Efficient Shapley Value Approximation Methods

for Cost Redistribution in Energy Communities

Sho Cremers

TU Delft

Efficient Shapley Value Approximation Methods
for Cost Redistribution in Energy Communities

by

Sho Cremers

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on on Wednesday the 13th of July 2022 at 15:00.

Student number:	5052602	
Project duration:	November 2021 - July 2022	
Thesis committee:	Dr. V. Robu	CWI & TU Delft, **supervisor**
	Prof. Dr. Ir. J.A. La Poutré	CWI & TU Delft
	Dr. M.M. de Weerdt	TU Delft

An electronic version of this thesis is available at: `http://repository.tudelft.nl/`.

Cover Image: Sunset by Usukhbayar Gankhuyag on Unsplash

TU Delft

Acknowledgements

Twelve months ago, while working on a course project, I was lucky enough to find a project that I found fascinating and thus began the journey to my Master's thesis (with many milestones on the way). My supervisor, Dr. Valentin Robu, to whom I owe my utmost appreciation, encouraged me to continue working on this project and also provided me with the opportunity to do so while interning at the famous CWI. He gave me invaluable guidance and support, which led to us successfully publishing my first-ever paper and presenting the work at the ACM e-Energy conference. This experience has taught me a number of things, not only about the topic at hand but also about myself as a researcher and how to best approach this thesis. I would also like to thank all of the Intelligent and Autonomous Systems group members at CWI for encouraging me and letting me pick their brilliant brains. I also want to thank Peter Zhang. We started working on our thesis simultaneously, with whom I have been able to discuss many ideas and keep company while cooped up working away in the office. I would also like to express my gratitude to my partner Sukhleen Kaur for cheering me up whenever I felt overwhelmed or stressed. Finally, I want to thank my parents for their continuous support from the other side of the world and for always believing in me.

Sho Cremers
Rotterdam, July 2022

Abstract

With the emergence of energy communities, where a number of prosumers (consumers with their own energy generation) invest in shared renewable generation capacity and battery storage, the issue of fair allocation of benefits and costs has become increasingly important. The Shapley value, a solution concept in cooperative game theory initially proposed by Nobel prize-winning economist Lloyd Shapley, has attracted increasing interest for redistribution in energy settings. However, due to its high time complexity, it is intractable beyond communities of a few dozen prosumers. This study proposes a new deterministic method for approximating the Shapley value in realistic community energy settings and compares its performance with existing methods. To provide a benchmark for the comparisons of these methods, we also design a novel method to compute the exact Shapley value for communities of up to several hundred agents by clustering consumers into a smaller number of demand profiles. Experimental analyses with large-scale case studies of a community of up to 200 household consumers in the UK show that the newly proposed method can achieve very close redistribution to the exact Shapley values but at a much lower (and practically feasible) computation cost. Furthermore, it performed similarly to the probabilistic, state-of-the-art approximation method while having smaller time complexity as well as other desirable characteristics for cost redistribution in energy communities.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
DF	Depreciation Factor
DoD	Depth of Discharge
LV	Low-Voltage
P2P	Peer-to-Peer
RES	Renewable Energy Source
SD	Standard Deviation
SoC	State of Charge

Symbols

Symbol	Definition	Unit
$\mathcal{N}$	set of agents in the community	–
N	number of agents	–
i	for agents (households)	–
T	number of time steps	–
t	for time steps	–
$d_i(t)$	power demand of agent i at time t	kW
$d_{\mathcal{N}}(t)$	power demand of community $\mathcal{N}$ at time t	kW
$g(t)$	power generated by community RES (wind turbine) at time t	kW
$p^{\text{bat}}(t)$	power of community battery at time t, negative when charging and positive when discharging	kW
$p^{bat,\text{max}}$	maximum (dis)charging power of battery	kW
$p^{\text{grid}}(t)$	power to/from utility grid at time t, negative when selling and positive when buying	kW
$SoC(t)$	battery state of charge at time t	%
SoC^{max}	maximum battery SoC	%
SoC^{min}	minimum battery SoC	%
Δt	duration of time period ts	hour
η^c	charging efficiency of battery	–
η^d	discharging efficiency of battery	–
DF	depreciation factor of battery	–
$\text{DF}^{\text{regular}}$	depreciation factor by regular cycles	–
$\text{DF}^{\text{irregular}}$	epreciation factor by irregular cycles	–
DoD	depth of discharge of battery	%
$N_{cycles}^{\text{DoD,max}}$	maximum number of cycles allowed at specific DoD, provided from manufacturer specification	–
$n_{cycles}^{\text{DoD,regular/irregular}}$	number of regular/irregular cycles at specific DoD	–
$SoC_l^{start/end}$	starting/ending SoC of cycle l	%
$e^b(t)$	imported energy at time t	kWh
$e^s(t)$	exported energy at time t	kWh
$\tau^s(t)$	export tariff at time t	pence/kWh

Symbol	Definition	Unit
$\tau^b(t)$	import tariff at time t	pence/kWh
$c_T^{\text{grid}}(\mathcal{N})$	annual cost of community $\mathcal{N}$ importing energy from utility grid, with $T = 1$ year	£
$c_T^{\text{wind}}(\mathcal{N})$	annual cost of wind turbine of community $\mathcal{N}$, with $T = 1$ year	£
$c_T^{\text{bat}}(\mathcal{N})$	annual cost of battery of community $\mathcal{N}$, with $T = 1$ year	£
$c_T(\mathcal{N})$	total annual energy cost of community $\mathcal{N}$, with $T = 1$ year	£
$c(\mathcal{N})$	cost function, equivalent to $c_T(\mathcal{N})$	–
ϕ_i	annual cost of agent i according to Shapley value	£
$\mathcal{S}$	for subcoalitions formed by agents in community $\mathcal{N}$	–
j	for strata	–
MC_i	annual cost of agent i according to last marginal contribution (unnormalised)	£
$\overline{MC}_i$	annual cost of agent i according to normalised last marginal contribution	£
$d_{p_{-i}}(t)$	average power demand of the community without agent i at time t	kW
SEV_i	annual cost of agent i according to stratified expected value (unnormalised)	£
$\overline{SEV}_i$	annual cost of agent i according to normalised stratified expected value	£
M	number of samples of marginal contributions per agent (for adaptive sampling)	–
m	for samples	–
$\pi_{i,j}(m)$	probability of choosing stratum j for agent i at sample m	–
$\epsilon(m)$	sigmoid function	–
β and γ	parameters for sigmoid function	–
$\hat{\mu}_{i,j}$	estimated expected marginal contribution of agent i at stratum j in adaptive sampling	–
$\hat{\sigma}_{i,j}$	estimated standard deviation expected marginal contribution of stratum j for agent i	–
$h_{i,j}$	count of agent i sampling from stratum j in adaptive sampling	–
$m2_{i,j}$	sum of squared differences from mean of stratum j for agent i in adaptive sampling	–
mc	sampled marginal contribution in adaptive sampling	–
Δ	difference between the sampled value and the mean in adaptive sampling	–
RL_i	annual cost of agent i according to adaptive sampling	£
K	number of classes of unique demands in the community	–
k	for classes (clusters)	–
N_k	number of agents that belongs to class k	–
$P()$	multivariate hypergeometric distribution	–
$\tilde{d}_k(t)$	L2-normalised power demand of class k at time t	–
$\hat{\phi}_k$	cost redistributed to class k by certain redistribution method ($\overline{MC}_k$, $\overline{SEV}_k$, or RL_k)	£
$RD_\phi(\hat{\phi}_k)$	relative difference of a redistributed cost to the Shapley value for class k	%

Symbol	Definition	Unit
$RD_{\phi}(\hat{\phi})$	average relative difference of a redistribution method to the Shapley value	%

List of Figures

List of Tables

1

Introduction

The present-day centralised energy generation methods play a critical role in the current climate crisis. As per the UN [37], coal, oil, and gas are the leading contributors to overall carbon emissions (over 90%). This is why a number of government bodies, such as the European Union [16] and the United Kingdom [23], are providing supportive regulations to encourage energy consumers to invest in renewable energy sources (RESs), such as solar panels and wind turbines. Such incentives gave rise to *prosumers*: consumers with their own local RESs. The shift to the use of these RESs means that prosumers satisfy more of their own energy needs from the local sources and rely less on centralised energy generation. However, the main downside of RESs is that energy production is highly affected by its intermittent natural phenomena (e.g., wind speed, solar irradiance), and hence the energy cannot be dispatched on demand. For that reason, prosumers are often equipped with an energy storage system [42], allowing them to charge the battery with the RES or discharge the battery to meet their demands. Nevertheless, if the local energy assets do not meet their energy demands, the prosumers would need to make up the deficit using energy from the utility grid at a higher cost. On the other hand, if the local energy sources generate more than the demand and storage capacity, the surplus is sold to the utility grid for a negligible return. For example, in the UK, the price utility suppliers buy power generated by local solar PV systems is significantly lower than the cost of importing power from these companies [57]. Therefore, any energy that is unused to meet the demand or charge the battery is considered a waste if sent to the utility grid. Thus, a local transactive energy system that allows consumers to use generated energy within the local community (and often within the same low-voltage (LV) network) can reduce such redundancy. Figure 1.1 depicts the electrical power flow of a single prosumer, which presents the relationships between its demands, assets and the utility grid.

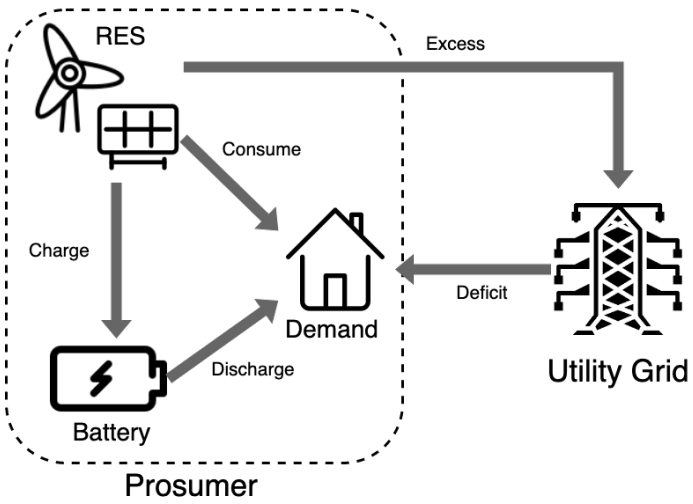
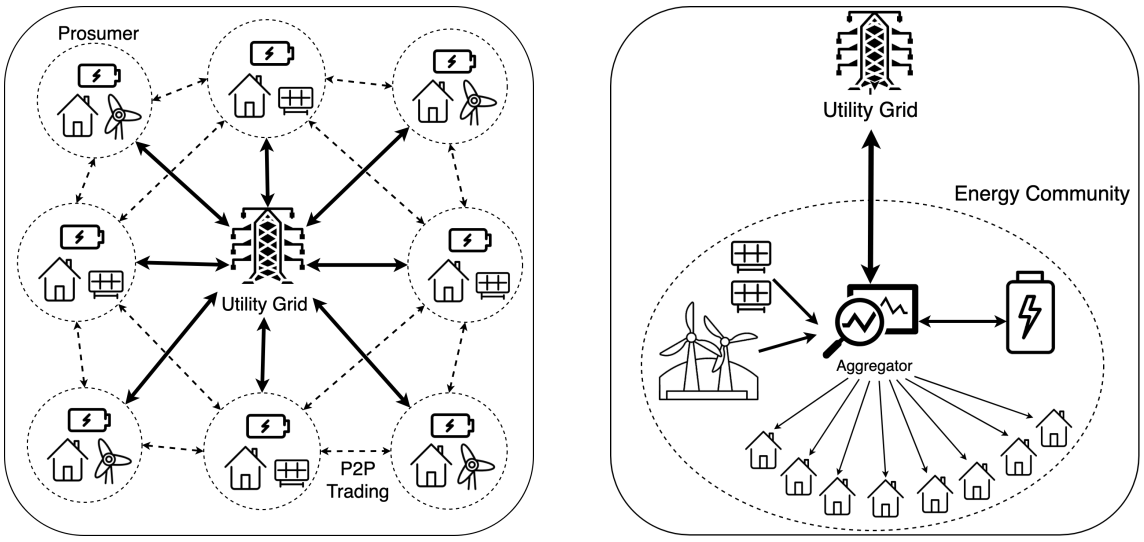


Figure 1.1: Power flow of a prosumer with a local energy source and battery storage, where the arrows indicate the direction.

In broad lines, there are two main models of organisation for local transactive energy systems [54]. One is peer-to-peer (P2P) energy trading systems, in which prosumers invest in their own energy assets and buy and sell energy with their neighbours directly, based on their individually-owned assets [56, 36, 46]. Another is the formation of *energy communities*, where prosumers group together and buy a shared generation resource (such as a community wind turbine) or a shared community battery. In the first scenario, each prosumer is metered separately and pays the value of its *net metered* electricity demand (demand after using its generation and storage capacity). In the second scenario, on the other hand, the whole community is *"behind the meter"*, i.e. pays for the net demand of the entire community over the billing period. The differences of two models are illustrated in Figure 1.2. The prosumers in Figure 1.2a each owns energy sources and a battery, and individually interacts with the central power grid in which the net demand is counted. It can be seen that power flow between a prosumer and the utility grid is bidirectional, and any excess generation by the prosumer is sold to the grid (also shown in Figure 1.1). Furthermore, a P2P trading scheme makes buying and selling energy to each other possible, which is represented by dotted arrows in Figure 1.2a. On the contrary, the energy community in Figure 1.2b presents a group of consumers sharing energy assets and interact with the utility grid as a single entity. Then, the net demand is also computed for the whole the community. These "behind the meter" models relies on a *community aggregator*, which control the energy assets and distribute the generated/discharge power to the households in the community. The aggregator is also in charge of receiving energy from the utility grid whenever there is a deficit and sending back energy to the grid when there is a surplus in the community.



(a) Prosumers with Individually Own Assets

(b) Energy Community with Shared Assets

Figure 1.2: Two different configurations of a community of prosumers, where (a) prosumers with their own energy assets are connected to the central grid individually, and (b) prosumers with jointly-owned assets interact with the grid as a whole through the aggregator.

This coalitional model around shared assets is increasingly popular, not just in academic research – for example, in Scotland, UK, Community Energy Scotland [1] identified 300+ energy communities that formed around a shared energy asset – typically a wind turbine, but similar examples exist all over the world. Such energy communities can consist of anywhere between several dozens to several hundred houses (e.g. a village or a city neighbourhood), often located on the LV network behind a local transformer. The prosumers share the outputs of jointly-owned energy assets, as well as the energy bill for the aggregate residual demand, i.e. the part of the demand not covered by the local generation and storage assets. Therefore, the community aggregator is not only responsible for the control and distribution of energy in the community but also for allocating any revenues from exporting energy and the bills of the residual demand. Clearly, one of the key challenges in this setting is the redistribution of such costs and benefits to the prosumers in a *fair* way.

[1] https://communityenergyscotland.org.uk/

Coalitional game theory has long studied such redistribution problems in a wide variety of systems [14]. A key concept is the Shapley value, first proposed by the Nobel prize-winning economist Lloyd Shapley [49]. The Shapley value has recently begun getting substantial attention in energy applications – with a rapid increase in the number of papers using Shapley value in energy systems in recent years. However, a key challenge with the Shapley value is that computing it is exponential in the number of agents, making exact computation intractable beyond a small number of agents.

The prior papers dealt with this in several ways. Most consider experimental models with up to a maximum of $\sim$10-20 agents to keep computations tractable. Another approach is to use some simpler heuristics for cost redistribution (e.g. Norbu et al. [39]), but it is not clear how close these are to the exact Shapley value.

Yet another approach is to use sampling. Sampling-based approaches do have merit, and in this study, we implemented the most advanced sampling-based method we are aware of, that of O'Brien, El Gamal, and Rajagopal [40], which uses reinforcement learning techniques to perform adaptive sampling to calculate Shapley. However, they also have disadvantages: for larger settings, a very large number of samples may be needed to get a reasonable approximation of the true Shapley value, which increases the computation cost considerably. Also, in community energy applications, sampling-based methods have the disadvantage that they may not produce a consistent result if they need to be rerun for verification purposes. Community energy schemes rely on the distributed trust of the prosumers in the community, and hence on the ability to sometimes rerun the calculations of the coalition coordinator if they wish. But, as the calculation at each run depends on which random samples are drawn, results will be slightly different, even on the same data. Hence, new methods to approximate the Shapley value in larger settings are clearly needed. In this study, we study both sampling-based and deterministic methods for approximating Shapley, compare their performance w.r.t. the true, exact Shapley value, and derive their computation costs. As one of the contributions of this study, we introduce a novel redistribution method that approximates the Shapley value well within polynomial time, and compare it to existing methods.

A key open challenge in this space remains determining the "ground truth", i.e. computing the exact, true Shapley value to compare other methods to, especially for larger realistically-sized communities (e.g. dozens to several hundred prosumers). Prior approaches, like O'Brien et al. [40] use a setting of only 20 agents as a "ground truth" to compute the exact Shapley, as they naturally find larger settings unfeasible to compute with unique agents. Yet, as we show in our experiments, an approximation method that does poorly for a small number of agents (e.g. 5-20) may actually do well for a realistically sized setting of 100-200 agents. Another important contribution of this study is that we develop a method to compute the *exact* Shapley value for larger communities, up to 200 agents. Intuitively, the core idea behind the method (see Chapter 5) is to cluster the agents in a much smaller number of consumption profiles, and use the symmetry of the combinations of agents to greatly reduce the cost of exact Shapley calculation.

Finally, as part of our contributions, we implemented our method in realistic community case studies, both in terms of demand, generation and battery data used and in terms of size (up to 200 households), granularity and duration (half-hourly data over a whole year). We used two different datasets, both containing household energy consumption data in the UK and the corresponding wind generation and battery data. One draws data from the Thames Valley vision trial, while the other draws data from the Low Carbon London project. This provides a highly realistic case study to provide confidence in the robustness of our experimental comparison results.

The rest of the thesis is organised as follows. First, Chapter 2 discusses related works on energy systems and Shapley value, whereas Chapter 3 presents the community energy model used. Chapter 4 introduces the Shapley value and different approximation methods, which Chapter 5 presents the method for exact Shapley computation for a fixed number of demand profile classes. Next, in Chapter 6, we provide the analytical comparison of the computation complexity of the different methods. Chapter 7 describes the dataset as well as the clustering of consumers for demand profiles in the community used in the experiments, while Chapter 8 presents the experimental comparison across a number of scenarios. Finally, Chapter 9 rounds off this thesis with a conclusion and insights into future works.

2

Related Work

This chapter gives an overview of the related work and an understanding of the current standard within the field of energy communities. Section 2.1 describes past and current work on energy communities with joint assets. Then, Section 2.2 describes studies on the use of Shapley value as a redistribution scheme in energy systems. Next, Section 2.3 explains current Shapley approximation in a wide range of application and its limitations. Finally, Section 2.4 describes studies on Shapley-approximated cost redistribution methods in energy communities that are the closest to this study.

2.1. Energy Communities with Shared Assets

Communities where consumers share energy assets have become increasingly popular, and examples can be seen all over the world. Feldhoff [21] describes a case study of Iida, a city located in Japan, where the city invests in community PV panels for local consumption to meet its goal of reducing greenhouse gas emissions. Veelen [58] mentions energy community projects in Scotland, such as Dingwall Wind Energy and Edinburgh Community Solar, where consumers jointly invest in RESs. Energy communities are especially popular in Europe, nesting around 3500 communities in which households jointly invest in RESs and share the output [11]. Energy generation sources are not the only shared assets. A case study in Japan by Li et al. [30] shows that a community-owned battery energy storage system could significantly reduce its capacity compared to the total capacity of individually owned batteries, making it more financially feasible for consumers to invest in such assets. Syed, Hansen, and Morrison [55] looks into the energy performances of an apartment building fitted with rooftop PV panels and a shared battery in Australia. Their study shows that the use of shared energy assets can help such apartments achieve a high level of self-sufficiency and hence reduce the cost of importing energy from the utility supplier. With the rise of such communities, fair allocation of benefits and costs within energy communities has attracted research, which can be seen in the upcoming sections of this chapter.

2.2. Shapley Value in Energy Systems

As the interest in energy communities grows, a notable number of recent papers consider using the Shapley value as an underlying redistribution method. Chiş and Koivunen [15] propose a coalitional cost-game optimization of a portfolio of energy assets using Shapley value as the underlying redistribution method, modelling a realistic case study of 9 households. Safdarian et al. [48] use the Shapley value for coalition-based value sharing in energy communities, modelling an energy community in southern Finland with up to 24 apartments. Vespermann, Hamacher, and Kazempour [59] study the market design of a local energy community and consider a number of solution concepts such as the nucleolus and Shapley values. Their numerical simulations study communities ranging in size from 4 up to 16 prosumers. Fleischhacker et al. [22] look at the Shapley value and Nash Bargaining solution as means of redistributing the energy costs of shared solar PV panels and battery storage, and perform experiments on a community of 4 prosumers. Robu et al. [47] consider a cooperative coalitional game for energy group buying. While they discuss Shapley value as a solution concept, their focus is on other coalition properties. Singh et al. [52] present the use of Shapley value for energy trading among microgrids, using a case study of 3 microgrids. Zhang et al. [63] consider the use of Shapley value to divide

gains in alliances among retailers in the Chinese energy settlement market, considering alliances up to a size of 9 agents. Besides the above, applications of Shapley value in energy systems have been seen in loss allocation in power distribution systems [50, 51, 3, 44, 4], distribution of profit from demand response [40, 34], and distribution of cost from congestion management [53, 61, 60]. All these recent papers show the potential of the Shapley value in energy applications and make valuable contributions to the literature. However, their numerical case studies consider only a small number of agents (up to $\sim$20), to keep the computation of the exact Shapley value tractable. Yet, realistically sized energy communities have more members, e.g., there are usually 50-200 consumers behind a substation/LV transformer in Europe [45], or potentially even more sharing an asset such as a large community wind turbine. Hence, new methods to approximate the Shapley value in larger settings are clearly needed.

2.3. Shapley Value Approximation

Due to its high computation cost, the Shapley value has received strong interest in efficient approximation methods since its introduction. Currently, many approximation methods compute the expected marginal contribution of an agent to the *sampled* coalitions, initially suggested by Mann and Shapley [35]. Furthermore, the seminal work of Castro et al. [12, 13] first proposes a polynomial calculation method which highlights the concept of *stratified sampling*, which has been refined in other works [2, 24], and is a key concept in the method we develop as well. Many recent works also provide theoretical error bound of sampling-based approximation methods [31, 33, 5, 6, 10].

One of the obstacles to approximating the Shapley value in a general case is that a fully polynomial-time randomized approximation scheme (FPRAS) is the best it can do [31], and so the approximation error can be bounded to the error term ϵ with a certain probability. Furthermore, there does not exist a deterministic Shapley value approximation method that is a fully polynomial-time approximation scheme (FPTAS) [31]. Yet, deterministic methods have desirable characteristics for cost redistributions of consumers. Such methods produce the same results after every run given the same inputs, allowing consumers to verify the calculated cost themselves, in contrast to random sampling methods where the redistributed cost can differ depending on the samples drawn. Furthermore, deterministic methods would also guarantee the same cost redistributed to consumers with the exact same demand profile. Such properties can provide consumers with additional trust on the model. Bhagat et al. [8] provides a deterministic Shapley approximation method to their newly proposed budgeted games. Their method is theoretically proven to approximate the Shapley value with a constant additive error by replacing the value function with a relaxed function. However, theoretical analyses of deterministic methods are especially difficult in many real-world energy applications, in which the cost function results from a control procedure over the energy assets over a long time horizon rather than in closed form. Hence, a realistically-sized empirical analysis would be crucial to evaluate the performances of the approximation methods. For example, the work of Voswinkel et al. [60] proposes a deterministic Shapley value approximation method for redistributing congestion costs by exploiting the problem-specific properties. The empirical results show that it outperformed the sampling method while having a smaller computation cost.

2.4. Shapley Value Approximation in Energy Communities

In this section, studies that look at Shapley approximation methods as the underlying redistribution scheme in energy system communities are described.

O'Brien, El Gamal, and Rajagopal [40] study redistribution of profit in a demand response program, where a group of consumers work together to alter their energy consumption behaviours to complement the supplied energy. They propose an enhancement of the method first outlined by Castro, Gómez, and Tejada [12], that uses reinforcement learning to do the stratified sampling in an adaptive way. Their method is shown to approximate the Shapley value better than random sampling and stratified sampling methods with the same number of samples, proving that it is a more sample- (and time-) efficient Shapley approximation method. Their algorithm is implemented in this study to compare with the newly proposed method, and the details of the algorithm are explained in Section 4.2.3. However, a key limitation of the prior work of O'Brien et al. is that they still use a comparison benchmark of only 20 agents, while we develop a way to compute it exactly for much larger communities. Moreover, we wanted to develop and test some deterministic methods of Shapley value approximation that do not depend on the number of samples and can be reproduced to give the same result.

Similarly, Bakr and Cranefield [7] look at Shapley value approximations for redistributing benefits in demand response programs. They provide experimental comparisons between a linear approximation method and stratified random sampling on their accuracies and scalabilities. They use a community of 10 consumers to see how well these methods approximate the true Shapley value, showing the stratified sampling method approximates the Shapley value with minimal error. The approximation methods are also tested on larger communities of up to 1000 consumers to evaluate their computation time, but no comparison of redistributed values with the true Shapley values is made in such a scenario.

Furthermore, Han, Morstyn, and McCulloch [24] propose a modified stratified sampling method, similarly to O'Brien et al., for the allocation of profits in energy communities. Their method takes a two-stage approach where in the first stage, samples are partitioned equally among the strata, and the expected marginal contributions and the variances of the strata are computed. In the second stage, the remaining samples are assigned according to the previous step, and more samples are given to strata with larger variances. Additionally, the algorithm stops sampling earlier than the maximum number of samples determined by the user when the (estimated) standard error to the true Shapley is below a certain threshold. Han et al. evaluate their method on a community with 14 prosumers and show a low approximation error to the true Shapley values. They also evaluate the scalability of their method on large communities of 100 prosumers, but again, no comparison with true Shapley is made for this scenario.

Finally, the work of Norbu et al. [39] provides a heuristic battery control algorithm that efficiently maximises the use of locally generated energy while also considering the battery degradation effect. The battery control algorithm was tested on both prosumers with individually owned assets and on a community with sharing assets (RESs and a battery). Their results indicate that investing in joint assets could reduce the total energy cost, providing economic benefit for consumers to join such communities. Their work also provides a highly efficient benefit redistribution method based on the marginal contribution principles, which is shown to have a fairer allocation than the ones often used in practice for these communities. Yet, they do not make any comparison to the Shapley value, and hence the level of fairness of this method is not clearly known. However, with their support, we use the same battery control algorithm and demand/generation dataset of a community of 200 prosumers in the UK, as it provides a realistic experimental case study to test the methods we develop. Additionally, we also look at a different dataset with a larger number of households to further provide confidence within our methods.

3

Community Energy Model

This chapter describes the setup of the community energy model used for this study. Section 3.1 explains the structure of the community, whereas Section 3.2 presents the battery control model. Next, the battery degradation model is shown in Section 3.3 . Finally, Section 3.4 describes how the energy cost of the community is derived.

3.1. Community Model

Consider an energy community $\mathcal{N}$ consisting of a set of $|\mathcal{N}| = N$ prosumers, a shared battery and renewable energy source (RES). In this study, a lithium-ion battery and Enercon E-33 wind turbines [19] with a rated power of 330 kW were considered as the community's energy storage system and RES, respectively. Each prosumer in the community has a half-hourly power demand profile represented as $d_i(t)$ for the power demand of agent i at time step t. The final time step of the operation of the system is denoted as T. In this study, the data consists of half-hourly demands and generation during a 1 year period, and hence $T = 365 \times 48 = 17520$.

The demand of the community at time t, $d_{\mathcal{N}}(t)$, is simply the sum of the demands of the agents in the community at t, described as the following.

$$d_{\mathcal{N}}(t) = \sum_{i \in \mathcal{N}} d_i(t), \quad \forall t \in \{1, ..., T\} \tag{3.1}$$

Furthermore, a community has generated power, $g(t)$, by the jointly owned local renewable energy generation, and the power of the battery, $p^{\text{bat}}(t)$. The battery is considered charging when $p^{\text{bat}}(t)$ is negative and discharging when $p^{\text{bat}}(t)$ is positive. Finally, a community is required to buy power from the utility grid if the community assets do not provide enough power for the demand. If there is a surplus of power, on the other hand, a community can sell excess power to the grid. The power of the utility grid is denoted as $p^{\text{grid}}(t)$, where the value is positive when power is bought from the grid and negative when power is sold to the grid.

Given these variables, the following constraint needs to be satisfied at every time step.

$$d_{\mathcal{N}}(t) = p^{\text{grid}}(t) + p^{\text{bat}}(t) + g(t), \quad \forall t \in \{1, ..., T\} \tag{3.2}$$

The constraint assures that the community power demand is met from the power sources. Additionally, when the generation is greater than the demand, all of the energy from the excess power is stored in the battery and or sold to the utility grid.

3.2. Battery Control Algorithm

The use of the battery was regulated at each time point using the heuristic-based battery control algorithm from Norbu et al. [39]. The algorithm decides whether to charge or discharge the community battery, and whether to sell or buy energy from the grid, according to the community power demand, community power generation, the battery power, and the battery's state of charge (SoC). Battery's SoC level (in %) signifies how much of its capacity is charged.

The battery keeps track of its SoC level and the power of the battery, $p^{\text{bat}}(t)$. The power of the battery is negative when charging and positive when discharging. While the battery is charging, the SoC level needs to remain below or equal to the maximum battery capacity, SoC^{max}. In addition, the magnitude of $p^{\text{bat}}(t)$ cannot exceed the maximum (dis)charging power of the battery, $p^{\text{bat, max}}$. These constraints are expressed as the followings.

$$SoC(t) \leq SoC^{\text{max}} \tag{3.3}$$

$$\left|p^{\text{bat}}(t)\right| \leq p^{\text{bat, max}} \tag{3.4}$$

When discharging the battery, similar constraints apply. First, the SoC level cannot go below the minimum battery capacity, SoC^{min}. Second, the magnitude of the battery power may not exceed the maximum discharging power. These constraints are represented as the following.

$$SoC(t) \geq SoC^{\text{min}} \tag{3.5}$$

$$\left|p^{\text{bat}}(t)\right| \leq p^{\text{bat, max}} \tag{3.6}$$

The heuristic-based battery control algorithm is described as the following. When the generated power from the RES is greater than the demand ($g(t) > d(t)$), the excess power can be used to charge the battery. However, if the battery is reaching its full capacity or the power exceeds the maximum charging power $p^{\text{bat, max}}$, not all the energy can be stored in the battery, and the surplus power will be sold to the utility grid. The updated $p^{\text{bat}}(t)$, the SoC level, and the exported energy to the utility grid ($e^s(t)$) are determined as the following:

$$p^{\text{bat}}(t) = -\min\left(\min((g(t)-d(t)), p^{\text{bat, max}}), \frac{SoC^{\text{max}} - SoC(t-1)}{\eta^c \Delta t}\right) \tag{3.7}$$

$$SoC(t) = SoC(t-1) - \eta^c p^{\text{bat}}(t)\Delta t \tag{3.8}$$

$$e^s(t) = (g(t) - d(t) + p^{\text{bat}}(t))\Delta t \tag{3.9}$$

where η^c is the charging efficiency and Δt is the duration of time step t in hours. The profit from exporting the energy to the grid can be expressed as the product of the energy exported, $e^s(t)$, and the export tariff, $\tau^s(t)$.

Similarly, if the demand is greater than the generated power from RES ($g(t) < d(t)$), the battery is discharged to meet the demand. If the remaining energy in the battery is not enough to meet the community's demand or if the battery's maximum power constraint restricts from discharging enough power, energy needs to be imported from the utility grid, denoted as $e^b(t)$. The followings are the updated $p^{\text{bat}}(t)$, SoC levels, and $e^b(t)$ in case of shortage of power:

$$p^{\text{bat}}(t) = \min\left(\min((d(t)-g(t)), p^{\text{bat, max}}), \frac{\eta^d}{\Delta t}(SoC(t-1) - SoC^{\text{min}})\right) \tag{3.10}$$

$$SoC(t) = SoC(t-1) - \frac{p^{\text{bat}}(t)}{\eta^d}\Delta t \tag{3.11}$$

$$e^b(t) = (d(t) - g(t) - p^{\text{bat}}(t))\Delta t \tag{3.12}$$

where η^d is the discharging efficiency. Again, the cost of importing energy from the grid at time t is the product of imported energy, $e^b(t)$, and the import tariff, $\tau^b(t)$.

Note that, in the heuristic-based battery control algorithm above, we considered *flat* import/export tariffs, in which the price remains the same throughout the time period of the operation. Moreover, importing or exporting energy to the grid is always worse price-wise than consuming/storing it locally, when possible. This is a realistic assumption in the current climate, when import prices are high, and so-called feed-in tariffs (i.e. tariffs paid to very small renewable generators) are being phased out. It is possible to have more advanced control heuristics in case of dynamic or time-of-use prices from the grid that include, e.g. a price prediction component. However, the Shapley computation methods proposed in this study can also be combined with more complex control cases. This is because the methods we develop apply to the overall cost function, working to minimise the times of iterations needed to recompute it - but are independent of how the control is performed.

3.3. Battery Degradation Model

Acceleration of battery degradation caused by frequent charging and discharging operations as well as deep discharging may shorten the battery lifetime to be less than the one specified by the manufacturer. With shortened lifetime, the community needs to replace the battery earlier, incurring additional costs to the households. Hence, the battery degradation model takes this factor into account when calculating the annual cost of the battery for a more accurate representation of the real-world simulation. In this section, the battery degradation model used in this study and developed by Norbu et al. [39] is explained.

The number of cycles and the depth of discharge (DoD) influences battery degradation. A full cycle is defined as the SoC returning to the starting value after a discharging and charging phase. On the other hand, a half cycle is defined to be merely one of the the charging or discharging phase. Furthermore, a cycle can be classified as regular or irregular. A regular cycle starts the cycle with the SoC of 100%, whereas an irregular cycle has the starting SoC to be other than 100%. Although regular and irregular cycles can have the same DoD (for example, a regular cycle of SoC 100% discharged to 50% and charged to 100%, and an irregular cycle of SoC starting at 80%, discharged to 30% and charged to 80% both have a DoD of 50%), the battery is depreciated differently. Norbu et al. [39] uses the rainflow cycle counting algorithm [18] to count the number of full and half cycles as well as identify whether the cycle is regular or irregular, which is also applied in this study.

By counting the number of cycles during the time period, the depreciation factor (DF) of the battery is computed to estimate the battery useful lifetime. As mentioned before, regular and irregular cycles influence the depreciation factor differently. Hence, DF can be defined as the following.

$$\mathsf{DF} = \mathsf{DF}^{\text{regular}} + \mathsf{DF}^{\text{irregular}} \tag{3.13}$$

where $\mathsf{DF}^{\text{regular}}$ and $\mathsf{DF}^{\text{irregular}}$ are the depreciation factors of regular and irregular cycles respectively. The depreciation factor of regular cycles is determined as follows,

$$\mathsf{DF}^{\text{regular}} = \sum_{DoD=0\%}^{100\%} \frac{n_{cycles}^{\mathsf{DoD,regular}}}{N_{cycles}^{\mathsf{DoD,max}}} \tag{3.14}$$

where $n_{cycles}^{\mathsf{DoD,regular}}$ is the number of regular cycles at a certain DoD value during the time period, and $N_{cycles}^{\mathsf{DoD,max}}$ is the lifetime of the battery in number of cycles for that DoD value given by the manufacturer. This study used the battery cycle life data of a lithium battery from the work of Xu et al. [62]. The depreciation factor of irregular cycles is expressed as the following.

$$\mathsf{DF}^{\text{irregular}} = \sum_{l \in L} \mathsf{n}_l \times \left| \frac{1}{N_{cycles}^{\mathsf{DoD}^{eq}(SoC_l^{Start}),max}} - \frac{1}{N_{cycles}^{\mathsf{DoD}^{eq}(SoC_l^{End}),max}} \right| \tag{3.15}$$

where L is the set of all irregular cycles, and SoC_l^{Start} and SoC_l^{End} are cycle l's starting and ending SoC levels, respectively. The value of n_l is determined by whether the cycle l is full or half, defined as the following.

$$\mathsf{n}_l = \begin{cases} \frac{1}{2}, & \text{if } l \text{ is a half cycle} \\ 1, & \text{if } l \text{ is a full cycle} \end{cases} \tag{3.16}$$

Finally, $N_{cycles}^{\mathsf{DoD}^{eq}(SoC_l^{Start}),max}$ is the lifetime in number of cycles for $\mathsf{DoD}^{eq}(SoC_l^{Start})$, a DoD of a cycle equivalent to starting at 100% SoC and ending at the value of SoC_l^{Start}. $\mathsf{DoD}^{eq}(SoC_l^{Start})$ is computed as the following.

$$\mathsf{DoD}^{eq}(SoC_l^{Start}) = 100 - \left(\frac{SoC_l^{Start}}{SoC^{\mathsf{max}}} \times 100 \right) \tag{3.17}$$

$\mathsf{DoD}^{eq}(SoC_l^{End})$ and $N_{cycles}^{\mathsf{DoD}^{eq}(SoC_l^{End}),max}$ are similarly computed.

DF resulting from the former calculation is used in Section 3.4 for computing the community battery cost during the time period.

3.4. Community Cost Calculation

A cost function is a key attribute of a coalitional game. In this section, energy cost calculation of the community (or any subset of prosumers) is explained. The community cost calculation in this study can be seen as the cost function, and it is required for redistribution methods described in Chapter 4. Here, we assume that the energy cost is subadditive; the total cost of the community is the lowest with the grand coalition, and hence optimal. This is an important assumption that provides incentives for prosumers to join the energy community, and not deviate from the community to form a smaller coalition.

The community energy cost is composed of three components. The first is the cost of energy bought from the grid, subtracted by the revenue of energy sold to the grid during the time period. In this study, the power demand and generation of the community are available every half hour. Hence, the energy cost and revenue of the community are also available every half hour. The energy bought and sold at each time point, $e^b(t)$ and $e^s(t)$ respectively, are determined by the battery control algorithm explained in Section 3.2. The cost $c_T^{\text{grid}}(\mathcal{N})$ is computed as the following.

$$c_T^{\text{grid}}(\mathcal{N}) = \sum_{t=1}^{T} e^b(t)\tau^b(t) - \sum_{t=1}^{T} e^s(t)\tau^s(t) \tag{3.18}$$

The second component of the cost is the wind turbine, $c_T^{\text{wind}}(\mathcal{N})$. The annual cost is calculated as the following.

$$c_T^{\text{wind}}(\mathcal{N}) = \frac{\text{WT generation capacity} * \text{cost per kW}}{\text{Lifetime (in years)}} \tag{3.19}$$

The wind turbine generation capacity is calculated as the maximum receiving power from the wind turbine in one time step. The receiving power from the wind turbine was chosen to be $0.006 \times N$ times the wind turbine output in this study. The maximum capacity of the wind turbine increases linearly with the number of prosumers in the community, and hence the cost also increases linearly with the size. The cost of the wind turbine was set to 1072 £(GB pounds)/kW and lifetime to 20 years, which is realistic for current technologies in the UK market [39].

The final component of the cost is the battery. The cost of the battery, $c_T^{\text{bat}}(\mathcal{N})$ is computed as the following.

$$c_T^{\text{bat}}(\mathcal{N}) = \frac{\text{battery capacity} * \text{cost per kWh}}{\max\left(\text{ Lifetime (in years)}\ , \frac{1}{\text{DF}}\right)} \tag{3.20}$$

In this study, the battery capacity is set to be $5 \times N$ kWh. Similarly to the wind turbine, the battery capacity increases linearly with the community size, and therefore the community battery cost also increases linearly with the community size. The cost of battery per kWh was set to 150 £/kWh and the lifetime of the battery to 20 years. The variable DF is the depreciation factor of the battery determined by the battery degradation model from Section 3.3. Although the battery is given a lifetime, the lifetime can be shortened or additional maintenance costs may be required depending on the number of charge cycles and depth of discharge (DoD). Hence, using a battery degradation model can give a better assessment of the annual battery cost.

The total cost of the community, $c_T(\mathcal{N})$ is the sum of the three components, which is the following.

$$c_T(\mathcal{N}) = c_T^{\text{grid}}(\mathcal{N}) + c_T^{\text{wind}}(\mathcal{N}) + c_T^{\text{bat}}(\mathcal{N}) \tag{3.21}$$

The community cost can be computed for any subset of agents, and thus the cost contribution of an agent to a group can be determined by comparing the cost of the group with and without the agent. Specifically, every agent in the group contributes equally to the cost of the wind turbine and the battery (from Equations (3.19) and (3.20)), but it does not mean the usage of the assets are equal among agents. For example, agents with demand profiles that are well-aligned to the energy generation of the wind turbine will make better use of the community generation assets, resulting in requiring less imported energy from the utility grid to match their demand. On the other hand, agents with demand profiles that are poorly aligned with the generation will put greater pressure on the community battery capacity and equivalently cause more energy to be imported. Therefore, the *marginal value* with which each prosumer causes the total cost to rise is a key factor to consider.

The community energy cost calculation can be seen as a cost function for a set of prosumers with demands. The notation of the community cost is simplified to $c(\mathcal{N})$ w.l.o.g., because time horizon $T = 1$ year is used to compute costs in the rest of the study.

4

Redistribution Methods

The cost redistribution methods for the energy community is described in this chapter. First, Section 4.1 describes the formal definition of the Shapley value. Then, Section 4.2 presents three Shapley value approximation methods used to redistribute the cost of the energy community in the case studies.

4.1. Exact Shapley Value

The redistribution of costs or benefits in a game using the Shapley values is considered to be fair [49, 9]. The cost of prosumer i according to the Shapley value, ϕ_i, is computed as the following.

$$\phi_i = \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{i\}} \frac{|\mathcal{S}|!(N - |\mathcal{S}| - 1)!}{N!} (c(\mathcal{S} \cup \{i\}) - c(\mathcal{S})) \tag{4.1}$$

The marginal contribution of prosumer i to the subcoalition of prosumers $\mathcal{S}$, denoted as $c(\mathcal{S} \cup \{i\}) - c(\mathcal{S})$, is how much the prosumer adds to the cost by joining the subcoalition. Then, the Shapley value of agent i can be seen as the mean marginal contribution of i for all possible subcoalitions in the community and all possible permutations of these subcoalitions.

An alternative way to write the Shapley equation that is particularly useful for our approach is through using the concept of *stratum*, given as the following.

$$\phi_i = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{\substack{\mathcal{S} \subseteq \mathcal{N} \setminus \{i\}, \\ |\mathcal{S}| = j}} \frac{j!(N - 1 - j)!}{(N - 1)!} (c(\mathcal{S} \cup \{i\}) - c(\mathcal{S})) \tag{4.2}$$

It can be seen that the marginal contribution of agent i to a subcoalition $\mathcal{S}$ is computed as in Equation (4.1). Then, the marginal contribution is multiplied by the relative frequency of $\mathcal{S}$ in the stratum. A stratum j is a set of all possible subcoalition $\mathcal{S}$ with $|\mathcal{S}| = j$. From this, the expected marginal contribution of agent i to a stratum 0 (empty subcoalition) up to stratum $N - 1$ (subcoalition of the whole community except i) can be computed. Then, the Shapley value of agent i is equivalent to averaged expected marginal contributions over the strata.

Yet, computing Shapley exactly from these equations has a very large time complexity (exponential to the number of agents in the community, as the marginal contributions to every subcoalition of the community is needed), which makes it intractable very quickly as the community size increases.

4.2. Approximation Methods

Given the Shapley value definition above, the remaining chapter describes three cost redistribution methods that approximate the Shapley value in an energy community setting.

4.2.1. Last Marginal Contribution

While computing Shapley value directly requires exponential number of steps, it is possible to use the marginal contribution principle to design a much simpler scheme that considers the marginal contribution of each agent w.r.t. the other $N - 1$. Formally, let the cost of an agent i in the community $\mathcal{N}$ be

simply the marginal contribution of agent i to the rest of the community, defined as the following.

$$MC_i = c(\mathcal{N}) - c(\mathcal{N} \setminus \{i\}) \tag{4.3}$$

The annual energy cost MC_i of agent i uses the same intuition as Equation (4.1) in Shapley value calculation. But, whereas the Shapley value takes the mean marginal contribution of agent i for *every* possible subcoalition in the community, this method computes the cost by only looking at the *last* marginal contribution, making it a much more time-efficient method. However, costs based on the last marginal contributions do not hold the same property of the Shapley values in which the sum of individual cost is equivalent to the total community cost [49]. Hence, the last marginal cost needs to be normalised. The final redistributed cost $\overline{MC}_i$ of agent i according to the normalised last marginal contribution (simply the marginal contribution method from now on) is given as:

$$\overline{MC}_i = c(\mathcal{N}) \frac{MC_i}{\sum_{q \in \mathcal{N}} MC_q} \tag{4.4}$$

The time complexity of the marginal contribution method is $\mathcal{O}(N)$, so, while simple, it is a very computationally efficient method.

4.2.2. Stratified Expected Values

The last marginal contribution method only takes into account the marginal contribution of the last stratum. Starting from this observation, we propose a novel Shapley redistribution scheme that goes a step further and considers the expected marginal contribution for *every stratum*, while still avoiding the huge combinatorial cost of the exact Shapley method. We call this the *stratified expected values* method.

Formally, for agent i, an agent profile p_{-i} that has average energy demands from the rest of the agents in the community is created. The demand of the agent profile p_{-i} at time t is calculated as:

$$d_{p_{-i}}(t) = \frac{\sum_{q \in \mathcal{N} \setminus \{i\}} d_q(t)}{N-1}, \quad \forall t \in \{1, ..., T\} \tag{4.5}$$

The main idea of the method is that since p_{-i} has the average demand of the rest of the community for every time step, computing the marginal contribution from a set of agents with such a demand profile can approximate the *expected* marginal contribution of that stratum. Since the Shapley value can also be seen as the mean of expected marginal contribution of every stratum, taking the mean of approximated marginal contribution of every stratum should give an average "in expectation" value that approximates the Shapley value. Hence, the cost of agent i based on the stratified expected values method SEV_i is calculated as the following.

$$SEV_i = \frac{1}{N} \sum_{j=0}^{N-1} c(\{1, ..., j\} \cup \{i\}) - c(\{1, ..., j\}), \quad \text{such that } d_1 = ... = d_j = d_{p_{-i}} \tag{4.6}$$

Similarly to the marginal contribution method, the sum of individual energy costs does not equal the community's total cost since this method uses fictitious agents with demand profiles $d_{p_{-i}}$. Hence, a normalisation step is required, given as follows.

$$\overline{SEV}_i = c(\mathcal{N}) \frac{SEV_i}{\sum_{q \in \mathcal{N}} SEV_q} \tag{4.7}$$

The time complexity of computing the individual costs with this method is $\mathcal{O}(N^2)$ since for each agent, it requires to calculate the average marginal contribution once for every stratum, ranging from 0 to $N-1$. While this is obviously more than the $\mathcal{O}(N)$ computation of the last marginal value method, it is still much less than the exponential cost of computing the Shapley values, and still very tractable for medium and relatively large community sizes, while potentially giving us a much better approximation.

4.2.3. Adaptive Sampling with Reinforcement Learning

Here, the sampling Shapley approximation using reinforcement learning adapted from the work by O'Brien, El Gamal, and Rajagopal [40] is described. Given the total number of samples per agent (predetermined

by the user), M, this redistribution method will sample marginal contributions of an agent randomly, and adjust the number of samples per stratum while running. Specifically, stratum j being selected for agent i's m-th sample $\pi_{i,j}(m)$ is given as the following.

$$\pi_{i,j}(m) = \frac{\epsilon(m)}{N} + (1 - \epsilon(m)) \frac{\hat{\sigma}_{i,j}}{\sum_{s=0}^{N-1} \hat{\sigma}_{i,s}} \tag{4.8}$$

where $\hat{\sigma}_{i,j}$ is the estimated standard deviation of the marginal contributions in stratum j, and $\epsilon(m)$ is a double sigmoid function which helps exploration at the beginning (small m value) and exploitation near the end (large m), defined as the following.

$$\epsilon(m) = 1 + \frac{1}{1 + e^{\frac{\gamma}{\beta}}} - \frac{1}{1 + e^{-\frac{m - \gamma M}{\beta M}}} \tag{4.9}$$

The parameters β and γ were set to 0.075 and 0.2 respectively during the experiment since it was found that the sigmoid function with these parameter setting to approximate ideal sampling well [40]. To approximate the Shapley value well, more samples may be required from certain strata in which the marginal contribution values can vary highly. On the other hand, if the marginal contributions are similar within a stratum, such strata would not need large samples to be approximated. Equation (4.8) helps to distribute the samples in such a way. The workings of the algorithm is described below.

For each agent i, the estimated expected marginal contributions of stratum j, $\hat{\mu}_{i,j}$ is initialized to 0. Similarly, the count of how many times stratum j was visited, $h_{i,j}$, and the sum of squared differences from the current mean of stratum j, $m2_{i,j}$, are initialized to 0. Finally, the estimated standard deviation of the marginal contributions of stratum j, $\hat{\sigma}_{i,j}$ is initialized to a very large value (here, 10000). This is repeated for all strata.

For every sample of agent i, the stratum j is selected according to the probabilities of each stratum given by Equation (4.8). A subcoalition $\mathcal{S}$ is chosen randomly from the selected stratum, and the marginal contribution of agent i to the subcoalition is computed as $mc = c(\mathcal{S} \cup \{i\}) - c(\mathcal{S})$. The difference between the sampled marginal contribution and the estimated expected marginal contribution, $\Delta = mc - \hat{\mu}_{i,j}$ is also calculated. Then, the variables are updated after each sample as followings.

$$h_{i,j} \leftarrow h_{i,j} + 1 \tag{4.10}$$

$$\hat{\mu}_{i,j} \leftarrow \hat{\mu}_{i,j} + \frac{\Delta}{h_{i,j}} \tag{4.11}$$

$$m2_{i,j} \leftarrow m2_{i,j} + \Delta(mc - \hat{\mu}_{i,j}) \tag{4.12}$$

Furthermore, $\hat{\sigma}_{i,j}$ is also updated if the stratum has been visited more than once by agent i, as the following.

$$\hat{\sigma}_{i,j} \leftarrow \sqrt{\frac{m2_{i,j}}{h_{i,j} - 1}} \tag{4.13}$$

Once all the variables are updated, it moves on to the next sample. After M samples are taken for agent i, the redistributed energy cost according to RL-based Shapley Approximation method, RL_i is calculated by taking the mean of estimated expected marginal contributions over the strata, i.e.,

$$RL_i = \frac{1}{N} \sum_{j=0}^{N-1} \hat{\mu}_{i,j} \tag{4.14}$$

A difference from the original work is that strata 0 and $N-1$ are only chosen once each in this study since no matter how many times these strata are sampled, they will always have the same marginal contributions (since there is only 1 possible subcoalition). By doing so, more samples can be used in different strata, allowing the algorithm to make use of samples slightly more efficiently.

The time complexity of this redistribution scheme is $\mathcal{O}(N \cdot M)$. Note that, in principle, when the community size increases, the number of samples required to approximate the Shapley value well increases faster, hence M is set to a value that is $M \gg N$. For this study, M was set to 1000 when running this method to assure multiple samples are taken from each stratum.

5

Exact Shapley Values for K Classes

Given the redistribution methods from Section 4.2, what is really needed is the "ground truth" consisting of the exact Shapley values, to compare the performance of these approximation methods for a realistic size community (e.g. $N = 200$ prosumers behind a transformer). Prior works that do this, like O'Brien, El Gamal, and Rajagopal [40], reduce the number of prosumers to $N = 20$ to compute the exact Shapley, but we argue this method is not really a satisfactory way to proceed. This is because, crucially, the quality of an approximation for a larger community (e.g. $N = 50, 100$ or 200 agents) can be very different than for a very small number of agents, up to 20 (we will clearly see this effect in our experiments as well).

The key intuition is that, while computing the Shapley values of N unique agents requires a time complexity that is exponential to N, the computation time can be significantly reduced if the community consists of a limited number of classes of agents, where agents in the same class are clustered in the same demand profile.

Let the new model be defined as the following. A community $\mathcal{N}$ still consists of N agents, with now K classes of demand profiles in the community such that every agent belongs to one class, and all the agents in the same class have equal half-hourly demands. We assume w.l.o.g. that classes are ordered by size, i.e.

$$N \geq N_1 \geq ... \geq N_K \geq 1 \tag{5.1}$$

where N_k is the size of the class k. Then, the number of all possible energy costs of subcoalition in the community is $(N_1 + 1) \times ... \times (N_K + 1)$, since from each class k you can have 0 to N_k agents being part of the subcoalition. This is important to note because computing the annual costs of subcoalitions (the cost function) is the most computationally expensive part of computing the redistributed costs, since it has to run through one year of half-hourly demands datapoints every time the cost function is called. The energy costs of every subcoalition may be used multiple times to compute the marginal contribution in Shapley calculation, hence storing the values in a table of the dimension $(N_1 + 1) \times ... \times (N_K + 1)$ can be time-saving. Algorithm 1 shows the creation of the table storing the costs of all possible subcoalitions in a community.

The table containing costs of every subcoalition can also be represented as a hyperrectangle of K dimensions and the size $N_1 \times ... \times N_K$. Each axis represents the number of agents in the class. A stratum can be represented in such a hyperrectangle as a hyperplane cutting through in which the sum of axes equals the size of the stratum. Hence, strata correspond to planes parallel to each other. Figure 5.1 shows an example case where $K = 3$ with $N_1 = 7$ (x-axis), $N_2 = 4$ (y-axis), and $N_3 = 2$ (z-axis). Stratum 5 is represented by the plane, where $x + y + z = 5$.

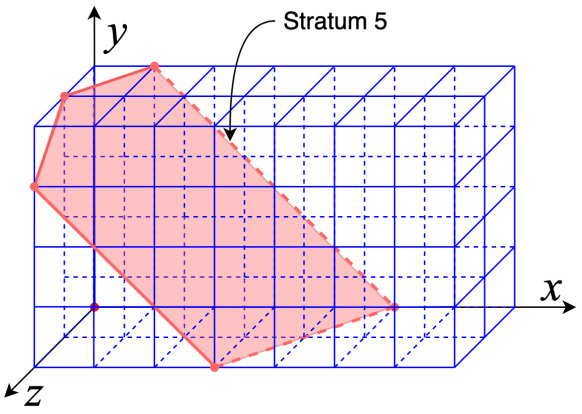


Figure 5.1: A representation of all possible subcoalitions in a community of $K = 3$ classes with $N_1 = 7$, $N_2 = 4$, and $N_3 = 2$.

Algorithm 1 Create a table containing energy costs of every possible combinations of number of classes

Input: Number of prosumers in the community, N. Number of classes, K. Number of prosumers in each class, $N_1, N_2, ..., N_K$ with $N_1 + N_2 + ... + N_K = N$ and $N_1 \geq N_2 \geq ... \geq, N_K \geq 1$. Demands of the classes, $d_1, d_2, ..., d_K$, where each d contains half hourly demands during time period of T (1 year).

Output: Table containing costs of all possible subcoalition combinations, CS.

1: **function** CreateTable_AllSubcoaltionCosts($N, N_1, ..., N_K, d_1, ..., d_K$)
2: **for** all $(n_1, n_2, ..., n_K) \in \prod_{k=1}^{K}\{0, 1, ..., N_k\}$ **do** ▷ Cartesian product
3: $\mathcal{S} = \bigcup_{k=1}^{K}\{1, ..., n_k\}$ with demands d_k ▷ Union of sets with $1, ..., n_k$ having same demand d_k
4: $CS[n_1, n_2, ..., n_K] = c(\mathcal{S})$ ▷ Cost of subcoalition, Eq. 3.21
5: **end for**
6: **return** CS
7: **end function**

Once the energy costs of all possible subcoalitions have been computed, the Shapley values of the agents, which is the energy cost the agent owes to the community, can be found. Because of the symmetry axiom [49], the Shapley values of agents in the same class (same energy demands) are equal, and hence it is only required to calculate the Shapley values once for each class. Algorithm 2 is used to determine the Shapley values when the community consists of K classes of agents. The algorithm first loops over the number of classes starting from $k = 2$ (Line 2, Algorithm 2). Class 1, the largest of the classes, is skipped for efficiency since it can be computed after the Shapley values of all other classes are determined. Then, for each class, it will iterate through the strata from 0 to $N - 1$ (Line 4, Algorithm 2). Between Line 5 and 12, the Shapley value of the iterating class is updated by adding the marginal contribution. Line 5 of Algorithm 2 shows that it will iterate through every possible subcoalition of the size of the stratum. As shown in Equation (4.1), the marginal contribution of an agent in a community is described as $c(\mathcal{S} \cup \{i\}) - c(\mathcal{S})$. Hence, it can be seen in Line 5 that the maximum number of agents from the iterating class k is one less than N_k, so that the marginal contribution can be computed. Line 6 assures that the formed subcoalition is from stratum j.

What makes it possible to compute the Shapley values efficiently for limited number of classes is the *(multivariate) hypergeometric distribution* [26]. The probability mass function of the multivariate hypergeometric distribution $P(\{n_1, ..., n_K\}, \{N_1, ..., N_K\}, N, n)$ computes the relative frequency of selecting n_1 agents from class 1 with the size of N_1, n_2 agents from class 2 of size of N_2, and repeating until class K, in a community of N agents. The function is formulated as the following.

$$P(\{n_1, ..., n_K\}, \{N_1, ..., N_K\}, N, n) = \frac{\prod_1^K \binom{N_k}{n_k}}{\binom{N}{n}} \tag{5.2}$$

where $\sum_1^K N_k = N$ and $\sum_1^K n_k = n$. The hypergeometric distribution allows to compute the probability of certain set of agents to be selected ahead over the chosen agent at the specific stratum. Line 7 in Algorithm 2 shows the step where the probability of the set of agents being ahead at stratum s is computed. The Shapley value of an agent can then be computed by replacing the relative frequency of a subcoalition in Eq. 4.2, with the hypergeometric function.

Line 8 in Algorithm 2 computes the marginal contribution of the agent from class k using the table containing costs of subcoalitions from Algorithm 1. The marginal contribution in Line 9 is added with the factor of the relative frequency of the subcoalition from stratum j. After iterating through every strata and subcoalitions, the value is divided by the total number of strata, which is N (Line 13). It can be seen that computation steps of Lines 8, 9, and 13 are equivalent to Equation (4.2), with the only difference being the relative frequency is computed using the hypergeometric function.

Line 15 computes the Shapley value of agents in class 1. Since the values of agents of all the other classes are known and the sum of Shapley values of all agents must equal to the community cost, the Shapley value of class 1 is equal to the remaining cost after subtracting the cost distributed to agents in class 2 to K from the community cost, then equally divide it by the agents from class 1, by the efficiency property [49].

Algorithm 2 Compute Shapley Exact values of K classes

Input: N, $N_1, ..., N_K$. Table of energy costs of subcoalitions, CS
Output: Array of Shapley values (redistributed cost) for agents in each class, $Shap$

1: **function** ComputeExactShapley($N, N_1, ..., N_K, CS$)
2: **for** $k \leftarrow 2$ to K **do** ▷ Iterate through every class except the first one
3: $Shap_k \leftarrow 0$
4: **for** $j \leftarrow 0$ to $N - 1$ **do** ▷ Iterate through every stratum
5: **for** all $(n_1, n_2, ..., n_K) \in \{0, 1, ..., \max(j, N_k - 1)\} \times \prod_{i=1, i\neq k}^K \{0, 1, ..., \max(j, N_i)\}$ **do**
6: **if** $j = \sum_{i=1}^K (n_i)$ **then**
7: $rel\_f \leftarrow P(\{n_1, ..., n_K\}, \{N_1, ..., N_k - 1, ..., N_K\}, N, j)$ ▷ Hypergeometric function
8: $mc \leftarrow CS[n_1, n_2, .., n_k + 1, .., n_K] - CS[n_1, n_2, .., n_k, .., n_K]$ ▷ Marginal contribution
9: $Shap_k \leftarrow Shap_k + rel\_f * mc$
10: **end if**
11: **end for**
12: **end for**
13: $Shap_k \leftarrow \frac{1}{N} Shap_k$
14: **end for**
15: $Shap_1 \leftarrow \frac{1}{N_1}(CS[N_1, N_2, ..., N_K] - \sum_{i=2}^K N_i * Shap_i)$
16: **return** $Shap$
17: **end function**

6

Computational Complexity of Redistribution Methods

This chapter presents the runtime analysis of the presented redistribution methods. The comparison is made between a community of N agents when having unique energy demands and when agents are grouped into K classes. Table 6.1 shows the time complexity of redistribution methods for both scenarios.

	Time Complexity	
Algorithm	**Unique**	K **Classes**
Exact Shapley	$\mathcal{O}(2^N \cdot (N-1))$	$\mathcal{O}(N^K \cdot (K-1))$
Marginal Contribution	$\mathcal{O}(N)$	$\mathcal{O}(K)$
Stratified Expected Values	$\mathcal{O}(N^2)$	$\mathcal{O}(K \cdot N)$
Approx. Shapley RL	$\mathcal{O}(N \cdot M)$	$\mathcal{O}(K \cdot M)$

Table 6.1: Time complexity per algorithm

When the community is restricted to K classes, the number of times the cost function needs to be computed by Algorithm 1 is equal to the number of all possible combinations of agents which is $(N_1 + 1) \times ... \times (N_K + 1)$ (illustrated in Figure 5.1). Considering w.l.o.g that the classes are ordered by the size, i.e., $N_1 \geq N_2 \geq ... \geq N_K$, and assuming there are at least two non-empty classes, i.e. $K \geq 2$, then it holds that $N_i + 1 \leq N$, $\forall i = 1, ..., K$. The number of cost function calculations is hence upper bounded by N^K. Iterating over different K classes gives the time complexity of $\mathcal{O}(N^K \cdot (K-1))$. For a community of unique agents ($K = N$), then $N_i = 1$, $\forall i = 1, ..., K$ and hence the time complexity of a community of unique agents is $\mathcal{O}(2^N \cdot (N-1))$.

While it seems that $\mathcal{O}(N^K \cdot (K-1))$ (for K classes) is large, in fact, for a large N and a small number of classes K this is much smaller than 2^N, hence in practice, exact Shapley computation with K classes has a much lower computation cost than unique agents.

For the last marginal contribution method (Section 4.2.1), the complexity was determined to be $\mathcal{O}(N)$ for a community of N agents, as it requires to compute the marginal contribution once for every agent. With K classes, however, this is reduced to $\mathcal{O}(K)$. Since agents with the same energy demands are assigned the same cost, Equations (4.3) and (4.4) are only required to be computed once per class.

The time complexity of stratified expected values method (Section 4.2.2) is $\mathcal{O}(N^2)$ for N unique agents, but it can be reduced to $\mathcal{O}(K \cdot N)$ for K classes. Similarly to the marginal contribution method, it is only required to compute Equation (4.6) once per class.

Finally, RL-based Shapley approximation method (Section 4.2.3) has the time complexity reduced from $\mathcal{O}(N \cdot M)$ to $\mathcal{O}(K \cdot M)$, where M is the number of samples per agent chosen by the user, by the same reasoning as the marginal and stratified expected values methods.

7

Data Pre-processing and Clustering

To be able to compute the exact Shapley values of a large community as the ground truth, the community needs to be made up of a limited number of classes, as explained in Chapter 5. In this study, the performance comparisons of the redistribution methods were made for communities of two, three, and four classes that are composed by clustering real consumer demand data. In the remaining of this chapter, the datasets used, as well as the formation of these classes and its demand profiles, are explained.

7.1. Datasets

7.1.1. Thames Valley Dataset

For the experiments with two classes (Section 8.1) and three classes (Section 8.2), the energy demands of 200 households in a realistically-sized energy community in the UK sharing a community wind turbine and battery were used, using the case study from Norbu et al. [39] (with the kind permission of the authors). The energy demands of the households are provided for every 30 minutes during one calendar year, which was largely collected in a well-known smart energy demonstrator project in the UK, the Thames Valley Vision project[1]. The half-hourly power generated by the wind turbine was calculated based on the power curve of Enercon E-33 wind turbines [19] and real wind data of the Kirkwall airport weather station in Orkney, Scotland from the UK Met Office Integrated Data Archive System (MIDAS) [41] provided by the British Atmospheric Data Centre (BADC).

7.1.2. London Dataset

For the case study with four classes (Section 8.3), the demands and the wind data were taken from a dataset in Kaggle[2], a ML data platform. The half-hourly demands of 5567 households in the London area, UK, between November 2011 and February 2014 were recorded by the UK Power Networks during the Low Carbon London project [20]. The corresponding London weather data was provided by Dark Sky [25], and the generated power by the wind turbine was calculated using the same method as Norbu et al. [39].

Both the demands and wind power data were aggregated to generate an averaged half-hourly data points for one year, aligned by the calendar weeks. From the demand data, households were removed if less than 95% of the data points from the year were missing, resulting in 5251 households left. The remaining miss data points were filled using linear interpolation.

7.2. Clustering & Consumer Profiles

7.2.1. Two classes

Two hundred prosumers from the Thames Valley dataset (7.1.1) were grouped into two classes of small consumers or large consumers according to the annual energy consumption. Small consumer

[1] https://ukerc.rl.ac.uk/DC/cgi-bin/edc_search.pl?WantComp=147

[2] https://www.kaggle.com/datasets/jeanmidev/smart-meters-in-london

and large consumer profiles were made from the average half-hourly demands of each group. In this study, two cases are tested; groups split into the 90% smallest and 10% largest consumers by total annual demand, and 80% smallest and 20% largest in the second test. The community consists of agents that had small and large consumer profiles, with the corresponding ratios (9:1 or 8:2).

7.2.2. Three classes

To create three classes of consumer profiles, the agents were grouped not by their total demands but rather by their consumption profiles over a day (24 hours). Clustering energy consumers into a number of classes, according to their daily consumption profile, is a well-established practice in energy demand modeling [27, 29, 64], used both in research and practice, by the energy suppliers. Identifying consumption patterns of customers can help the energy suppliers provide customers with recommendations as well as managing energy loads.

The 200 agents from the Thames Valley dataset (7.1.1) were clustered with the following steps. The half-hourly energy demands were normalised using L2 normalisation for each agent. From the normalised dataset, only the winter months of the UK were kept, which were January, February, November, and December. This is because the energy consumption is larger during winter, and hence clearer consumption patterns should be observable. From the remaining data, the days on and near Christmas and new year were also removed, which were January 1st to 6th and December 22nd to 31st. This is because many households would likely have abnormal consumption behaviour, such as being absent from home during these days or not working. Finally, only weekdays (from Monday to Thursday) were kept so that working days are what is being clustered. The remaining data consisted of 60 days $\times$ 48 normalised demands per day $=$ 2880 datapoints per agent. The new dataset is aggregated so that it contains averaged half-hourly normalised demands of each agent (48 datapoints per agent). Then, K-means clustering is used to group the agents on winter days' energy consumption. Both the normalisation and the clustering were done with scikit-learn [43] (version 1.0.1).

A major limitation of K-means clustering is that the number of clusters k is a parameter that the user sets. To identify the ideal number of clusters, using the elbow method is common practice [1, 17, 28, 32]. In the elbow method, the inertia, i.e., the sum of squared distances of samples to the nearest cluster centre, is calculated for resulting clusters of different k. By plotting the inertia on increasing k, the "elbow" point where the decrease in inertia slows down. Figure 7.1 shows the resulting elbow curve. The figure does not provide a clear "elbow", but it can be seen to be $4 \leq k \leq 6$. By also looking at the resulting clusters and their average daily demand, the cluster size is set to be 5. The average daily demands and the sizes of the 5 resulting clusters are presented in Figure 7.2.

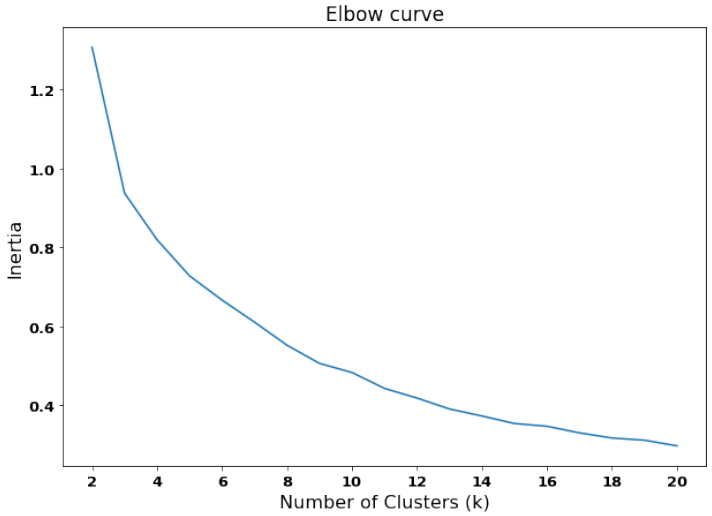


Figure 7.1: Change in inertia with increasing number of clusters in K-means clustering on Thames Valley dataset.

Clusters 1, 3, and 4, which make up the majority, show an increase in demand in the morning and a large peak in the evening. Especially cluster 4 shows a pattern in which there is a small morning consumption peak followed by minimal consumption during the day and a final large peak in the evening.

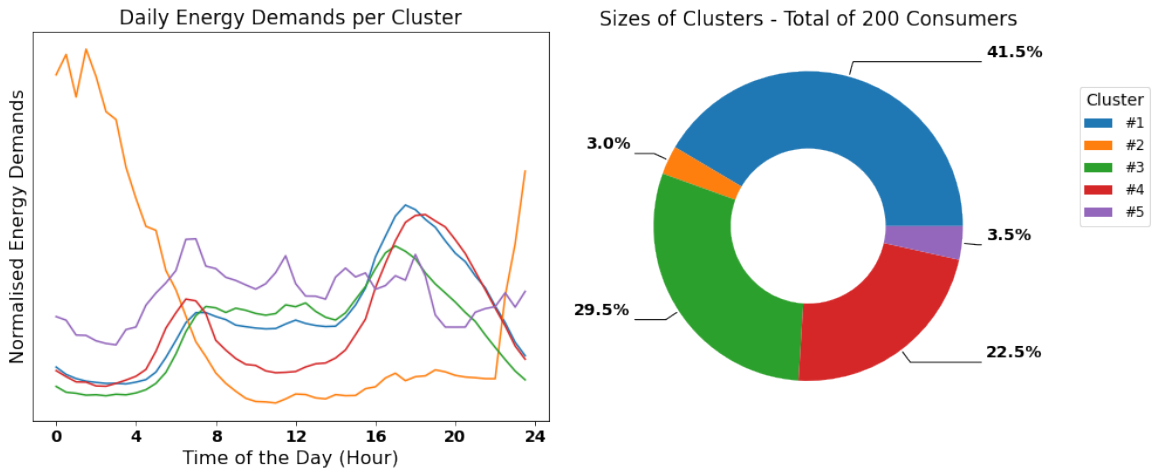


Figure 7.2: Daily energy demands and the relative sizes of the 5 consumption clusters.

Cluster 1 also shows a similar pattern but slightly high consumption during the day. Consumers in these two clusters combined make up over 60% of the community, representing households that are often absent during the day. Note that the data is before the onset of the COVID-19 pandemic, and therefore working from the office during the day was common, making consumption during the day low. We have named a class “work at office” and used the demands of cluster 4 as it shows a more clear consumption pattern. Cluster 3 also shows a resembling pattern to that of clusters 1 and 4, however, a higher consumption during the day and a smaller evening peak can be seen. Such a behaviour can be representative of households working at home, as staying at home requires certain energy consumption during the day, such as for heating, computers, and kitchen appliances, resulting in an overall high consumption that is not too concentrated in the evening. We suspect that consumers with such consumption behaviour have increased since the outbreak of COVID-19, and hence class 3 was chosen as the “work from home” class to see the impact of such behaviour in the community. One clear outlier cluster is class 2. Although only 3% of the consumers belong to this class, it has the most distinct behaviour from the rest of the classes. This class has a very high demand during the night, yet very little demand during the day, a behaviour that is also observed in the work by Jeong et al. [27]. This “night owl” class was chosen as the fourth class for this study as the outliers of the community. Cluster 5 was not considered a valid consumption behaviour since the consumers in this class do not seem to have a common pattern, resulting in a nonsensical average daily consumption. When clustering on larger data, as shown in the next section, such cluster did not occur. It is also important to note that clustering consumer behaviour is an important study by itself. In this work, the consumer classes are formed in order to reduce the number of demand profiles in the community, and clustering is not the main focus. Hence, it is possible that better clusters can be formed using different pre-processing/clustering methods. However, for the purpose of this study, the chosen three classes present different consumption behaviours that are representative of an energy community.

The average daily consumption of the three classes (“work at office”, “work from home”, and “night owl”) are presented in Figure 7.3. In 2020, due to COVID-19, working from home became the norm, and thus it would be of interest to look at a change of behaviour from working at the office to home. Hence these clusters were chosen. Additionally, a community would normally consist of outliers, and hence “night owl” cluster was also included. From grouped agents, the half-hourly energy demands are created for three consumer profiles.

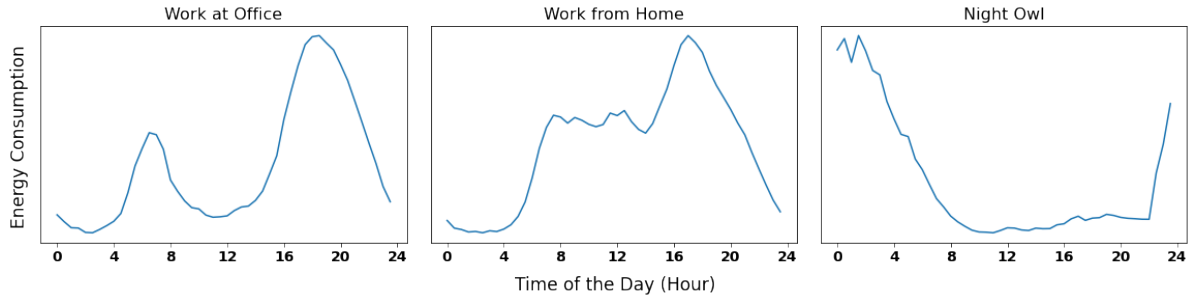


Figure 7.3: Average daily energy demands of consumer profiles of the three chosen classes, work at office, work at home, and night owl

Once the clustering is done and consumption behaviours are identified, half-hourly energy demands for one year of the three consumer profiles need to be identified. For each class, the L2 normalised half-hourly demands of the year (no days removed and not aggregated) were averaged over the agents belonging to that cluster. Next, energy demands were multiplied by a certain ratio for each class (in this study, wo: 1, wh: 1.1, no: 1) to ensure that the "work from home" class had larger total consumption than other classes since it is logical to consume more energy by staying at home. The yearly demand of the community of these agents is kept equal to N times the average yearly consumption of the 200 agents from the original dataset. Given the number of classes N_{wo}, N_{wh}, and N_{no} and $N = N_{wo} + N_{wh} + N_{no}$, the restriction is represented as the following.

$$\frac{N}{200} \sum_{i=1}^{200} \sum_{t=1}^{T} d_i(t) = \sum_{k \in \{wo,wh,no\}} \sum_{t=1}^{T} N_k d_k(t) \tag{7.1}$$

From the restriction, the demand profiles of each class can be computed as the following, given $\tilde{d}_k(t)$, the normalised (and multiplied by the ratio) demand of class k at time step t.

$$d_k(t) = \tilde{d}_k(t) \cdot \frac{\frac{N}{200} \sum_{i=1}^{200} \sum_{t=1}^{T} d_i(t)}{\sum_{q \in \{wo,wh,no\}} \sum_{t=1}^{T} N_q \tilde{d}_q(t)}, \quad \forall k \in \{wo, wh, no\}, \forall t \in \{1, ..., T\} \tag{7.2}$$

7.2.3. Four classes

Four classes of consumer profiles were generated from the London dataset (7.1.2) with 5251 consumers. The annual demands were pre-processed as in Section 7.2.2, which results in 48 half-hourly demands representing the normalised average daily consumption during the winter weekdays per consumer. The consumers are then clustered using K-means clustering. Again, the elbow curve was considered when choosing the number of clusters. Figure 7.4 shows the elbow curve of the London dataset. Similarly to the three classes, no clear "elbow" can be observed, but the point can be estimated to be around $6 \leq k \leq 9$. By also looking at the resulting clusters and their average daily demand, the cluster size is set to be 9. The average daily demands and the sizes of the 9 resulting clusters are presented in Figure 7.5.

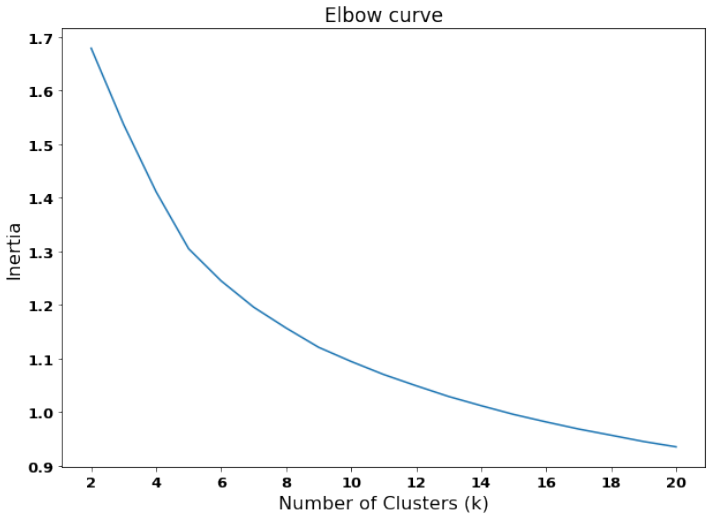


Figure 7.4: Change in inertia with increasing number of clusters in K-means clustering on London dataset.

It can be seen from Figure 7.5 that although the detailed consumption behaviours are unique, many classes shows similarity in terms of having a small morning peak in the morning and an evening peak. Clusters 1, 2, 3, 7, and 9 show variations of such behaviour, and it makes up more than 60% of the 5251 consumers, providing similar results to the three class case. We have chosen the “work at office” class and used the demand profile of cluster 7 as the representative as it is the largest class out of the evening peaking classes. Next, cluster 5 shows a high and constant consumption during the day, which represents the households that are home during the day. Hence, similarly to the three class case, the “work from home” class is selected for the experiment. Clusters 4 and 6 are unique from the rest of the classes as they both have two almost equal peaks in the morning and in the evening, though the morning peak is slightly larger. This behaviour is a minority in the community, yet 17.5% of the consumers belong to these groups. Hence, these consumers need to be represented in the community as well, and the demand profile of cluster 4 is used for the “M-shaped” class in this study as it shows the behaviour more clearly. Finally, the consumption behaviour of the “night owl” agent is clearly visible in cluster 8, where there is a high demand overnight but very little during the day. The “night owl” class was again chosen as the fourth class, providing diversity in the community. The daily average demands of the chosen four classes are presented in Figure 7.6.

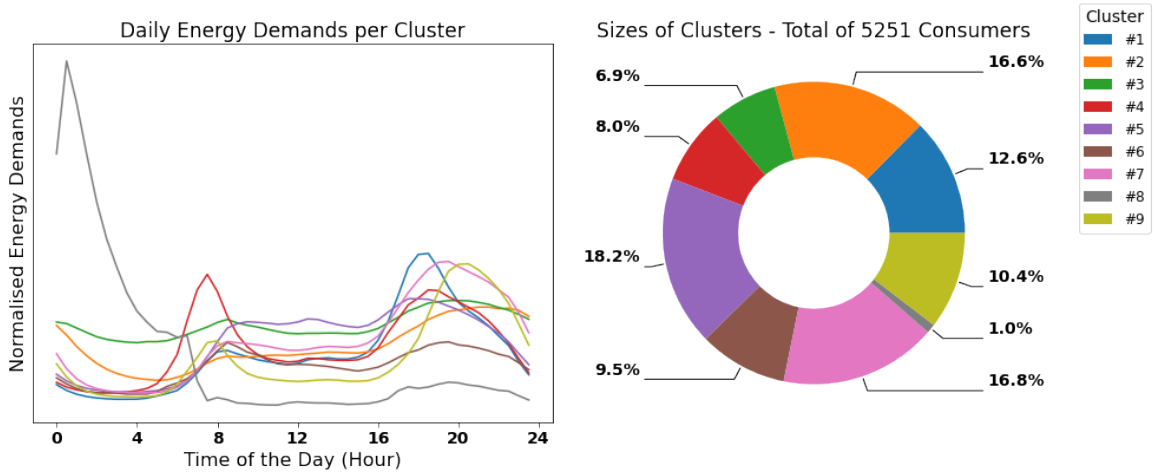


Figure 7.5: Daily energy demands and the relative sizes of the 9 consumption clusters.

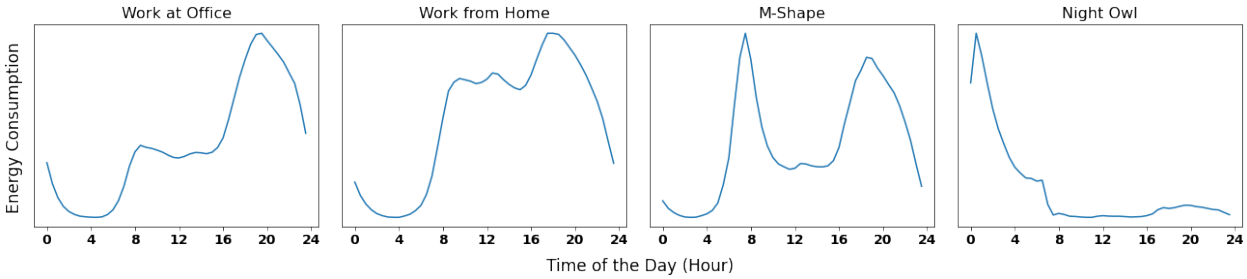


Figure 7.6: Average daily energy demands of consumer profiles of the four chosen classes, work at office, work at home, M-shape, and night owl

Once the four classes were chosen, their demand profiles were created by following the procedure as in the three classes. The average L2 normalised half-hourly demands of each class are multiplied by a ratio (wo: 1, wh: 1.1, ms: 1, no: 1) to make sure that the “work from home” class have a slightly higher annual consumption. Then, the demand profile of the class can be computed by Equation (7.2).

8

Experimental Comparison

In this chapter, the experimental comparisons of the redistribution methods are presented. The energy community was created using the consumption classes described in Chapter 7. All experiments were run under macOS Monterey on an Apple M1 with a 8 core CPU, with a 16 GB RAM and a 12 MB L2 cache. The experiments were run on Python 3 (version 3.8.5). Furthermore, for all the experiments, the import tariff was set to 16 UK pence/kWh based on past price comparisons with UK energy suppliers [39]. The export tariff was set to 0 pence/kWh since most energy suppliers provide minimal or no returns for selling energy back to the grid [39]. Hence we assumed that no profit could be made from exporting energy to the grid.

The remaining chapter describes each case study, including the experimental setup and results. Finally, Section 8.4 provides discussions on the results from the three case studies.

8.1. Case Study 1: Two Classes

8.1.1. Experimental Setup

Data and Community Setup

In the first case study, the small and large consumer profiles derived from the Thames Valley dataset in Section 7.2.1, and the corresponding wind turbine power generation data were used. In this case study, two scenarios are tested; One community where 90% of the consumers belong to the small consumer class and the rest of the 10% belong to the large consumer class, and another community with 80% of small consumers and 20% of large consumers. In the experiment, the community size is increased while keeping the composition ratios of the community constant. How the Shapley value approximations of the redistribution methods are influenced by the community size is tested in this case study.

Performance Measure

The performances of the redistribution methods were measured by how well they approximated the ground truth, exact Shapley values. Hence, the *relative difference* to the exact Shapley values was used for comparing the redistribution methods. The percentage relative difference of a cost to the Shapley value is defined as the following.

$$RD_{\phi}(\hat{\phi}_k) = \frac{|\hat{\phi}_k - \phi_k|}{\phi_k} \times 100 \tag{8.1}$$

where $\hat{\phi}_k$ is the energy cost of an agent of class k from a particular redistribution method, which are $\overline{MC}_k$, $\overline{SEV}_k$, and RL_k. The variable ϕ_k is the cost redistributed to class k according to the Shapley value. The relative difference does not only take the magnitude of difference between the approximation method and the exact value but also considers how large the exact value is. This provides a fairer evaluation between different demand profiles, as demand profiles with naturally large energy costs could have significant approximation error in terms of magnitude only from slight deviation.

The relative difference is computed for each agent/class. To measure the performances of the redistribution methods on the overall community, the *average* relative difference is used. The average

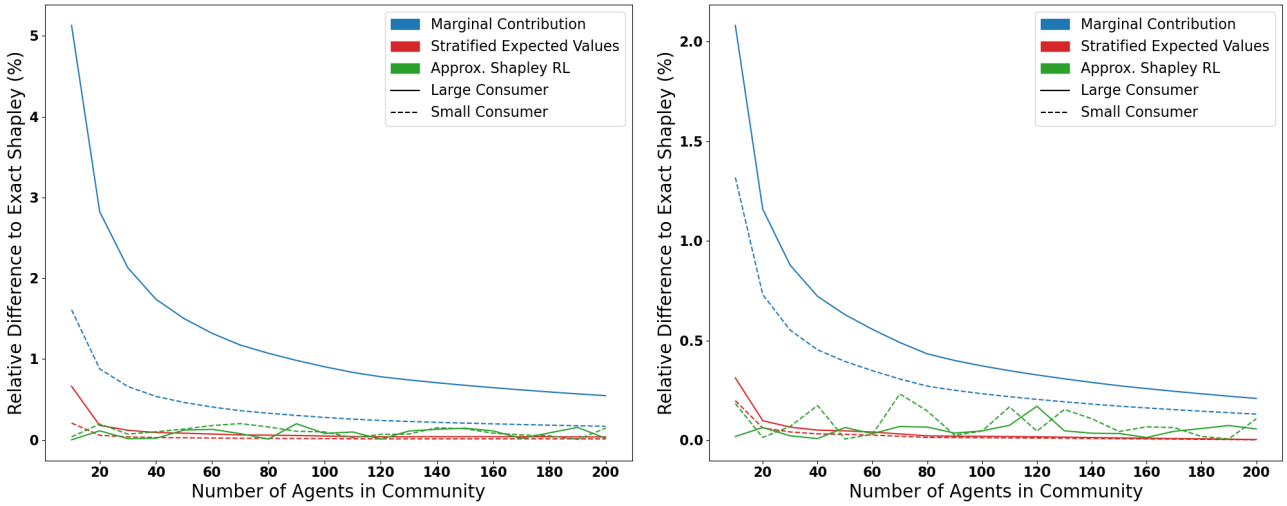
relative difference to the Shapley value of a redistribution method is the mean relative difference of the community:

$$RD_{\phi}(\hat{\phi}) = \frac{1}{N}\sum_{k}^{K} N_k \cdot RD_{\phi}(\hat{\phi}_k) \tag{8.2}$$

where $\hat{\phi}$ is a redistribution method with costs $\hat{\phi}_1, ...\hat{\phi}_K$ assigned to K classes.

8.1.2. Results

Whether the size of the community influences how well the redistribution methods approximate the Shapley values were investigated. Figure 8.1 shows the change in relative difference to the exact Shapley values of the redistribution methods for each of the small and large consumer classes with increasing community size up to 200 households while keeping the same ratio of small and large consumer profiles. In Figure 8.1a, the ratio of small consumer agents and large consumer agents was kept to 9:1, while Figure 8.1b used the ratio of 8:2.



(a) 90/10 split

(b) 80/20 split

Figure 8.1: Relative differences of the small and large consumer agent profiles to the exact Shapley values for the redistribution methods with increasing size of the community.

Figure 8.1 showed that all the tested redistribution methods approximated Shapley values well for a large community. Although the marginal contribution method yielded high errors for small community size (for example, 5% difference with exact Shapley for large consumer profile in 90/10 split scenario, Figure 8.1a), for community size of over 100 prosumers, all methods were below 1% difference with the exact Shapley in all scenarios. As expected, the stratified expected values method outperforms the simpler marginal contribution method in all cases and all scenarios. Furthermore, there is a minimal difference in the performances between the stratified expected values and the adaptive sampling methods in most cases. In fact, paired two-sample t-tests on 2 class experiments with 0.05 significance level showed that the stratified expected values method (90/10 split: $M = 0.0339$, $SD = 0.0520$. 80/20 split: $M = 0.0285$, $SD = 0.0471$) had a smaller average relative difference to true Shapley values compared to RL-based adaptive sampling method (90/10 split: $M = 0.1110$, $SD = 0.0496$. 80/20 split: $M = 0.0934$, $SD = 0.0444$) for both 90/10 split($t(19) = -3.919$, $p < 0.001$) and 80/20 split ($t(19) = -4.236$, $p < 0.001$).

Table 8.1 shows the comparison of redistributed yearly energy costs of a small and large consumer agent for both 90/10 and 80/20 splits of 200 households community. It can be seen that, for both splits, the energy costs of the small consumer agent are very close across the four redistribution methods, although the redistributed costs based on the marginal contribution are slightly higher than the other methods. On the other hand, the marginal contribution method is lower than the other methods for the large consumer agent. The values for the large consumer agent are still similar across the methods, but the difference between the marginal contribution methods and the Shapley value is more significant

compared to the small consumer agent.

Table 8.1: Redistributed costs of small and large consumer agents in a community of 200 agents.

Split	**90 / 10**		**80 / 20**	
Agent Class	**Small**	**Large**	**Small**	**Large**
Shapley	245.30	674.88	221.78	554.18
Marginal Contribution	245.71	671.19	222.07	553.02
Stratified Expected Values	245.33	674.64	221.78	554.16
Approx. Shapley RL	245.50	674.47	222.04	553.65

8.2. Case Study 2: Three Classes

8.2.1. Experimental Setup

Data and Community Setup

In the next case study, communities composed of three classes ("work at office", "work from home", and "night owl") from Section 7.2.2 were used. These classes are derived from the consumption data from the Thames Valley dataset (Section 7.1.1), and the corresponding wind turbine power generation data was used. In this case study, we again perform experiments on two scenarios. In the first setting, the composition of the consumer profiles in the community was kept constant with the increasing size of the community, similarly to the experiments in Section 8.1. In the second setting, the community size was kept constant, but the demand class composition in the community changed. With this scenario, we looked at how different composition (such as concentrated to one class or more equally distributed) affects the Shapley approximations of the redistribution methods.

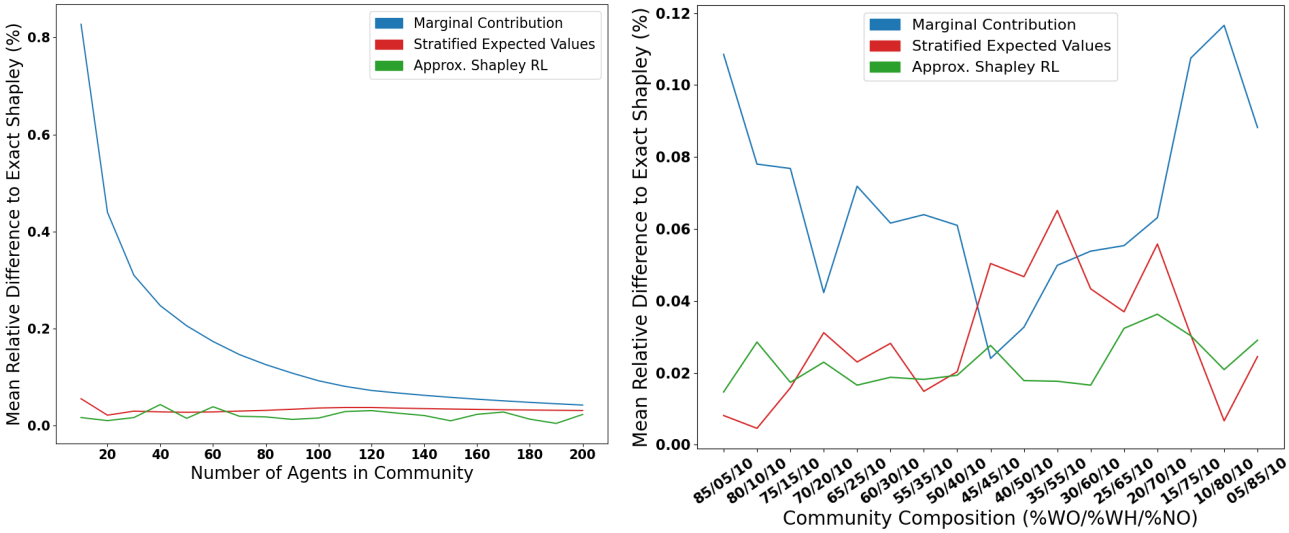
Performance Measure

To compare the performances of the redistribution methods, we again use the relative difference to the exact Shapley values, given as Equation (8.1), to evaluate the approximation error of a class by a redistribution method. The relative differences of the classes are used to compute the average relative differences (overall approximation error of the community) of the redistribution methods, given as Equation (8.2).

8.2.2. Results

Figure 8.2 shows the average relative differences to the exact Shapley values of the redistribution methods in different scenarios. Figure 8.2a shows the change in average relative differences with increasing community size, starting from $N = 10$ up to $N = 200$. The community was composed of 70% "work at office" agents, 20% "work from home" agents, and 10% of "night owl" agents. The results show similarity with that of the two class community in Section 8.1. First, the marginal contribution method has a large error in small communities but approximates the Shapley values well for larger communities. Second, both the stratified expected values and the adaptive sampling methods approximate well for all community sizes and outperform the simpler marginal contribution method for all community sizes. The difference between the results in the two classes case is that the adaptive sampling method slightly outperforms the stratified expected values method. A paired two-sample t-test with 0.05 significance level also showed that the adaptive sampling method ($M = 0.0207$, $SD = 0.0095$) has lower average relative difference to the Shapley values than the stratified expected values method ($M = 0.0331$, $SD = 0.0063$) ($t(19) = 4.6951$, $p < 0.001$). Still, the difference is small, and there were certain cases where the stratified expected values approximated the Shapley values better.

The relative difference of each class from Figure 8.2a is shown in Figure 8.3. The "work at office" class in Figure 8.3a shows that the approximation error by the marginal contribution method converges as the community size increases, similarly to the previous results. Nevertheless, the approximation error is only 0.5% in the worst case. The other two methods approximate the Shapley value of the "work at office" class well for any community size. Figure 8.3b shows that all three methods approximated the Shapley value of the "work from home" class well. Interestingly, the marginal contribution outperforms



(a) Different Community Size

(b) Different Compositions

Figure 8.2: Change in performance of redistribution methods with (a) change in the community size, and (b) change in composition of the consumer profiles in the community.

the other two methods in most cases. Finally, Figure 8.3c shows the "night owl" class has the largest approximation error for the marginal contribution and the stratified expected values methods out of the three classes. The approximation error of the marginal contribution method converges to a smaller error as the community size increases, but the error starts from almost 5%. In fact, the "night owl" agents contribute the most to the overall error (average relative difference) of the marginal contribution and the stratified expected values method in Figure 8.2a, even though they only make up 10% of the community.

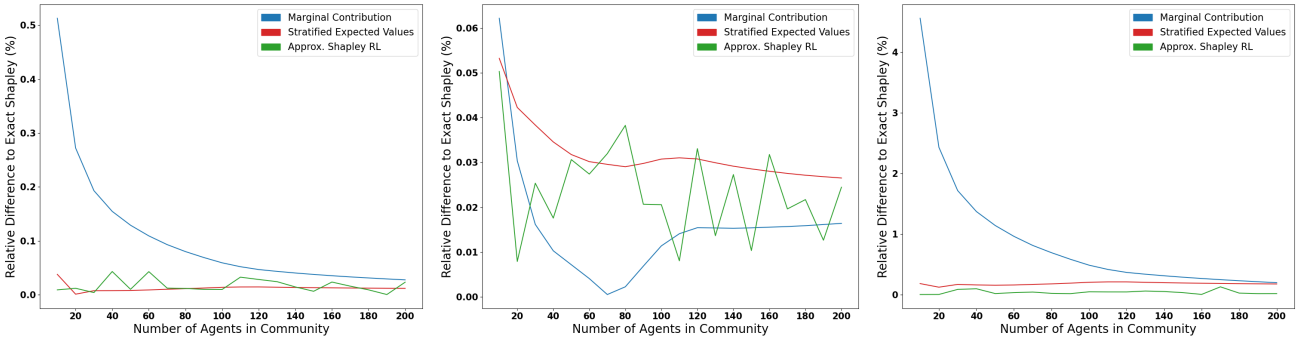
Table 8.2 shows the comparison of redistributed costs of three consumption profiles in a community of 200 agents with 70% "work at office", 20% "work from home", and 10% "night owl". All methods approximate the Shapley value well for every consumption profile. Especially, the differences between the Shapley values for "work at office" and "work from home" agents are very small. For the "night owl" agent, the difference is slightly larger for marginal contribution and the stratified expected values methods, but the difference is still less than 0.2%.

Table 8.2: Redistributed costs of three consumer profiles in a community of 200 agents with the composition, 70% of "work at office" agents (WO), 20% of "work from home" agents (WH), and 10% "night owl" agents (NO).

Split	**70 / 20 / 10**		
Agent Class	**WO**	**WH**	**NO**
Shapley	292.07	317.35	233.89
Marginal Contribution	292.15	317.30	233.43
Stratified Expected Values	292.03	317.27	234.30
Approx. Shapley RL	292.13	317.43	233.93

Figure 8.2b shows the change in average relative differences of redistribution methods with change in the composition of the community, with $N = 200$. In this experiment, the "night owl" class, the outliers of the community, was kept constant by making up 10% of the community. The rest of the community was made up of "work at office" and "work from home" classes. Initially, the "work at office" was the majority of the community, making up 85% of the community. Then, after every run, the ratio of the "work at office" class in the community was reduced by 5% and instead, the ratio of the"work from home" class increased by 5%. The simulation ran until the community contained a majority of the "work

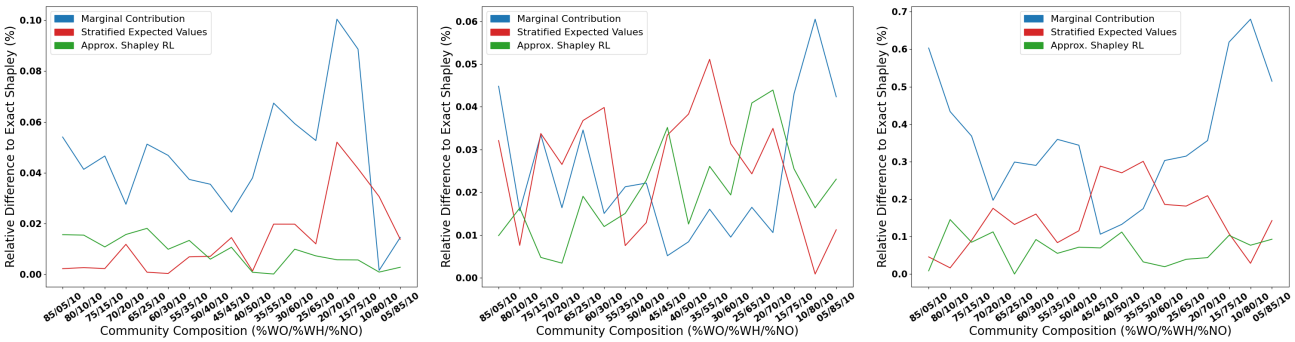
from home" class with 85%. The results show that all three methods approximate Shapley values well (all the methods in every composition scenario have less than 0.12% difference), as the community size is large already. Still, some patterns are observable. The marginal contribution method seems to have a better Shapley value approximation when the community is more mixed where the community composition ratios of "work at office" and "work from home" classes are about the same. On the other hand, the stratified expected values method shows the opposite trend, where the error is slightly higher when the community is concentrated on one class. Overall, the marginal contribution method shows a slightly higher error. Furthermore, a paired two-sample t-test showed that stratified expected values ($M = 0.0298$, $SD = 0.0173$) and adaptive sampling ($M = 0.0226$, $SD = 0.0064$) methods did not have statistical difference in average relative difference to the Shapley value ($t(16) = 1.6749$, $p = 0.11$).



(a) Work at Office **(b)** Work from Home **(c)** Night Owl

Figure 8.3: Relative differences to the exact Shapley values for the redistribution methods of three consumer profiles with increasing size of the community.

Figure 8.4 presents the change in the relative difference of the redistribution methods for each demand profile ("work at office", "work from home", "night owl") with different composition of the community. For the "work at office" class in Figure 8.4a, the marginal contribution method shows a larger error than the other two methods, although the error is less than 0.1% for most cases. Similarly, Figure 8.4b shows that the approximation errors of the "work from home" class are very small for all three redistribution methods. Furthermore, it shows many overlaps in approximation error, and hence no method clear outperforms another one. Finally, Figure 8.4c once again shows that the redistribution methods struggle the most to approximate the "night owl" class. The "night owl" agents, which only consist 10% of the community, contribute significantly to the overall error of the marginal contribution and the stratified expected values methods. It can be seen that the lines of the two methods in Figure 8.4 follow the same patterns as in Figure 8.4c.



(a) Work at Office **(b)** Work from Home **(c)** Night Owl

Figure 8.4: Relative differences to the exact Shapley values for the redistribution methods of three consumer profiles with different community compositions.

8.3. Case Study 3: Four Classes

8.3.1. Experimental Setup

Data and Community Setup

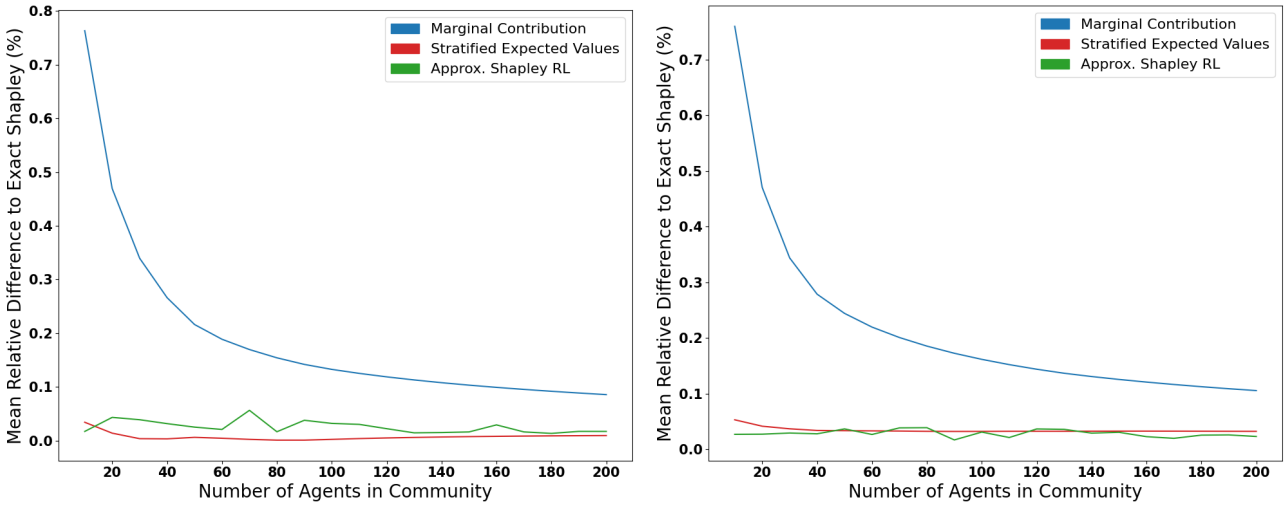
In the final case study, the performances of redistribution methods were tested on communities consisting of four classes (“work at office”, “work from home”, “M-shape”, and “night owl”), explained in Section 7.2.3. These demand profiles were generated from the London dataset (Section 7.1.2), and the corresponding wind turbine power generation was used for the experiments. In this case study, we performed two experiments. In the first experiment, the ratios of the community were kept constant, and the performances of the redistribution methods were tested for community sizes of $N = 10$ to $N = 200$. We tested two scenarios; one in which the community is concentrated to one class and one in which the community is more evenly spread out between the classes. In the second setting, the community size is kept constant, but the composition (ratios) of the consumer classes in the community changes.

Performance Measure

The performances of the redistribution methods are again compared by computing the relative differences to the true Shapley values of the class (Equation (8.1)) and the average relative differences of the community (Equation (8.2)).

8.3.2. Results

Figure 8.5 shows the change in average relative differences with increasing community size, starting from $N = 10$ up to $N = 200$. Figure 8.5a presents the results of the community with compositions of 70% “work at office”, 10% “work from home”, 10% “M-shape”, and 10% “night owl”, and Figure 8.5b with compositions of 30% “work at office”, 30% “work from home”, 30% “M-shape”, and 10% “night owl”.



(a) 70/10/10/10 split

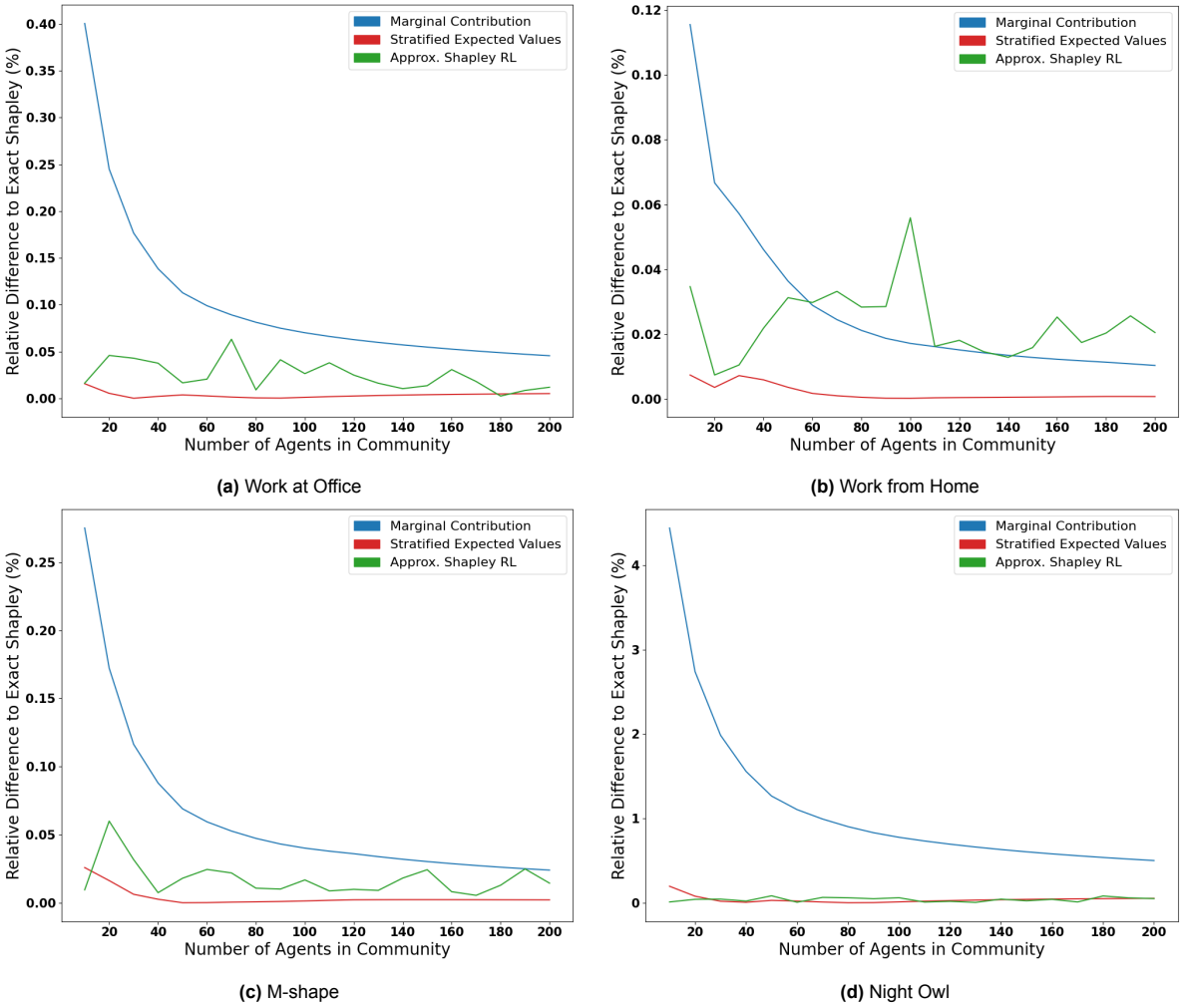
(b) 30/30/30/10 split

Figure 8.5: Average relative differences to the exact Shapley values for the redistribution methods with increasing community size in a community with four consumption profiles.

From both experiments, it can be seen that the redistribution methods approximate the Shapley values well for large communities. The errors of the marginal contribution method converge as the community size increases, whereas the stratified expected values and adaptive sampling methods show very low errors for all community sizes. The two methods also clearly outperforms the marginal contribution method. Furthermore, the stratified expected values shows smaller error than the adaptive sampling method for most community sizes in Figure 8.5a. Paired two-sample t-tests show that, in the four class case with community concentrated to one class (Figure 8.5a), the stratified expected values ($M = 0.0071$, $SD = 0.0070$) outperformed the adaptive sampling method ($M = 0.0254$, $SD = 0.0114$) ($t(19) = -5.376$, $p < 0.001$). On the other hand, when the community is more evenly spread

(Figure 8.5b), the adaptive sampling method ($M = 0.0283$, $SD = 0.0062$) outperformed the stratified expected values ($M = 0.0341$, $SD = 0.0048$) ($t(19) = 3.173$, $p = 0.005$).

Figure 8.6 contains the relative differences to the exact Shapley values of the redistribution methods for each consumer profile with increasing community size and the composition of 70/10/10/10 (concentrated community). It can be seen that for all classes, the marginal contribution method has a similar performance curve, where the error is very large for small community size but decreases quickly as the community size increases. Still, the error value is significantly larger for "night owl" class (Figure 8.6d) than the other three. It is also noticeable for "stay at home" class (Figure 8.6b) that it outperforms the adaptive sampling method from medium-sized communities ($N \geq 70$). The stratified expected values method, on the other hand, shows high similarity to the exact Shapley values for any community size and outperforms the marginal contribution method in every scenario. Furthermore, it also outperforms the computationally larger adaptive sampling method in most scenarios, especially for "work at office" (Figure 8.6a), "work from home" (Figure 8.6b), and "M-shape" (Figure 8.6c) classes.



(a) Work at Office

(b) Work from Home

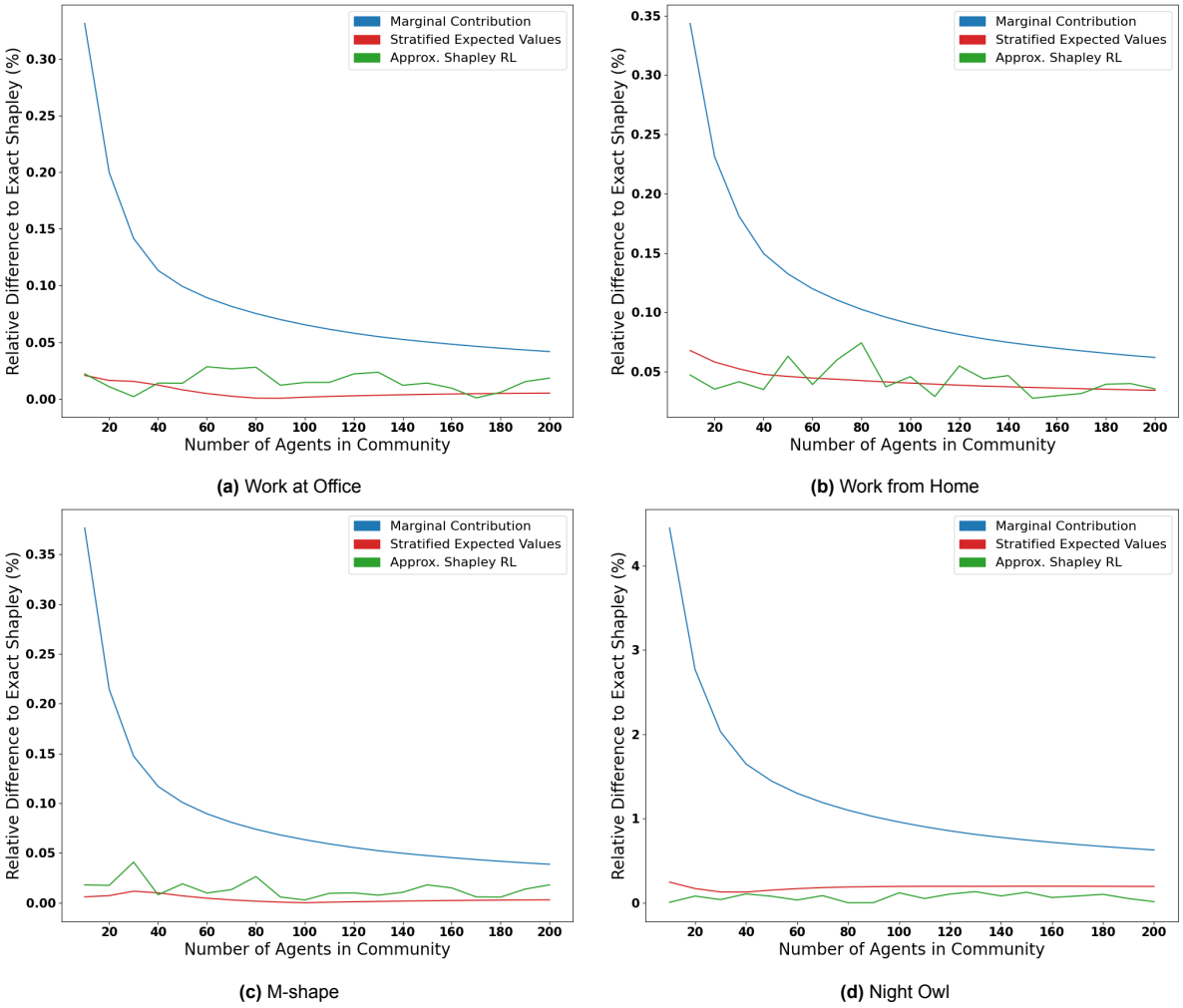
(c) M-shape

(d) Night Owl

Figure 8.6: Individual relative differences to the exact Shapley values for the redistribution methods of four consumer profiles (70% "work at office", 10% "work from home", 10% "M-shape", and 10% "night owl") with increasing size of the community.

Similarly, Figure 8.7 shows the relative differences of each class with increasing community size and community composition of 30/30/30/10 (even community). Again, the marginal contribution method shows that the approximation error converges as the community size grows for every class. Still, it is outperformed by the stratified expected values and the adaptive sampling methods. The stratified expected values does not perform as well as in Figure 8.6, only outperforming the adaptive sampling method on "M-shape" class (Figure 8.7c). Although there seems little difference in the overall performances between the two methods for "work at office" (Figure 8.7a) and "work from home" (Figure 8.7b)

classes, the adaptive sampling outperforms the stratified expected values for “night owl” class (Figure 8.7d) with relatively larger values. In fact, the “night owl” class is, again, the main contributor of the overall error of the stratified expected values method in Figure 8.5b.



(a) Work at Office

(b) Work from Home

(c) M-shape

(d) Night Owl

Figure 8.7: Individual relative differences to the exact Shapley values for the redistribution methods of four consumer profiles (30% “work at office”, 30% “work from home”, 30% “M-shape”, and 10% “night owl”) with increasing size of the community.

Table 8.3 shows the comparison of redistributed costs of four consumption profiles in a community of 200 agents for 70/10/10/10 and 30/30/30/10 compositions. All methods approximate the Shapley value well for every consumption profile. Especially, the differences between the Shapley values for “work at office”, “work from home”, and “M-shape” classes are very small. For the “night owl” class, the difference is slightly larger for the marginal contribution method, but the difference is still only about 0.6% for the 70/10/10/10 composition and 0.5% for the 30/30/30/10 composition.

Table 8.3: Redistributed costs of four consumer profiles ("work at office": WO, "work from home": WH, "M-shaped": MS, and "night owl": NO) in a community of 200 agents with two different compositions.

Split	**70 / 10/ 10 / 10**				**30 / 30 / 30 / 10**			
Agent Class	**WO**	**WH**	**MS**	**NO**	**WO**	**WH**	**MS**	**NO**
Shapley	341.55	369.41	325.76	239.86	334.90	364.49	321.34	234.65
Marginal Contribution	341.70	369.45	325.84	238.65	335.04	364.71	321.47	233.18
Stratified Expected Values	341.56	369.41	325.77	239.72	334.92	364.61	321.35	234.19
Approx. Shapley RL	341.59	369.48	325.81	239.73	334.96	364.62	321.40	234.62

Figure 8.8 shows the change in average relative differences of redistribution methods with change in the composition of the community. The community size was set to 200, and the ratios of "M-shape" and "night owl" agents were also kept constant to 20% and 10%, respectively. Initially, the "work at office" class is set to be 65% of the community and the "work from home" class to 5%. After every run, the ratio of "work at office" is reduced by 5% and "work from home" increased by 5%, until the "work from home" makes up 65% of the community and "work at office" with only 5%. Here, all three methods approximate Shapley values well (all methods in every scenario less than 0.15% difference), as the community size is large already. Still, the stratified expected values and the RL-based adaptive sampling methods outperform the marginal contribution method. Additionally, a paired t-test that there is no significant difference between the stratified expected values ($M = 0.0199$, $SD = 0.0114$) and the adaptive sampling ($M = 0.0220$, $SD = 0.0056$) methods on their average relative differences to the Shapley values ($t(12) = -0.660$, $p = 0.52$). Yet, it can be seen from Figure 8.8 that the stratified expected values method has a smaller difference to true Shapley values when the community is concentrated on one class and shows larger errors when the community is more even.

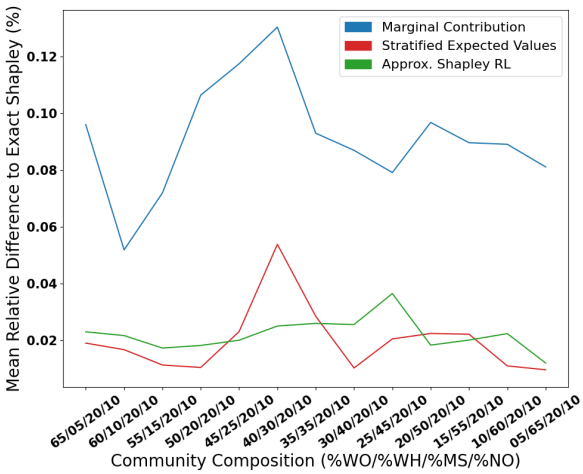
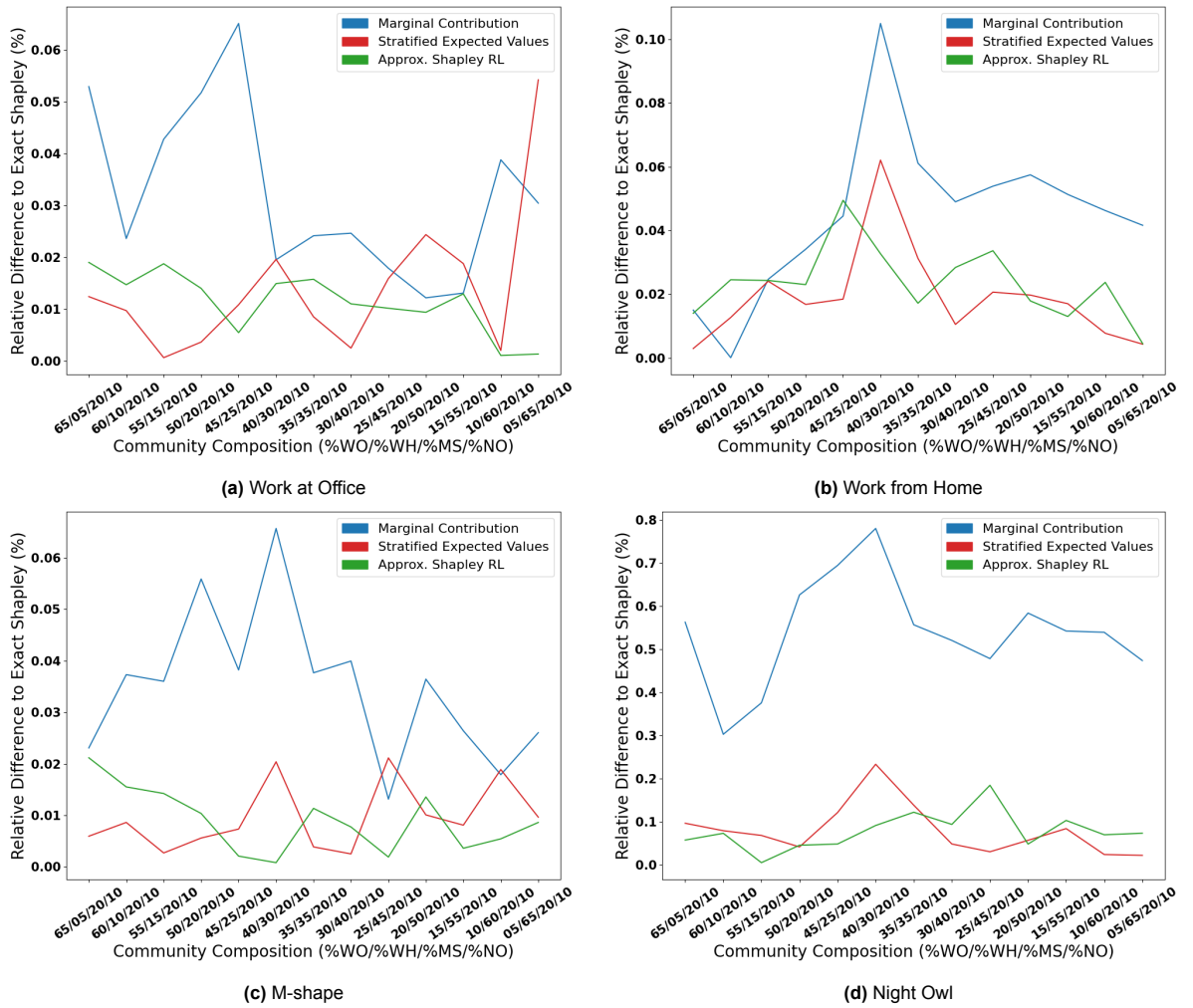


Figure 8.8: Average relative differences to the exact Shapley values for the redistribution methods of different community compositions with $N = 200$.

Figure 8.9 shows the relative differences of each class with changing community composition. Again, the marginal contribution method shows higher error than the other two redistribution methods. It is also noticeable that the trends of the lines of the marginal contribution and the stratified expected values methods are almost identical between Figure 8.8 and Figure 8.9d, displaying that the error caused by the "night owl" class is once again the main contribution of the overall error of these two methods.



(a) Work at Office

(b) Work from Home

(c) M-shape

(d) Night Owl

Figure 8.9: Individual relative differences to the exact Shapley values for the redistribution methods of four consumer profiles of different community compositions with $N = 200$.

8.4. Discussion

The three case studies presented above showed that all three redistribution methods approximated the Shapley values well for large communities. The marginal contribution method showed high error for small communities, but the error converged to a smaller value as the community size grew. We attribute the smaller difference to the exact Shapley value in larger communities to a smoothing effect that a large number of agents have to individual variations. In a large community, the Shapley calculation is dominated by marginal contributions of the agent to already large-sized subcoalitions. Variations in marginal contributions to large subcoalitions are often small, making it possible for less complex methods to approximate well for large communities.

As expected, the stratified expected values method outperforms the simpler marginal contribution method in all cases. Furthermore, there is a minimal difference in the performances between the stratified expected values and the RL-based adaptive sampling methods in most cases. The number of samples per agent was set to 1000 for the adaptive sampling method, meaning that for a case of 100 prosumers community, the adaptive sampling method had a time complexity ten times higher than the stratified expected values method (from Table 6.1). Yet, interestingly, the two methods showed very similar performance. Furthermore, statistical analyses showed that the stratified expected values method showed better Shapley value approximations than the adaptive sampling method in most scenarios when the community was highly concentrated to one demand profile class. Thus, both the stratified expected value and the sampling method approximate Shapley value seem to approximate the real Shapley very well, although the stratified expected value is computationally cheaper.

The results of the three class and the four class case studies also suggest that the stratified expected values may be sensitive to the composition of the community. The stratified expected values method had a smaller difference to true Shapley values when the community is concentrated on one class and shows larger errors when the community is more even. A potential reason for this method performing better in a concentrated community is that it approximates the expected marginal contribution of each stratum by creating a fictitious agent that has the average consumption of the rest of the community. This fictitious demand profile defined in Equation (4.5) becomes closer to the demand profile of one class as the number of this class in the community grows, and hence the "fictitious" demand profile will highly resemble an existing demand profile. As the average demand profile represents the community better in such cases, computing the approximated expected marginal contributions of the strata as in Equation (4.6) becomes more accurate. Still, it can be seen from when clustering consumer behaviours in Sections 7.2.2 and 7.2.3 that most consumers had an evening peak (belonging to "work at office") and made up 60% of the households studied. Hence it is common to have a community that contains a majority of the same consumption behaviour class, making the stratified expected values method applicable in real-world scenarios. Nevertheless, it would be interesting to study how the redistribution methods are affected when the community consists of several consumption behaviour types but also with different consumption quantities.

Both the marginal contribution and the stratified expected values method, but especially the marginal contribution method, showed large approximation error in the rarer, unusual class (large consumer in two class, and "night owl" in three and four classes). Especially the "night owl" agents are the main contributor to the overall performance of the two redistribution methods even though they only make up a small portion of the community. Overall, this shows that the precision of the estimation methods is sensitive to more unusual, rarer demand profiles. However, by definition, such agents will always be a very small minority of the total population, and hence the overall performance should not be substantially affected. It is still an issue that would be interesting to explore in more depth.

Although it approximates the exact Shapley values very well, a potential disadvantage of the RL-based method (and any method using random sampling) aside from its time complexity is that the redistributed values can vary every time the algorithm is run. In practice, the random output of the method can have an undesirable effect on the *perceived fairness* of the redistribution, as prosumers with the same demand profile can result in being assigned slightly different costs.

9

Conclusions & Further Work

While the use of the Shapley value is increasingly popular in energy systems, previous works often sidestep the issue of how it can be efficiently computed in large, realistically-sized settings. The issue is made more pressing by the increasing popularity of community energy projects, where prosumers share joint renewable generation and storage assets and costs.

This study aims to close this gap by proposing a new method to efficiently approximate the Shapley value, and characterizing both their computational complexity and performance (in terms of distance to the exact Shapley value), using large-scale, realistic case studies of energy communities in the UK. We compare the performance of the new method with an already-existing deterministic method and a non-deterministic, state-of-the-art sampling method. Moreover, in order to develop a "ground truth" benchmark to compare these approximations, we propose a novel method to compute the Shapley value *exactly* even for large population sizes by clustering agents into a smaller number of consumption profiles or classes.

Our experimental analysis shows that the relative difference to the true Shapley (while large for a few agents) converges to under 1% for larger scenarios, basically for all methods considered. In particular, in almost all scenarios studied, the newly proposed stratified expected value method and the state-of-the-art adaptive sampling method perform extremely close to true Shapley values. Interesting to observe that the stratified expected value method performs comparatively with the adaptive sampling method [40] for large populations, although its computational cost is lower. In fact, the stratified expected values method outperformed the adaptive sampling method when the community was concentrated to one class, showing a high potential for application in real-world energy communities.

There are a number of directions we find promising to explore in future work. On the experimental side, we plan to explore a larger number of classes in the community, as well as more scenarios (e.g., different compositions as well as different consumption quantities). This will enable us to further validate the robustness of our Shapley approximation methods on a wide variety of demand profiles. On the modelling side, an interesting question to explore is the case when the local distribution network where the energy community is based is subject to physical capacity constraints (voltage, power) [38]. Such constraints could potentially restrict all prosumers from participating in the scheme equally at certain times and would lead to changes in the coalitional game, as well as in the computation of fair redistribution payments based on the Shapley value.

References

[1] Dahlan Abdullah et al. “The application of K-means clustering for province clustering in Indonesia of the risk of the COVID-19 pandemic based on COVID-19 data”. In: *Quality & Quantity* 56.3 (2022), pp. 1283–1291.

[2] Muddasser Alam, Sarvapali D. Ramchurn, and Alex Rogers. “Cooperative Energy Exchange for the Efficient Use of Energy and Resources in Remote Communities”. In: *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*. AAMAS ’13. 2013, pp. 731–738.

[3] Hortensia Amaris et al. “Loss Allocation in Distribution Networks Based on Aumann–Shapley”. In: *IEEE Transactions on Power Systems* 33.6 (2018), pp. 6655–6666. doi: 10.1109/TPWRS.2018.2844740.

[4] Ehsan Azad-Farsani et al. “Stochastic LMP (Locational marginal price) calculation method in distribution systems to minimize loss and emission based on Shapley value and two-point estimate method”. In: *Energy* 107 (2016), pp. 396–408. issn: 0360-5442. doi: https://doi.org/10.1016/j.energy.2016.04.036. url: https://www.sciencedirect.com/science/article/pii/S036054421630442X.

[5] Haris Aziz and Bart de Keijzer. “Shapley meets shapley”. In: *arXiv preprint arXiv:1307.0332* (2013).

[6] Yoram Bachrach et al. “Approximating power indices: Theoretical and empirical analysis”. In: *Autonomous Agents and Multi-Agent Systems* 20 (Mar. 2010), pp. 105–122. doi: 10.1007/s10458-009-9078-9.

[7] Salma Bakr and Stephen Cranefield. “Using the Shapley Value for Fair Consumer Compensation in Energy Demand Response Programs: Comparing Algorithms”. In: *2015 IEEE International Conference on Data Science and Data Intensive Systems*. 2015, pp. 440–447. doi: 10.1109/DSDIS.2015.83.

[8] Smriti Bhagat et al. “The shapley value in knapsack budgeted games”. In: *International Conference on Web and Internet Economics*. Springer. 2014, pp. 106–119.

[9] René van den Brink. “An axiomatization of the Shapley value using a fairness property”. In: *International Journal of Game Theory* 30.3 (2002), pp. 309–319.

[10] Mark Burgess and Archie Chapman. “Approximating the Shapley Value Using Stratified Empirical Bernstein Sampling”. In: Aug. 2021, pp. 73–81. doi: 10.24963/ijcai.2021/11.

[11] A Caramizaru and A Uihlein. “Energy Communities: An Overview of Energy and Social Innovation; JRC Science for Policy Report JRC119433”. In: *Publications Office of the European Union: Luxembourg, Luxembourg* (2020).

[12] Javier Castro, Daniel Gómez, and Juan Tejada. “Polynomial calculation of the Shapley value based on sampling”. In: *Computers & Operations Research* 36.5 (2009), pp. 1726–1730.

[13] Javier Castro et al. “Improving polynomial estimation of the Shapley value by stratified random sampling with optimum allocation”. In: *Computers & Operations Research* 82 (2017), pp. 180–188.

[14] G. Chalkiadakis, E. Elkind, and M. Wooldridge. *Computational Aspects of Cooperative Game Theory*. Morgan Claypool, 2011.

[15] Adriana Chiş and Visa Koivunen. “Coalitional Game-Based Cost Optimization of Energy Portfolio in Smart Grid Communities”. In: *IEEE Transactions on Smart Grid* 10.2 (2019), pp. 1960–1970. doi: 10.1109/TSG.2017.2784902.

[16] European Commission. *COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS 'Fit for 55': delivering the EU's 2030 Climate Target on the way to climate neutrality*. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021DC0550. July 2021.

[17] Jovi D'Silva and Uzzal Sharma. "Unsupervised automatic text summarization of Konkani texts using K-means with Elbow method". In: *Int J Eng Res Technol* (2020), pp. 2380–2384.

[18] S.D. Downing and D.F. Socie. "Simple rainflow counting algorithms". In: *International Journal of Fatigue* 4.1 (1982), pp. 31–40. issn: 0142-1123. doi: https://doi.org/10.1016/0142-1123(82)90018-4. url: https://www.sciencedirect.com/science/article/pii/0142112382900184.

[19] Enercon. *ENERCON Wind energy converters Product overview*. July 2010. url: https://docs.wind-watch.org/Enercon.pdf.

[20] Centre for Environmental Data Analysis. In: (2019). url: https://catalogue.ceda.ac.uk/uuid/dbd451271eb04662beade68da43546e1.

[21] Thomas Feldhoff. "Asset-based community development in the energy sector: energy and regional policy lessons from community power in Japan". In: *International Planning Studies* 21.3 (May 2016), pp. 261–277. doi: 10.1080/13563475.2016.1185939. url: https://doi.org/10.1080/13563475.2016.1185939.

[22] Andreas Fleischhacker et al. "Stabilizing Energy Communities Through Energy Pricing or PV Expansion". In: *IEEE Transactions on Smart Grid* 13.1 (2022), pp. 728–737. doi: 10.1109/TSG.2021.3121283.

[23] GOV.UK. *Biggest ever renewable energy support scheme opens*. [Online; accessed 2022-01-21]. Dec. 2021. url: https://www.gov.uk/government/news/biggest-ever-renewable-energy-support-scheme-opens.

[24] Liyang Han, Thomas Morstyn, and Malcolm McCulloch. "Estimation of the Shapley Value of a Peer-to-peer Energy Sharing Game Using Multi-Step Coalitional Stratified Sampling". In: *International Journal of Control, Automation and Systems* 19.5 (Feb. 2021), pp. 1863–1872. doi: 10.1007/s12555-019-0535-1. url: https://doi.org/10.1007/s12555-019-0535-1.

[25] Apple Inc. 2022. url: https://darksky.net/dev.

[26] K. G. Janardan and G. P. Patil. "A Unified Approach for a Class of Multivariate Hypergeometric Models". In: *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)* 34.4 (1972), pp. 363–376. issn: 0581572X. url: http://www.jstor.org/stable/25049821.

[27] Hyun Cheol Jeong et al. "Clustering of Load Profiles of Residential Customers Using Extreme Points and Demographic Characteristics". In: *Electronics* 10.3 (2021). issn: 2079-9292. doi: 10.3390/electronics10030290. url: https://www.mdpi.com/2079-9292/10/3/290.

[28] Weizheng Kong et al. "Analysis of energy consumption structure based on K-means clustering algorithm". In: *E3S Web of Conferences*. Vol. 267. EDP Sciences. 2021, p. 01054.

[29] Jungsuk Kwac, June Flora, and Ram Rajagopal. "Household Energy Consumption Segmentation Using Hourly Data". In: *IEEE Transactions on Smart Grid* 5.1 (2014), pp. 420–430. doi: 10.1109/TSG.2013.2278477.

[30] You Li et al. "Techno-economic performance of battery energy storage system in an energy sharing community". In: *Journal of Energy Storage* 50 (2022), p. 104247. issn: 2352-152X. doi: https://doi.org/10.1016/j.est.2022.104247. url: https://www.sciencedirect.com/science/article/pii/S2352152X22002766.

[31] David Liben-Nowell et al. "Computing Shapley Value in Supermodular Coalitional Games". In: *Computing and Combinatorics*. Ed. by Joachim Gudmundsson, Julián Mestre, and Taso Viglas. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 568–579. isbn: 978-3-642-32241-9.

[32] Fang Liu et al. "Use of latent profile analysis and k-means clustering to identify student anxiety profiles". In: *BMC psychiatry* 22.1 (2022), pp. 1–11.

[33] Sasan Maleki et al. "Bounding the estimation error of sampling-based Shapley value approximation". In: *arXiv preprint arXiv:1306.4265* (2013).

[34] Sasan Maleki et al. “The Shapley value for a fair division of group discounts for coordinating cooling loads”. In: *PloS one* 15.1 (2020), e0227049.

[35] Irwin Mann and Lloyd S. Shapley. *Values of Large Games, IV: Evaluating the Electoral College by Montecarlo Techniques*. Santa Monica, CA: RAND Corporation, 1960.

[36] Esther Mengelkamp et al. “Designing microgrid energy markets: A case study: The Brooklyn Microgrid”. In: *Applied Energy* 210 (2018), pp. 870–880. doi: 10.1016/j.apenergy.2017.06.054.

[37] United Nations. *Renewable energy – powering a safer future*. [Online; accessed 2022-07-02]. May 2022. url: https://www.un.org/en/climatechange/raising-ambition/renewable-energy.

[38] Sonam Norbu et al. “Modeling Economic Sharing of Joint Assets in Community Energy Projects Under LV Network Constraints”. In: *IEEE Access* 9 (2021), pp. 112019–112042. doi: 10.1109/ACCESS.2021.3103480.

[39] Sonam Norbu et al. “Modelling the redistribution of benefits from joint investments in community energy projects”. In: *Applied Energy* 287 (2021), p. 116575. issn: 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2021.116575.

[40] Geaórid O’Brien, Abbas El Gamal, and Ram Rajagopal. “Shapley Value Estimation for Compensation of Participants in Demand Response Programs”. In: *IEEE Transactions on Smart Grid* 6.6 (2015), pp. 2837–2844. doi: 10.1109/TSG.2015.2402194.

[41] Met Office. *Met Office MIDAS Open: UK Land Surface Stations Data (1853-current)*. https://catalogue.ceda.ac.uk/uuid/dbd451271eb04662beade68da43546e1. [Online; accessed 2022-05-31]. 2019.

[42] S. Ould Amrouche et al. “Overview of energy storage in renewable energy systems”. In: *International Journal of Hydrogen Energy* 41.45 (2016), pp. 20914–20927. issn: 0360-3199. doi: https://doi.org/10.1016/j.ijhydene.2016.06.243. url: https://www.sciencedirect.com/science/article/pii/S0360319916309478.

[43] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[44] Farzaneh Pourahmadi and Payman Dehghanian. “A Game-Theoretic Loss Allocation Approach in Power Distribution Systems with High Penetration of Distributed Generations”. In: *Mathematics* 6.9 (2018). issn: 2227-7390. doi: 10.3390/math6090158. url: https://www.mdpi.com/2227-7390/6/9/158.

[45] Giuseppe Prettico et al. *DISTRIBUTION SYSTEM OPERATORS OBSERVATORY - European Electricity Distribution Systems to Representative Distribution Networks*. June 2016. doi: 10.2790/701791.

[46] Sarvapali D. Ramchurn et al. “Agent-Based Control for Decentralised Demand Side Management in the Smart Grid”. In: *The 10th International Conference on Autonomous Agents and Multiagent Systems*. AAMAS ’11. 2011, pp. 5–12.

[47] Valentin Robu et al. “Efficient Buyer Groups With Prediction-of-Use Electricity Tariffs”. In: *IEEE Transactions on Smart Grid* 9.5 (2018), pp. 4468–4479. doi: 10.1109/TSG.2017.2660580.

[48] Amir Safdarian et al. “Coalitional Game Theory Based Value Sharing in Energy Communities”. In: *IEEE Access* 9 (2021), pp. 78266–78275. doi: 10.1109/ACCESS.2021.3081871.

[49] Lloyd S Shapley. “A value for n-person games”. In: *Contributions to the Theory of Games* 2.28 (1953), pp. 307–317.

[50] Satish Sharma and A. R. Abhyankar. “Loss Allocation for Weakly Meshed Distribution System Using Analytical Formulation of Shapley Value”. In: *IEEE Transactions on Power Systems* 32.2 (2017), pp. 1369–1377. doi: 10.1109/TPWRS.2016.2571980.

[51] Satish Sharma and A.R. Abhyankar. “Loss allocation of radial distribution system using Shapley value: A sequential approach”. In: *International Journal of Electrical Power & Energy Systems* 88 (2017), pp. 33–41. issn: 0142-0615. doi: https://doi.org/10.1016/j.ijepes.2016.11.015. url: https://www.sciencedirect.com/science/article/pii/S0142061516311310.

[52] Amit Singh et al. “Shapley Value Method and Stochastic Dantzig–Wolfe Decomposition for Decentralized Scheduling of Multimicrogrid”. In: *IEEE Systems Journal* (2021), pp. 1–13. doi: 10.1109/JSYST.2021.3119426.

[53] H. Singh, S. Hao, and A. Papalexopoulos. “Transmission congestion management in competitive electricity markets”. In: *IEEE Transactions on Power Systems* 13.2 (1998), pp. 672–680. doi: 10.1109/59.667399.

[54] Tiago Sousa et al. “Peer-to-peer and community-based markets: A comprehensive review”. In: *Renewable and Sustainable Energy Reviews* 104 (2019), pp. 367–378. issn: 1364-0321. doi: 10.1016/j.rser.2019.01.036.

[55] Moiz Masood Syed, Paula Hansen, and Gregory M. Morrison. “Performance of a shared solar and battery storage system in an Australian apartment building”. In: *Energy and Buildings* 225 (2020), p. 110321. issn: 0378-7788. doi: https://doi.org/10.1016/j.enbuild.2020.110321. url: https://www.sciencedirect.com/science/article/pii/S0378778820312044.

[56] Wayes Tushar et al. “Peer-to-peer energy systems for connected communities: A review of recent advances and emerging challenges”. In: *Applied Energy* 282 (2021), p. 116131. issn: 0306-2619. doi: 10.1016/j.apenergy.2020.116131.

[57] Solar Energy UK. *Smart Export Guarantee*. [Online; accessed 2022-07-02]. Feb. 2022. url: https://solarenergyuk.org/resource/smart-export-guarantee/?cn-reloaded=1.

[58] Bregje van Veelen. “Making Sense of the Scottish Community Energy Sector – An Organising Typology”. In: *Scottish Geographical Journal* 133.1 (Aug. 2016), pp. 1–20. doi: 10.1080/14702541.2016.1210820. url: https://doi.org/10.1080/14702541.2016.1210820.

[59] Niklas Vespermann, Thomas Hamacher, and Jalal Kazempour. “Access Economy for Storage in Energy Communities”. In: *IEEE Transactions on Power Systems* 36.3 (2021), pp. 2234–2250. doi: 10.1109/TPWRS.2020.3033999.

[60] Simon Voswinkel et al. “Sharing congestion management costs among system operators using the Shapley value”. In: *Applied Energy* 317 (2022), p. 119039. issn: 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2022.119039. url: https://www.sciencedirect.com/science/article/pii/S030626192200441X.

[61] Hongfei Xiao and Weidong Li. “Allocation of Congestion Cost in a Pool Based Model Using Shapley Value”. In: *2009 Asia-Pacific Power and Energy Engineering Conference*. 2009, pp. 1–5. doi: 10.1109/APPEEC.2009.4918127.

[62] Bolun Xu et al. “Modeling of Lithium-Ion Battery Degradation for Cell Life Assessment”. In: *IEEE Transactions on Smart Grid* 9.2 (2018), pp. 1131–1140. doi: 10.1109/TSG.2016.2578950.

[63] Zhi Zhang et al. “Optimal Alliance Strategies Among Retailers Under Energy Deviation Settlement Mechanism in China’s Forward Electricity Market”. In: *IEEE Transactions on Power Systems* 35.3 (2020), pp. 2059–2071. doi: 10.1109/TPWRS.2019.2955479.

[64] Kai-le Zhou, Shan-lin Yang, and Chao Shen. “A review of electric load classification in smart grid environment”. In: *Renewable and Sustainable Energy Reviews* 24 (2013), pp. 103–110. issn: 1364-0321. doi: https://doi.org/10.1016/j.rser.2013.03.023. url: https://www.sciencedirect.com/science/article/pii/S1364032113001822.