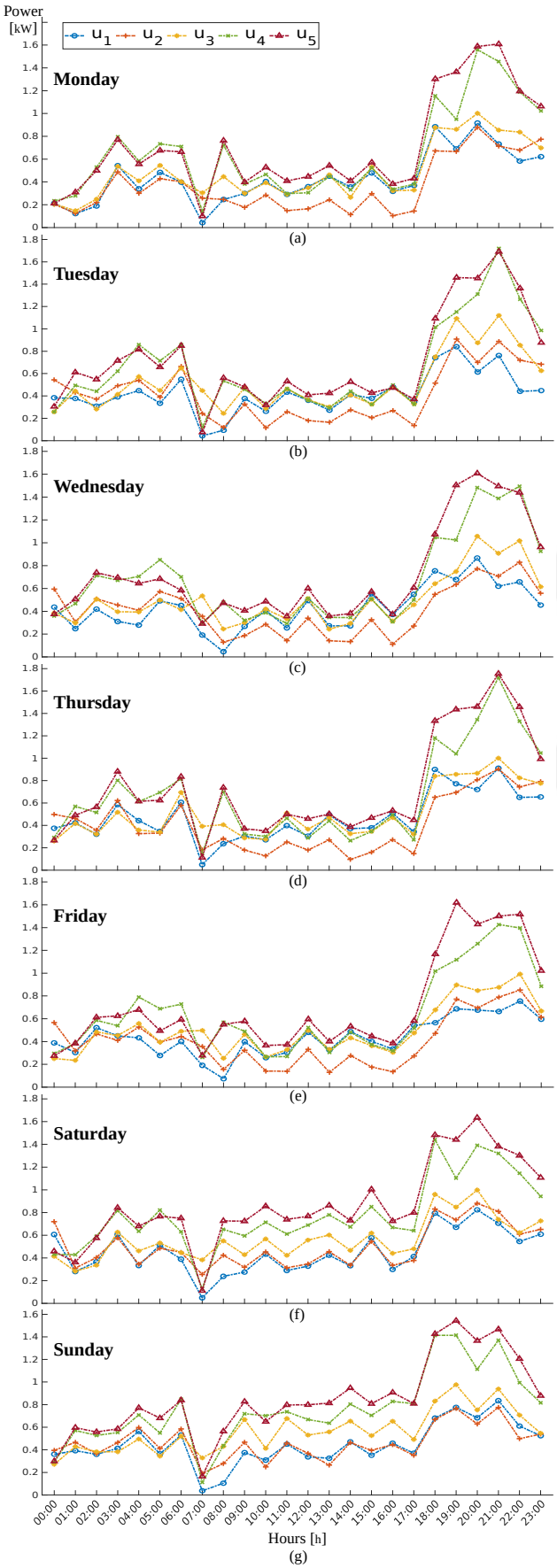



**Figure 4.5** Comparison of the user load profiles for each day

Fig.4.6 shows the load profiles of different household types. The purpose of this graph is to illustrate the pattern formed when overlaying the intended profiles. The resulting patterns reflect the average consumption graph given in Fig.4.1. Additionally, we can observe the proximity relationship between households of the same type through Fig.4.6. These results demonstrate that our study generates load profiles in accordance with the input parameters.

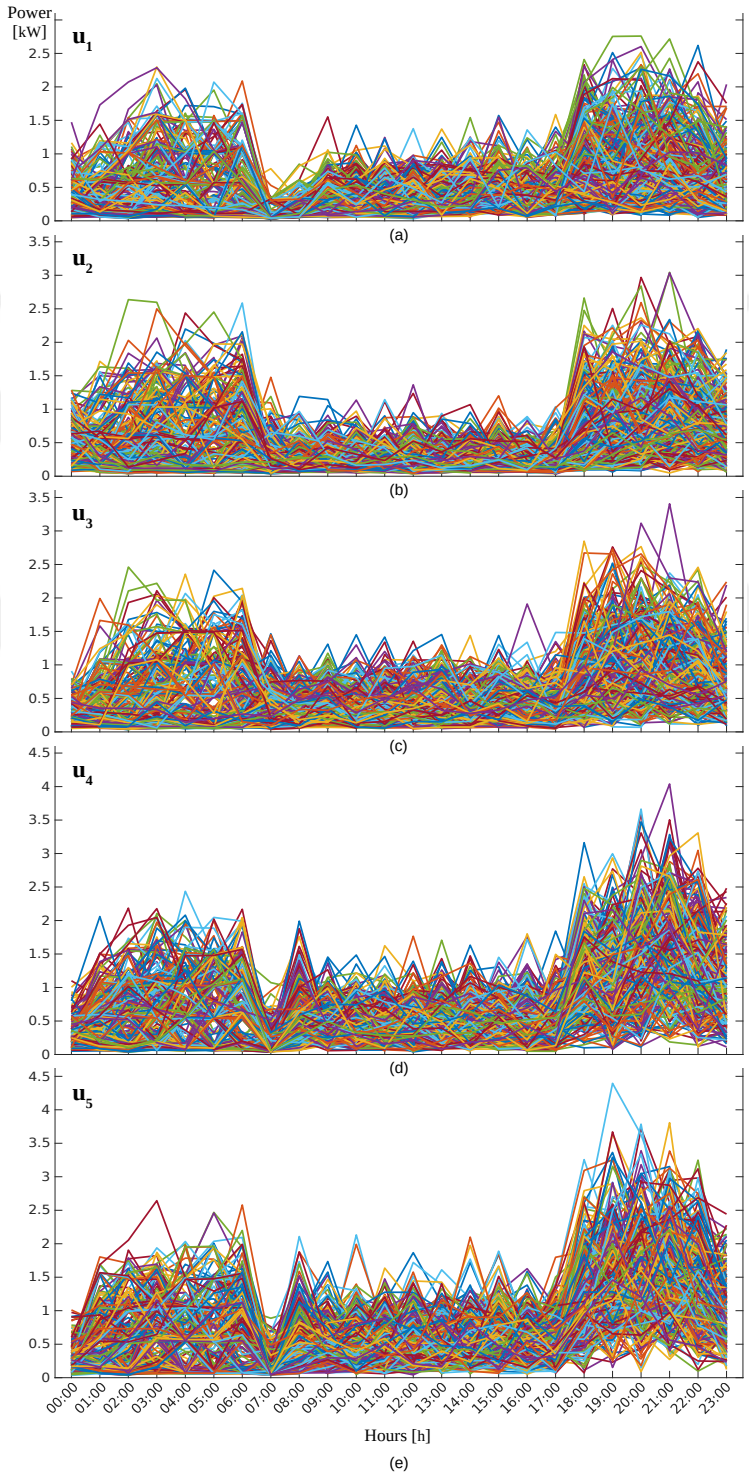



**Figure 4.6** 2000 profiles corresponding to the user types

# 5
# CONCLUSION

With the increasing decentralization of grid control and management, driven by the rise in renewable power sources, electrified vehicles, and end-user DR applications, the stability of the entire system faces new challenges. Key factors now involve compensating for the intermittency of renewables and accurately predicting trends and changes in end-user load profiles. Real-time load data collection is optimal, but it presents challenges related to cost, privacy concerns, and technical complexities. In response to this, we present a novel load demand generator tool designed to create datasets for DR applications, clustering, and other purposes.

The proposed algorithms, combined with a customized genetic algorithm, address the fundamental requirements of accuracy, randomness, and diversity in load profiling. To ensure the validity of synthetic profiles, we impose stopping criteria on the customized GA, guaranteeing a minimum of 85% accuracy compared to statistical data. Pseudo codes and mathematical formulations for the proposed algorithms are provided. The entire process is thoroughly simulated, yielding significant and noteworthy results. These results can be effectively utilized in DR, clustering, or load prediction applications. Furthermore, the load profiling scheme can be readily implemented in any region by acquiring statistical data on residents' consumption habits.

When the averages of the obtained load profile data are taken according to household types, the differences of the initially configured five household types become evident. The emergence of such a table led us to make the following interpretation: The process presented in this study can be reversed to determine to which household type a household belongs if we have load profile data measured with APM devices. Thus, it becomes possible to easily obtain information about which household types exist in any region. Such a study could create a subheading for "clustering" studies on load profiles.

Another point is that the simulation results show that the integration of electric

vehicles into the grid creates very sharp spikes in load profiles. Especially when the majority of the population has similar habits regarding electric vehicles, these spikes become even more critical. Based on these findings, applying dynamic pricing or load shifting techniques for electric vehicles emerges as an important research topic to smooth out these spikes and achieve a more flattened load profile curve. Additionally, although household appliances of the consumer type were used in the simulation study, this study actually allows for the inclusion of producer-type household appliances in the equation. By using accurate statistical data, the impact of producer-type household appliances on the overall load profiles can be observed, and the advantages of the smart grid concept over the traditional grid model can be observed.

As part of our future endeavors, we aim to extend the proposed method to address profile generation challenges for other load types, including commercial and industrial sectors. Additionally, we plan to explore and investigate the relationship between various survey types and their impact on energy usage patterns.

# REFERENCES


[1] V. Marissen, "The european parliament and eu secondary legislation: Improved scrutiny practices and upstream involvement for delegated acts and implementing acts," in *The European Parliament in Times of EU Crisis*, Springer, 2019, pp. 143–159.

[2] I. E. Agency, *Electricity Information 2019*. 2019, p. 703. doi: 10.1787/e0ebb7e9-en. [Online]. Available: https://www.oecd-ilibrary.org/content/publication/e0ebb7e9-en.

[3] Z. Wang, T. Hong, "Generating realistic building electrical load profiles through the generative adversarial network (gan)," *Energy and Buildings*, vol. 224, p. 110 299, 2020.

[4] Y. Ge, C. Zhou, D. M. Hepburn, "Domestic electricity load modelling by multiple gaussian functions," *Energy and Buildings*, vol. 126, pp. 455–462, 2016.

[5] Y. Gu, Q. Chen, K. Liu, L. Xie, C. Kang, "Gan-based model for residential load generation considering typical consumption patterns," in *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, IEEE, 2019, pp. 1–5.

[6] A. Satre-Meloy, M. Diakonova, P. Grünewald, "Cluster analysis and prediction of residential peak demand profiles using occupant activity data," *Applied Energy*, vol. 260, p. 114 246, 2020.

[7] S. Yilmaz, J. Chambers, M. K. Patel, "Comparison of clustering approaches for domestic electricity load profile characterisation-implications for demand side management," *Energy*, vol. 180, pp. 665–677, 2019.

[8] G. Le Ray, P. Pinson, "Online adaptive clustering algorithm for load profiling," *Sustainable Energy, Grids and Networks*, vol. 17, p. 100 181, 2019.

[9] B. V. M. Vasudevarao, M. Stifter, P. Zehetbauer, "Methodology for creating composite standard load profiles based on real load profile analysis," in *2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, IEEE, 2016, pp. 1–6.

[10] M. Ali, M. Adnan, M. Tariq, H. V. Poor, "Load forecasting through estimated parametrized based fuzzy inference system in smart grids," *IEEE Transactions on Fuzzy Systems*, 2020.

[11] L. Chuan, A. Ukil, "Modeling and validation of electrical load profiling in residential buildings in singapore," *IEEE Transactions on Power Systems*, vol. 30, no. 5, pp. 2800–2809, 2014.

[12] B. Stephen, A. J. Mutanen, S. Galloway, G. Burt, P. Järventausta, "Enhanced load profiling for residential network customers," *IEEE Transactions on Power Delivery*, vol. 29, no. 1, pp. 88–96, 2013.

[13] J. A. Machado, P. M. Carvalho, L. A. Ferreira, "Building stochastic non-stationary daily load/generation profiles for distribution planning studies," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 911–920, 2017.

[14] J. Jeyaranjani, D. Devaraj, "Load profile generation for dr program," in *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, IEEE, 2019, pp. 1–5.

[15] W. Labeeuw, G. Deconinck, "Residential electrical load model based on mixture model clustering and markov models," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1561–1569, 2013.

[16] M.-S. Kang, C.-S. Chen, Y.-L. Ke, C.-H. Lin, C.-W. Huang, "Load profile synthesis and wind-power-generation prediction for an isolated power system," *IEEE Transactions on Industry Applications*, vol. 43, no. 6, pp. 1459–1464, 2007.

[17] M. Parti, C. Parti, "The total and appliance-specific conditional demand for electricity in the household sector," *The Bell journal of economics*, pp. 309–321, 1980.

[18] D. J. Aigner, C. Sorooshian, P. Kerwin, "Conditional demand analysis for estimating residential end-use load profiles," *The Energy Journal*, vol. 5, no. 3, 1984.

[19] Y. J. Huang, J. Brodrick, "A bottom-up engineering estimate of the aggregate heating and cooling loads of the entire us building stock," 2000.

[20] R. Subbiah, K. Lum, A. Marathe, M. Marathe, "Activity based energy demand modeling for residential buildings," in *2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*, IEEE, 2013, pp. 1–6.

[21] D. G. Fiebig, R. Bartels, D. J. Aigner, "A random coefficient approach to the estimation of residential end-use load profiles," *Journal of Econometrics*, vol. 50, no. 3, pp. 297–327, 1991.

[22] I. Richardson, M. Thomson, D. Infield, C. Clifford, "Domestic electricity use: A high-resolution energy demand model," *Energy and buildings*, vol. 42, no. 10, pp. 1878–1887, 2010.

[23] C. Bucher, G. Andersson, "Generation of domestic load profiles-an adaptive top-down approach," in *Proceedings of PMAPS*, vol. 2012, 2012, pp. 10–14.

[24] J. Ortiz, F. Guarino, J. Salom, C. Corchero, M. Cellura, "Stochastic model for electrical loads in mediterranean residential buildings: Validation and applications," *Energy and Buildings*, vol. 80, pp. 23–36, 2014.

[25] K. Foteinaki, R. Li, C. Rode, R. K. Andersen, "Modelling household electricity load profiles based on danish time-use survey data," *Energy and Buildings*, vol. 202, p. 109 355, 2019.

[26] R. Yao, K. Steemers, "A method of formulating energy load profile for domestic buildings in the uk," *Energy and buildings*, vol. 37, no. 6, pp. 663–671, 2005.

[27] J. Bajada, M. Fox, D. Long, "Load modelling and simulation of household electricity consumption for the evaluation of demand-side management strategies," in *IEEE PES ISGT Europe 2013*, IEEE, 2013, pp. 1–5.

[28] A. J. Collin, G. Tsagarakis, A. E. Kiprakis, S. McLaughlin, "Development of low-voltage load models for the residential load sector," *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2180–2188, 2014.

[29] M. Muratori, M. C. Roberts, R. Sioshansi, V. Marano, G. Rizzoni, "A highly resolved modeling technique to simulate residential power demand," *Applied Energy*, vol. 107, pp. 465–473, 2013.

[30] C. F. Walker, J. L. Pokoski, "Residential load shape modelling based on customer behavior," *IEEE transactions on power apparatus and systems*, no. 7, pp. 1703–1711, 1985.

[31] A. Capasso, W. Grattieri, R. Lamedica, A. Prudenzi, "A bottom-up approach to residential load modeling," *IEEE transactions on power systems*, vol. 9, no. 2, pp. 957–964, 1994.

[32] R. Bartels, D. G. Fiebig, "Residential end-use electricity demand: Results from a designed experiment," *The Energy Journal*, vol. 21, no. 2, 2000.

[33] G. Mihalakakou, M. Santamouris, A. Tsangrassoulis, "On the energy consumption in residential buildings," *Energy and buildings*, vol. 34, no. 7, pp. 727–736, 2002.

[34] Y. Shimoda, T. Fujii, T. Morikawa, M. Mizuno, "Residential end-use energy simulation at city scale," *Building and environment*, vol. 39, no. 8, pp. 959–967, 2004.

[35] J. V. Paatero, P. D. Lund, "A model for generating household electricity load profiles," *International journal of energy research*, vol. 30, no. 5, pp. 273–290, 2006.

[36] J. Widén, M. Lundh, I. Vassileva, E. Dahlquist, K. Ellegård, E. Wäckelgård, "Constructing load profiles for household electricity and hot water from time-use data—modelling approach and validation," *Energy and buildings*, vol. 41, no. 7, pp. 753–768, 2009.

[37] M. M. Armstrong, M. C. Swinton, H. Ribberink, I. Beausoleil-Morrison, J. Millette, "Synthetically derived profiles for representing occupant-driven electric loads in canadian housing," *Journal of Building Performance Simulation*, vol. 2, no. 1, pp. 15–30, 2009.

[38] J. Widén, E. Wäckelgård, "A high-resolution stochastic model of domestic activity patterns and electricity demand," *Applied energy*, vol. 87, no. 6, pp. 1880–1892, 2010.

[39] Z. Ren, P. Paevere, C. McNamara, "A local-community-level, physically-based model of end-use energy consumption by australian housing stock," *Energy policy*, vol. 49, pp. 586–596, 2012.

[40] E. B. Alzate, N. H. Mallick, J. Xie, "A high-resolution smart home power demand model and future impact on load profile in germany," in *2014 IEEE International Conference on Power and Energy (PECon)*, IEEE, 2014, pp. 53–58.

[41] B. J. Johnson, M. R. Starke, O. A. Abdelaziz, R. K. Jackson, L. M. Tolbert, "A matlab based occupant driven dynamic model for predicting residential power demand," in *2014 IEEE PES T&D Conference and Exposition*, IEEE, 2014, pp. 1–5.

[42] D. Fischer, A. Härtl, B. Wille-Haussmann, "Model for electric load profiles with high time resolution for german households," *Energy and Buildings*, vol. 92, pp. 170–179, 2015.

[43] A. Marszal-Pomianowska, P. Heiselberg, O. K. Larsen, "Household electricity demand profiles–a high-resolution load model to facilitate modelling of energy flexible buildings," *Energy*, vol. 103, pp. 487–501, 2016.

[44] M. Nijhuis, M. Gibescu, J. Cobben, "Bottom-up markov chain monte carlo approach for scenario based residential load modelling with publicly available data," *Energy and Buildings*, vol. 112, pp. 121–129, 2016.

[45] T. Ding, H. Liang, W. Xu, "An analytical method for probabilistic modeling of the steady-state behavior of secondary residential system," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2575–2584, 2016.

[46] B. Gao, X. Liu, Z. Zhu, "A bottom-up model for household load profile based on the consumption behavior of residents," *Energies*, vol. 11, no. 8, p. 2112, 2018.

[47] M. Sepehr, R. Eghtedaei, A. Toolabimoghadam, Y. Noorollahi, M. Mohammadi, "Modeling the electrical energy consumption profile for residential buildings in iran," *Sustainable cities and society*, vol. 41, pp. 481–489, 2018.

[48] N. Mihaylov, B. Evstatiev, S. Kadirova, T. Gueorguiev, T. Georgieva, A. Evtimov, "Load profile of typical residential buildings in bulgaria," in *IOP Conference Series: Earth and Environmental Science*, IOP Publishing, vol. 172, 2018, p. 012 035.

[49] S. Picek, D. Jakobovic, M. Golub, "On the recombination operator in the real-coded genetic algorithms," in *2013 IEEE Congress on Evolutionary Computation*, IEEE, 2013, pp. 3103–3110.

[50] M. Haahr, *RANDOM.ORG: true random number service*, `https://www.random.org`, Accessed: 2023-07-14, 1998–2023.

[51] S. Katoch, S. S. Chauhan, V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia tools and applications*, vol. 80, pp. 8091–8126, 2021.

[52] A. E. Eiben, J. E. Smith, *Introduction to evolutionary computing*. Springer, 2015.

[53] D. Dumitrescu, B. Lazzerini, "B., lc jain and a. dumitrescu," *Evolutionary Computation. CRC Press, Florida, USA*, 2000.

[54] D. Ortiz-Boyer, C. Hervas-Martinez, N. Garcia-Pedrajas, "Improving crossover operator for real-coded genetic algorithms using virtual parents," *Journal of Heuristics*, vol. 13, pp. 265–314, 2007.

[55] L. J. Eshelman, J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," in *Foundations of genetic algorithms*, vol. 2, Elsevier, 1993, pp. 187–202.

[56] A. H. Wright, "Genetic algorithms for real parameter optimization," in *Foundations of genetic algorithms*, vol. 1, Elsevier, 1991, pp. 205–218.

[57] M. Takahashi, H. Kita, "A crossover operator using independent component analysis for real-coded genetic algorithms," in *Proceedings of the 2001 congress on evolutionary computation (ieee cat. no. 01th8546)*, IEEE, vol. 1, 2001, pp. 643–649.

[58] Z. Michalewicz, Z. Michalewicz, *GAs: Why Do They Work?* Springer, 1996.

[59] O. Erdinc, "Economic impacts of small-scale own generating and storage units, and electric vehicles under different demand response strategies for smart households," *Applied Energy*, vol. 126, pp. 142–150, 2014.

[60] C. Pakula, R. Stamminger, "Electricity and water consumption for laundry washing by washing machine worldwide," *Energy efficiency*, vol. 3, no. 4, pp. 365–382, 2010.

[61] R. Krantz-Kent, "Television, capturing america's attention at prime time and beyond," *Beyond the Numbers: Special Studies & Research*, vol. 7, no. 14, pp. 1–11, 2018.

[62] M. C. Kisacikoglu, A. Bedir, B. Ozpineci, L. M. Tolbert, "Phev-ev charger technology assessment with an emphasis on v2g operation," *Oak Ridge Nat. Lab., Oak Ridge, TN, USA, Tech. Rep. ORNL/TM-2010/221*, 2012.

# PUBLICATIONS FROM THE THESIS

## Papers

1. F. Yılmaz, Y. Eren, "A novel load profile generation method based on the estimation of regional usage habit parameters with genetic algorithm," *Electric Power Systems Research 217 (2023): 109165.*