



DEMAND RESPONSE DRIVEN LOAD SCHEDULING IN FORMAL SMART GRID FRAMEWORK

Electrical and Computer Engineering
Technical Report ECE-TR-24



AARHUS
UNIVERSITY
DEPARTMENT OF ENGINEERING

DATA SHEET

Title: Demand Response Driven Load Scheduling in Formal Smart Grid Framework

Subtitle: Electrical and Computer Engineering

Series title and no.: Technical report ECE-TR-24

Author: Armin Ghasem Azar

Department of Engineering – Electrical and Computer Engineering,
Aarhus University

Internet version: The report is available in electronic format (pdf) at the Department of Engineering website <http://www.eng.au.dk>.

Publisher: Aarhus University©

URL: <http://www.eng.au.dk>

Year of publication: 2016 Pages: 27

Editing completed: March 2016

Abstract: In this technical report, we present the current state of the research conducted during the first part of the PhD project named "Demand Response Driven Load Scheduling in Formal Smart Grid Framework". The PhD project focuses on smart grids which employ information and communication technologies to assist the electricity production, distribution, and consumption. Designing smart grid applications is a novel challenging task that requires modeling, integrating, and validating different grid aspects in an efficient way. The project tackles such challenges by proposing an effective framework to formally describe smart grid elements along with their interactions. To validate this framework, the report concentrates on deploying efficiency in managing the electricity consumption in households which requires focusing on different impacts of demand response programs running in the smart grid to engage consumers to participate. A demand response system is considered which is connected to all households and utilizes their information to determine an effective load management strategy taking into account the grid constraints imposed by distribution system operators. The main responsibility of the demand response system is scheduling the operation of appliances of a large number of consumers in order to achieve a network-wide optimized performance. Finally, the PhD report demonstrates the simulation results, publications, courses, and dissemination activities done during this period. They are followed by envisaging future plans that will lead to completion of the PhD study.

Keywords: Smart Grid, formal framework, formal modeling, UML profile, demand response, load management, load scheduling, optimization, multi-objective optimization, evolutionary computation.

Supervisor: Rune Hylsberg Jacobsen

Financial support: The research leading to publishing this document has received funding from the European Union Seventh Framework program (FP7/2007-2013) under grant agreement no 619560 (SEMI-AH).

Please cite as: Armin Ghasem Azar, 2016. Demand Response Driven Load Scheduling in Formal Smart Grid Framework. Department of Engineering, Aarhus University, Denmark. 27 pp. - Technical report ECE-TR-24

Cover image: Armin Ghasem Azar

ISSN: 2245-2087

Reproduction permitted provided the source is explicitly acknowledged

DEMAND RESPONSE DRIVEN LOAD SCHEDULING IN FORMAL SMART GRID FRAMEWORK

Armin Ghasem Azar
Aarhus University, Department of Engineering

Abstract

In this technical report, we present the current state of the research conducted during the first part of the PhD project named "Demand Response Driven Load Scheduling in Formal Smart Grid Framework". The PhD project focuses on smart grids which employ information and communication technologies to assist the electricity production, distribution, and consumption. Designing smart grid applications is a novel challenging task that requires modeling, integrating, and validating different grid aspects in an efficient way. The project tackles such challenges by proposing an effective framework to formally describe smart grid elements along with their interactions. To validate this framework, the report concentrates on deploying efficiency in managing the electricity consumption in households which requires focusing on different impacts of demand response programs running in the smart grid to engage consumers to participate. A demand response system is considered which is connected to all households and utilizes their information to determine an effective load management strategy taking into account the grid constraints imposed by distribution system operators. The main responsibility of the demand response system is scheduling the operation of appliances of a large number of consumers in order to achieve a network-wide optimized performance. Finally, the PhD report demonstrates the simulation results, publications, courses, and dissemination activities done during this period. They are followed by envisaging future plans that will lead to completion of the PhD study.

Table of Contents

List of Figures	iii
List of Tables	iii
List of Algorithms	iii
Acronyms	iv
Chapter 1 Introduction	1
1.1 Field of Research	1
1.2 Project Aim: Hypothesis and Purpose	2
1.3 Document Structure	3
Chapter 2 Background	4
2.1 What is SEMIAH?	4
2.1.1 SEMIAH System Model	4
2.1.2 Aggregative Scheduling	5
2.2 Formal Framework Background	5
2.2.1 Hardware Aspect	6
2.2.2 Software Aspect	6
2.2.3 Network Aspect	7
2.2.4 Modeling: UML/SmartGrid Profile	7
Chapter 3 Formal Smart Grid Framework and Load Scheduling Problem	8
3.1 Smart Grid Formalization	8
3.1.1 Customers Domain	8
3.1.2 Operations Domain	10
3.1.3 Modeling: UML/SmartGrid Class diagram	10
3.2 Load Scheduling Algorithm	10
3.2.1 Event Selection Problem	13
3.2.1.1 Single-objective Optimization: Dynamic Programming	14
3.2.1.2 Multi-Objective Optimization: Evolutionary Algorithm	14
Chapter 4 Simulation Analysis and Future Plans	15
4.1 Simulation Analysis	15
4.1.1 Simulation Setup	15
4.1.2 Simulation Analysis	16
4.2 Future Plans	20
4.2.1 Intelligent Grid Constraints	20
4.2.2 Load Scheduling Algorithms in the Pilot-Testing Households	20
4.2.3 Flexibility Trading Potential in the Electricity Market	20
4.2.4 Decentralizing Load Scheduling Algorithms	21
4.2.5 Time planning	21

Table of Contents

Bibliography	22
A Publications	24
A.1 Scientific Publications	24
A.1.1 Published	24
A.1.1.1 Conferences	24
A.1.2 Submitted	24
A.1.2.1 Journals	24
A.1.2.2 Conferences	24
A.1.3 Planned	25
A.1.3.1 Journals	25
A.1.3.2 Conferences	25
A.2 SEMIAH Project Deliverables	25
A.2.1 Published	25
A.2.2 Submitted	25
B Courses	26
B.1 Scientific Courses	26
B.2 Transferable Skill Courses	26
C Dissemination Activities	27
C.1 Performed	27
C.2 Planned	27

List of Figures

Fig. 1.1	Top-down design methodology of the formal framework	2
Fig. 1.2	Conceptual view of the load scheduling problem	2
Fig. 2.1	SEMIAH technical architecture	5
Fig. 2.2	UML profile diagram of the framework’s aspects	7
Fig. 3.1	UML class diagram of the framework’s elements	11
Fig. 3.2	Flowchart of NSGA2	14
Fig. 3.3	A Pareto-front	14
Fig. 4.1	Conceptual view of the framework as a load scheduling problem	16
Fig. 4.2	Aggregated load consumption of 100 customers in one minute time resolution	16
Fig. 4.3	Peak Demand Reduction percentage in three different flexibility scenarios	17
Fig. 4.4	Aggregated load consumption before and after scheduling	18
Fig. 4.5	load consumption of a random household before and after scheduling	18
Fig. 4.6	Customer Flexibility Usage percentage in three different flexibility scenarios	18
Fig. 4.7	Appliance Serving Delay percentage in three different flexibility scenarios	19
Fig. 4.8	Daily average Customer Electricity Cost in three different flexibility scenarios	19
Fig. 4.9	Average Scheduling Computation Time in three different flexibility scenarios	20
Fig. 4.10	Gantt diagram of the future plans	21

List of Tables

4.1	Setting of framework elements	16
4.2	Peak-to-Average Ratio in three different flexibility scenarios	17

List of Algorithms

3.1	Load scheduling	12
-----	---------------------------	----

Acronyms

AFO Appliance Full Operation	IEC International Electrotechnical Commission
ASD Appliance Serving Delay	MEC Maximum Electricity Consumption
AU Aarhus University	MOEA Multi-Objective Evolutionary Algorithm
CEC Customer Electricity Cost	NIST National Institute of Standards and Technology
CEM CO ₂ Emission Minimization	NSGA2 Non-dominated Sorting Genetic Algorithm-II
CFU Customer Flexibility Usage	OMS Operation Management System
CLM Comfort Level Maximization	PAR Peak-to-Average Ratio
DER Distributed Energy Resource	PDF Peak Demand Flattening
DR Demand Response	PDR Peak Demand Reduction
DRS Demand Response System	QoS Quality of Service
DSO Distribution System Operator	RES Renewable Energy Source
ECM Electricity Cost Minimization	REST REpresentational State Transfer
ECT Electricity Consumption Threshold	SCT Scheduling Computation Time
EFSM Extended Finite State Machine	SEMIAH Scalable Energy Management Infrastructure for Aggregation of Households
EMS Energy Management System	SEP2 Smart Energy Profile 2.0
GSST Graduate School of Science and Technology	SGAM Smart Grid Architecture Model
GWAC GridWise Architecture Council	SysML Systems Modeling Language
HEMG Home Energy Management Gateway	UML Unified Modeling Language
HTTP Hypertext Transfer Protocol	USEF Universal Smart Energy Framework
HVAC Heating, Ventilation, and Air Conditioning	
ICT Information and Communication Technology	

Introduction

The purpose of this report is to present the progress in relation to the PhD project named *Demand Response Driven Load Scheduling in Formal Smart Grid Framework* carried out at the Department of Engineering at Aarhus University (AU). The document describes the work conducted during the first 1.5 years of the PhD and details the future plans envisioned for the remaining time.

1.1 Field of Research

The current structure of the electrical grid is ineffective in responding to the growing demand for electricity. The smart grid aims to revolutionize this structure increasing the use of Information and Communication Technology (ICT) to improve its reliability and efficiency [1]. Before the smart grid becomes fully operational, it requires technological advancements in a number of interdisciplinary perspectives to manage electricity operations in a sustainable and reliable manner. National Institute of Standards and Technology (NIST) has developed a smart grid conceptual model which identifies main smart grid domains describing their stakeholders and feasible communication paths. Domains are customers, markets, service providers, operations, bulk generation, transmission, and distribution. Standardizing and formalizing a smart grid, that has to meet decisive requirements of its domains, is a challenging procedure, especially when engineering approaches are concerned. International Electrotechnical Commission (IEC) has proposed reliable and reproducible standards to make it easy to identify those that are needed for any part of the smart grid. In the same context, Universal Smart Energy Framework (USEF) has recently delivered a common standard clearly specifying interactions and roles of the aforementioned domains [2]. Nevertheless, far too little attention has been paid by smart grid application designers to build a formal framework based on these standards concerning the scalability, interoperability, and updatability of domains.

This PhD project, at the first stage, puts some efforts into proposing a practical and robust formal framework to model smart grid applications, as Figure 1.1 presents. The framework formally describes each domain and its actors on the basis of “separation of concerns” design principle considering three main aspects named *hardware* (HW), *software* (SW), and *network* (NW). Such descriptions can be trajected into formal models using either Unified Modeling Language (UML) or Systems Modeling Language (SysML) techniques. Finally, these models can be converted into executable models, i.e., by following the Model-to-Text transformation approach, such as Python or Matlab code. This PhD project, for simplicity, only relies on the interconnectivity of *customers* and *operations* domains in the smart grid. The former domain enables electricity customers to manage their consumption behaviors while the latter domain supports grid operators to continuously perform ongoing grid stabilization functions. As the second stage, the project continues

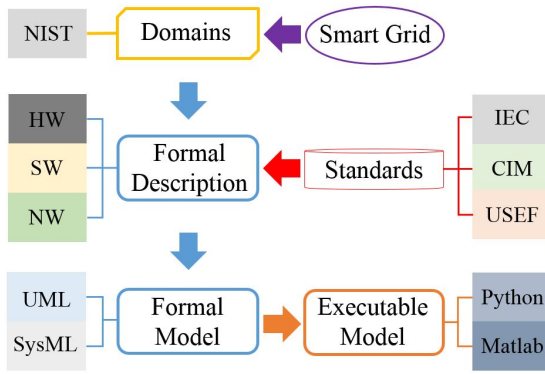


Figure 1.1: Top-down design methodology of the formal framework

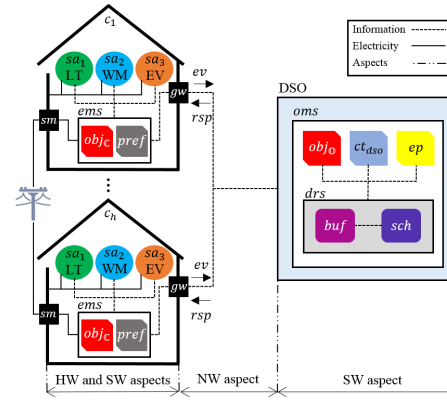


Figure 1.2: Conceptual view of the load scheduling problem

by proposing a Demand Response (DR) framework based on the smart grid formal framework. This DR framework is used to first, validate the usability and applicability of the smart grid formal framework and then, address the load scheduling challenge in DR programs, as Figure 1.2 conceptually pictures. From the customers' perspective, DR programs intend to encourage them to voluntarily modify their daily electricity consumption behavior in order to decrease the peak demand while increasing their comfort level. In contrast to this point of view, these programs help the Distribution System Operators (DSOs), as the main grid operators, to equilibrate demands with responses to flatten electricity peak periods as much as possible [3]. Therefore, existence of a robust and efficient *load scheduling* approach for DR programs is necessary.

This PhD project provides a high-potential and scalable load scheduling approach to flatten the peak demand and increase the customer satisfaction considering physical grid stability constraints. It discusses not only customers benefit from participating in DR programs, but also, grid operators invest on maintaining the grid functionality efficiently. The DSO employs a Demand Response System (DRS) which receives and schedules a large number of customers' load requests. Its main advantage is its ability to streamline the control of received load requests while optimally schedule them in each time interval by decreasing the peak-to-average ratio. This DRS uses an efficient load scheduling optimization algorithm to shift or to interrupt demands to flatten the aggregated consumption, where each customer has a desirable usage scenario of his/her appliances. Although customers provide flexibility to the DRS, they are not interested in waiting too long to receive their appliances in the completed status. This flexibility is defined for both physically-controllable (e.g., washing machine) and thermostatically-controllable (e.g., Heating, Ventilation, and Air Conditioning (HVAC)) appliances. All together, the PhD project is dealt with a multi-objective multi-constraint load scheduling optimization problem. To this end, it uses multi-objective optimization techniques to provide a set of feasible load scheduling solutions to the DSO. According to miscellaneous circumstances, the DSO can choose a different strategy toward communicating with customers in the course of scheduling their appliances.

1.2 Project Aim: Hypothesis and Purpose

The main goal of this PhD project is to provide insights in formalizing a smart grid, designing an efficient DR framework, and developing a scalable load scheduling structure. The underlying hypothesis of the PhD project is:

It is my hypothesis that a scalable load scheduling structure (centralized and/or distributed) deployed on an efficient ICT-driven formal smart grid framework, can coordinate and control a large group of customers' appliances using adequate optimization algorithms. This is enabled as a business model provisioned to the electricity market through an aggregator in the context of a DR-driven formal smart grid framework.

The raised hypothesis will be validated by:

- Formalizing, developing, and evaluating a smart grid framework;
- Materializing a robust DR program from the ICT point of view;
- Proposing and evaluating suitable load scheduling algorithms to employ in DR programs;
- Studying the state of art in methods for controlling the appliances, evaluating these methods, and proposing a scalable and optimal load scheduling structure;
- Analyzing and evaluating the possibilities of trading flexibilities, customers provide through DR programs, in the electricity market via an aggregator.

Thus, the PhD project advances the state of the art as follows:

- Proposing a robust formal framework for modeling smart grid applications;
- Providing a scalable load scheduling framework based on ICT-driven DR programs;
- Applying the full grid topology on the frameworks to integrate various grid stability constraints with the load scheduling approach;
- Adapting the frameworks to a wide range of multi-class varied-specific smart appliances;
- Defining various DR flexibility types considering characteristics of appliances;
- Proposing and evaluating different load scheduling methods through simulation toolboxes;
- Defining a novel consecutive event-based appliance load scheduling algorithm to make DR programs automated.

1.3 Document Structure

The document is organized in four chapters and three appendices, as follows:

Chapter 1 - Introduction: Introduces the research field, presents the purpose of the PhD project, and outlines the document structure.

Chapter 2 - Background: Presents an overview of the SEMIAH project and the contribution of the current PhD project to it.

Chapter 3 - Load Scheduling Optimization Problem in DR Programs: Describes the contributions conducted in the first 1.5 years of the PhD program.

Chapter 4 - Current Results and Future Plans: Presents simulation results achieved so far and provides the future plans for completion of the PhD project.

Appendix A - Publications: Outlines the publications accepted, submitted and planned related to the PhD project.

Appendix B - Courses: Describes the courses carried out during the first 1.5 years of the PhD.

Appendix C - Dissemination Activities: Lists the activities performed in and planned for the PhD project.

Background

This chapter is divided into two sections. The first section provides the background information necessary for understanding the SEMIAH project which the current PhD project aims to contribute to it. The second section presents a brief background in relation to the smart grid formal framework developed in the current PhD project.

2.1 What is SEMIAH?

With the advent of smart grids, new solutions for energy managements become available [4]. During the last decade, manufacturers have focused on the development of smart appliances. However, not only a large market uptake of smart appliances is not expected to occur in the short-term, but also, no automated DR programs have been implemented for European households despite the fact that households represented approximately 27% of the total energy consumption in Europe in 2010 and were responsible for 10% of the carbon dioxide emissions in 2007. The consortium behind *Scalable Energy Management Infrastructure for Aggregation of Households (SEMIAH)* project is interested in pursuing a major technological, scientific and commercial breakthrough by developing a novel ICT infrastructure for the implementation of DR programs in households [5]. The Scalable Energy Management Infrastructure for Aggregation of Households (SEMIAH) infrastructure enables the shifting of energy consumption of high energy-consuming loads to off-peak periods with high generation of electricity from Renewable Energy Sources (RESs). The SEMIAH consortium will develop a novel solution for households, where the central aggregator system will simultaneously optimize and manage a large number of electricity consumption loads according to the generation of electricity from RES (bulk or Distributed Energy Resources (DERs)). To the best of the consortium's knowledge, this will imply a step-change innovation in the field where there are currently no similar solutions.

2.1.1 SEMIAH System Model

The SEMIAH concept will enable aggregation of all households connected to the system and will act through direct load control to remotely shift or curtail electrical loads in a secure manner taking the privacy and flexibility of customers into account. Until now, implementation of DR has been strongly inhibited by the following barriers: System Cost and Complexity, Lack of ICT infrastructure and aggregators, Lack of clear business models for DR systems. These challenges must be overcome to ensure the deployment of technologies to efficiently and securely manage energy consumption in households so as to significantly increase the substitution of conventional generation (fossil fuels-based) with RES and in order to reduce/shift peak loads. Figure 2.1 demonstrates the

SEMIAH technical architecture. SEMIAH is characterized by a *back-end* system, a *Home Energy Management Gateway (HEMG)*, and a *User Interface*. The last two elements represent the front-end system of SEMIAH. This PhD project endeavors to contribute to *SEMIAH Algorithms* and *DSO* parts as components of the back-end system [5].

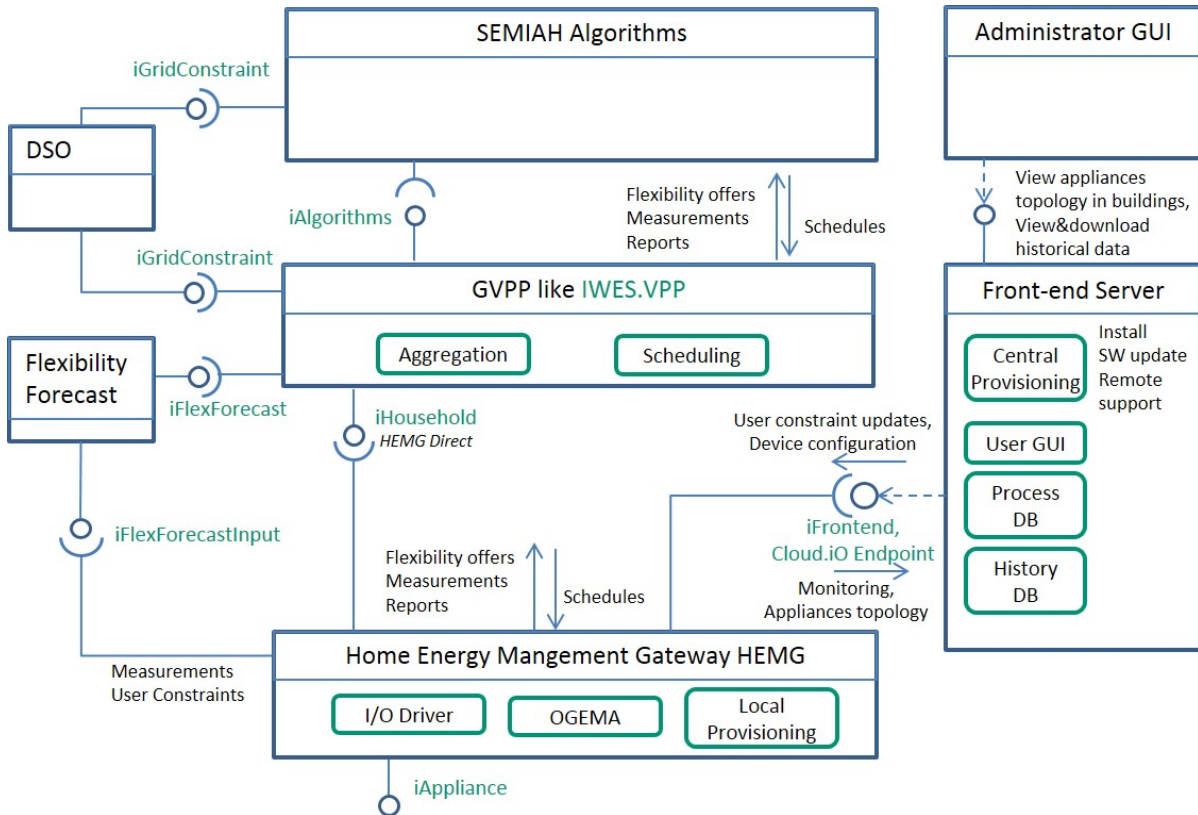


Figure 2.1: SEMIAH technical architecture

2.1.2 Aggregative Scheduling

The back-end infrastructure is built on a central server that registers and manages the flexible electricity consumptions offered by the customers at the front-end. It provides an interface towards the front-end and is the engine of the system operations. Customers register electrical loads which are subjected to intelligent load control. Load planning and scheduling are based on the aggregation of electrical loads of customers in “DR ready” mode. When the customer chooses to operate an appliance in “DR ready” mode the customer is offering flexibility to the grid and allowing the SEMIAH Back-end system to take control of the appliance, e.g., decide when to run the appliance. Restrictions from DSO grid management and market energy prices are also taken into account. Since customers can decide to shift between modes in real time, the optimization should also occur continuously. This leads to a rather complex optimization problem that has to satisfy both the flexibility constraints of the customer as well as the needs or offers of the DSO and which also has to be solved in real time. For further investigations, the readers are referred to [6].

2.2 Formal Framework Background

This section sets the bases for building a formal framework to model, simulate and validate smart grid applications. The proposed framework is based on the “separation of concerns” design principle

that allows reusability, development, and upgrading to its components independently. Therefore, the framework orchestrates the smart grid system using three main aspects; hardware, software, and network. The framework is formalized using grid component definitions obtained from common standards series. It follows the International Electrotechnical Commission (IEC) standards and the architecture guidelines of the future smart grid from GridWise Architecture Council (GWAC) [7] with its adapted Smart Grid Architecture Model (SGAM) [8]. They organize how application characteristics exchange data model specifications among described aspects.

2.2.1 Hardware Aspect

The smart grid comprises miscellaneous devices as constitutes of the hardware aspect. Devices can be digital, analog, or heterogeneous with discrete and/or continuous behavior considering their different structures and responsibilities, for instance a smart meter. These devices are employed in various active/passive domains throughout the electricity distribution grid. Hence, it is indispensable to understand the general device constituents. This helps the formal framework to employ different types of devices comprising varied-specific communication media (network aspect). Equation (2.1) defines the device entity:

$$\begin{aligned}
 dev = [dig, ana] \in \mathbb{D} \text{ where:} & \quad comp = [r, q] \text{ where:} & \quad comm = [i, z] \text{ where:} \\
 dig = [comp, comm], & \quad r = [f_1, f_2, \dots, f_n] \in app, & \quad i = [elec, info] \in nw, \\
 ana = [phy, mech]. & \quad q = [o_1, o_2, \dots, o_m] \in \mathbb{R}_0^+. & \quad z = [e_1, e_2, \dots, e_v] \in \mathbb{R}_0^+.
 \end{aligned}
 \tag{2.1} \qquad \tag{2.2} \qquad \tag{2.3}$$

Each device dev is a member of a multiset of devices \mathbb{D} , where it includes digital dig and analog ana components. A digital component consists of computational $comp$ and communication $comm$ components. An analog component ana in each device dev consists of physical phy and mechanical $mech$ components (out of the scope of this PhD project). The computational component $comp$ includes computation r and overhead q vectors. The former includes n function elements. Each function element, as a software application app (will be described later), performs a specific procedure running in a hardware entity. The latter contains m overhead elements. Each overhead element represents the processing time of a subset of functions. The communicational component $comm$ includes vectors of communication interfaces i and communication overheads z . The first vector, as a member of a network component nw (will be described later), includes electricity $elec$ and information $info$ elements. The former is responsible for satisfying the electricity demand while the latter is performed on top of a communication protocol. Accordingly, the second vector comprises v communication overhead elements, each is caused by characteristics of the communication media, imposed by each device-to-device connection.

2.2.2 Software Aspect

This aspect aims to describe smart grid applications independently from the specific technological platform, as Equation (2.4) defines:

$$\begin{aligned}
 app = [sv, bv] \in Apps \text{ where:} & \\
 sv = [en, re] \in SV, & \tag{2.4} \\
 bv = [s, \tau] \in BV. &
 \end{aligned}$$

Each application app , as a member of a multiset of applications $Apps$, consists of two correlated structural sv and behavioral bv views. The former describes the structure of the application while the latter presents its dynamics. Structural view sv , as a member of a multiset of structural views SV , is a combination of entities en and relationships re . An entity represents the functionality part of an application which can be periodic or aperiodic. A relationship defines the logical connection

between entities. Afterwards, the behavioral view bv , as a member of a multiset of behavioral views BV , represents the dynamical aspect of an application, in which it complements the application structure. An Extended Finite State Machine (EFSM) can capture such behavior. Here, a set of states s describes the application's actions while a set of transitions τ provides conditional paths between them. The detailed description of EFSM can be found in [9].

2.2.3 Network Aspect

This aspect enables the communication link between two or more smart grid device entities dev . Equation (2.5) formulates this aspect:

$$\begin{aligned}
 nw &= [QoS, dist, mob] \in NW \text{ where:} \\
 QoS &= [x_1, x_2, \dots, x_y] \in \mathbb{R}_0^+, \\
 dist &\in \mathbb{R}_0^+, mob \in \mathbb{B}.
 \end{aligned}
 \tag{2.5}$$

Each network component nw , as a member of a multiset of network components NW possesses some elements. The vector Quality of Service (QoS) includes y elements of parameters x to represent the overall performance of the communication network. The distance $dist$ indicates the topological distance value of two connected devices. Finally, mobility mob represents the network mobility type (wired or wireless).

2.2.4 Modeling: UML/SmartGrid Profile

Figure 2.2 demonstrates a novel UML profile of the framework aspects. The semantic of three UML metaclasses, i.e., $\ll Device \gg$, $\ll Artifact \gg$, and $\ll CommunicationPath \gg$, has been studied and extended by the predefined UML elements. Due to the lack of overhead data types of the UML, new data types have been created to describe the computation and communication overheads.

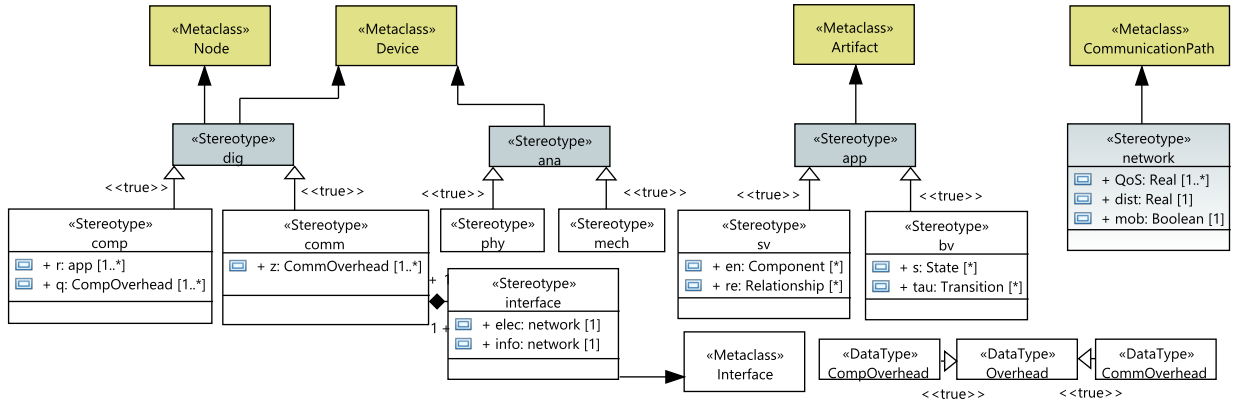


Figure 2.2: UML profile diagram of the framework's aspects

Formal Smart Grid Framework and Load Scheduling Problem

This chapter first describes the formal framework based on the aspects defined in Section 2.2. Then, it presents the DR-driven load scheduling problem and proposes a load scheduling algorithm.

3.1 Smart Grid Formalization

The smart grid formalization requires a high-level conceptual framework taking its domains into account. This paper serves the formal framework as an application to appropriately identify the actors inside each smart grid domain and establish their possible communication routes. Equation (3.1) defines a smart grid:

$$\begin{aligned} sg &= [w] \in \Psi \text{ where:} \\ w &\subseteq \omega. \end{aligned} \quad (3.1)$$

A smart grid sg , as an entity, is a member of a multiset of smart grids Ψ . A set of smart grid domains w is a member of a multiset of smart grid domains ω . A major challenge in relation to these domains is to organize them to work consistently focusing on delivering correct services to their relevant interior actors. As we employ the concept of the separation of concerns, each domain corresponds to an *add-in* feature to the framework. Hence, adding and/or removing a domain will not affect the framework's functionality which strengthen its robustness and flexibility. This PhD Project considers $w = [\mathbb{C}, \mathbb{O}]$, where \mathbb{C} and \mathbb{O} correspond to *customers* and *operations* domains, respectively [10].

3.1.1 Customers Domain

This domain typically provides applications to customers to manage their electricity consumption behavior. Equation (3.2) defines this domain:

$$\begin{aligned} \mathbb{C} &= [c_1, c_2, \dots, c_h] \text{ where:} \\ c &= [SA, ems, sm, gw], \\ SA &\subseteq \mathbb{A}, ems \in app, \{sm, gw\} \in dev. \end{aligned} \quad (3.2)$$

This domain includes h customers, where each customer c has an individual set of smart appliances SA as a subset of a multiset of smart appliances \mathbb{A} . Inherently, customers are interested in enhancing the efficiency and profitability of the electricity consumption of their smart appliances. This is done using an Energy Management System (EMS) as a software application. Moreover,

each customer has a smart meter sm installed in the house. The smart meter measures electricity consumption of smart appliances and periodically sends them to the electrical utility via the power line communication. Finally, each customer has a gateway gw that is responsible for routing different device-to-device communications. Since smart meters and gateways are predefined entities, their descriptions are discarded. The following defines SA and ems precisely.

$$\begin{aligned}
 SA &= [sa_1, sa_2, \dots, sa_p] \text{ where:} & ems &= [obj_C, pref^p, ev^p, rsp^p] \text{ where:} \\
 sa &= [sf, ct_{sa}, lp], & obj_C &= [\sigma_1, \sigma_2, \dots, \sigma_t] \in \mathbb{R}_0^+, \\
 sf &= [shift, intr] \in \mathbb{B}, & pref &= [ost, opr, ofl] \in \mathbb{R}_0^+, \\
 ct_{sa} &= [\zeta_1, \zeta_2, \dots, \zeta_a] \in \mathbb{R}_0^+ & ev &= [est, ept, obj_C, pref, sf, ct_{sa}, lp], \\
 lp &= [ec, \Delta \nabla], & \{est, ept\} &\in \mathbb{R}_0^+, \\
 ec &\in \mathbb{R}_0^+, \Delta \nabla \in \mathbb{R}^+. & rsp &= [dec, rst], \\
 & & dec &\in \mathbb{B}, rst \in \mathbb{R}_0^+.
 \end{aligned} \tag{3.3} \tag{3.4}$$

Each customer c possesses p smart appliances, as main drivers of electricity demands. Each smart appliance $sa \in dev$ has a smart feature pair sf including two dependent Boolean features named shiftability $shift$ and interruptibility $intr$ [6]. Shiftability allows smart appliances to shift their operating start times to the future. Interruptibility allows smart appliances to interrupt their operating cycles in the middle. In addition, each smart appliance has a set of constraints ct_{sa} including a constraint elements ζ , e.g., the appliance full operation, technical operation, etc. [6, 11]. Finally, the smart appliance sa , in each operating cycle, follows a specific load profile lp with respect to its program predefined by the corresponding customer. Each lp is presented as a vector of time-series electricity consumptions ec in a specific time resolution $\Delta \nabla$. ems for customers is a software application app running on top of a device dev . Each customer adjusts his/her own set of objectives obj_C including t distinct objectives, for instance minimizing the electricity cost and CO₂ emission, maximizing comfort level, minimizing appliance service delay, etc. In addition, for each smart appliance sa in SA , the customer provides a 3-tuple operating preference $pref$. By operating start time ost customers adjust the time at which they want to operate their smart appliance. By operating program opr customers set a specific program to operate each smart appliance which directly influences the load profile lp . By operating flexibility ofl , customers offer a voluntary flexibility ofl to operate each shiftable smart appliance [6].

The EMS is responsible for sending events ev to the Operation Management System (OMS) of the operations domain (will be described later) according to the individual 3-tuple preference set $pref$ for each smart appliance. Then, it waits until receiving responses rsp of corresponding sent events. Notation est refers to the time at which the event has been sent. In addition, event pooling time ept defines the time at which the relevant smart appliance can wait to receive a response from the EMS after sending the event. If no response is arrived, another event will be sent after ept minutes/seconds. Each event also comprises the objective set obj_C and operating preferences $pref$ defined by customers, smart feature values sf , constraints ct_{sa} , and load profile lp of the corresponding smart appliance. Then, each response is a pair including a Boolean decision value dec and a time rst at which the response has been sent. Decision dec indicates whether the corresponding smart appliance should operate or wait. The network aspect is responsible for sending events from the EMS and receiving the responses from the OMS. Finally, the EMS starts operating smart appliances in accordance to the received responses. These procedures are inspired from a DR communication protocol named Smart Energy Profile 2.0 (SEP2), which has been adopted by the IEEE P2030 group, attempting to formalize the smart grid application requirements reliably including communication and information sharing aspects [12]. Employing the IPv6 protocol for its communication channel confirms its scalability and addressability characteristics. This protocol is built on a REpresentational State Transfer (REST) architecture over the Hypertext Transfer Protocol (HTTP) as a client-server model, in which servers, such as an EMS, provide to the resources, e.g., smart appliances [13].

3.1.2 Operations Domain

This domain manages the movement of electricity. It facilitates the ongoing grid management functions by efficiently maintaining and operating the electricity distribution infrastructure while securely delivering the electricity to customers. This domain includes a software applications Operation Management System (OMS) derived from the IEC 61970 [14], as Equation (3.5) formulates:

$$\begin{aligned}
 oms &= [obj_{\mathbb{O}}, ct_{dso}, ep, drs] \text{ where:} \\
 obj_{\mathbb{O}} &= [\varsigma_1, \varsigma_2, \dots, \varsigma_g] \in \mathbb{R}_0^+, \\
 ct_{dso} &= [\chi_1, \chi_2, \dots, \chi_l] \in \mathbb{R}_0^+, \\
 ep &= [\rho_1, \rho_2, \dots, \rho_d] \in \mathbb{R}_0^+, \\
 drs &= [buf, sch, rsp^p], \\
 buf &= [iw, is, do, dw] \in \mathbb{R}_0^+, \\
 sch &= [obj_{\mathbb{O}}, ct_{dso}, ep, ev^p, \lambda], \\
 \lambda &\in \mathbb{R}^+.
 \end{aligned} \tag{3.5}$$

OMS is responsible for performance monitoring and optimization of the electrical grid, e.g., load balancing and scheduling. Here, DSOs are main actors of this domain. Current shortcomings of the electrical grid motivate them to employ the ICT to react upon the grid information to meet reasonable demands for the distribution of electricity. Each DSO, similar to customers (see Equation (3.4)), has a set of objectives $obj_{\mathbb{O}}$ including g distinguishable objectives ς , e.g., flattening the aggregated electricity consumption, reducing the number of electricity outages, reducing the CO₂ emission, etc. In addition, ct_{dso} corresponds to a set of l grid stability constraints χ that the DSO imposes to the grid, for instance, hard and soft feeder thresholds, active and reactive power flow capacities, etc. The DSO also adjusts a set of electricity prices ep over d time periods daily. Finally, the DSO, in order to be able to respond to the event received from the EMS of customers, employ a DRS. The DRS is a software application app composing of a set of buffers buf , a scheduler sch , and responses rsp . Once an event arrives, it is stored in the immediately wait buffer iw . Then, the scheduler decides to move each to either the immediately start buffer is , decided to operate buffer do , or decided to wait buffer iw . The scheduler sch follows a scheduling strategy λ to make these decisions based upon the objectives, constraints, and electricity prices coupled with the information stored in the events. The scheduling strategy can be either stochastic or deterministic while applying single-objective or multi-objective optimization techniques [15]. The next section proposes a load scheduling algorithm.

3.1.3 Modeling: UML/SmartGrid Class diagram

To integrate several framework elements, explained before, we create a UML class diagram to traject the previously defined elements into UML classes and relationships. As Figure 3.1 illustrates, this diagram, combined with the profile diagram (see Figure 2.2), enables a complete description of the framework. Objects can be instantiated and linked to compose a variety of smart grid applications.

3.2 Load Scheduling Algorithm

The scheduler sch schedules events once they arrive. Its main responsibility is to select some events and allow their corresponding smart appliance to operate. Unselected events are then, responded to wait causing corresponding smart appliances to send new events according to their event pooling time ept value. Therefore, as the main novelty of the proposed load scheduling algorithm, it is not necessary for it to forecast the future or be aware of the whole operating period of smart appliances. As Algorithm 3.1 presents, once an event arrives, it is located in immediately wait buffer iw . Then,

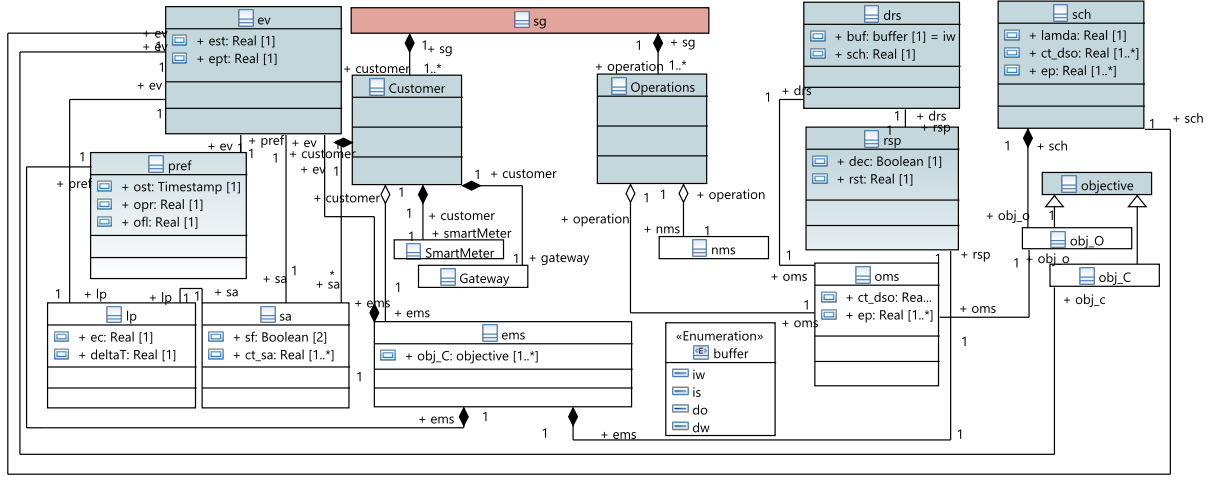


Figure 3.1: UML class diagram of the framework's elements

the scheduler starts making decisions based on the shiftability value. For each event, if $shift = 0$ then, it is removed from immediately wait buffer iw and is located in immediately start buffer is . Once the corresponding EMS receives the response, it allows the smart appliance to start working. In contrast, if $shift = 1$, the scheduler checks the interruptibility value $intr$. If $intr = 0$, then, the scheduler investigates if the corresponding smart appliance has been allowed to operate before, i.e., there is an event in decided to operate buffer do which confirms that the corresponding smart appliance has been allowed to operate in the previous time interval. If so, the event is moved to the same buffer. Otherwise, it is checked whether dispatching the operation of the corresponding smart appliance exceeds the provided flexibility ofl . In that case, it is transferred to decided to operate buffer do . Finally, if $intr = 1$, the preceding procedure is executed again.

We define two different types of flexibility ofl named deadline and temperature. Deadline flexibility is an additional time for the required time period of the main operating cycle of physically-controllable smart appliances [6]. Offering this flexibility, corresponding smart appliances can be shifted and/or interrupted until reaching the adjusted deadline flexibility. Each customer is aware of the total period time that each appliance needs to complete its work with respect to its load profile lp . For instance, one may provide two hours flexibility of his/her electric vehicle to the DRS, when he/she can wait at most two additional hours to receive the charged electric vehicle. More precisely, this flexibility is applicable to both start and finish times of appliance operation, since the DRS can shift the starting time, however, it should finish the appliance operation cycle at maximum the provided flexibility. This is a constraint of smart appliances ζ named Appliance Full Operation (AFO). Here, the DRS should consider the difference time between the total number of remaining time intervals and sum of the desired reception time and provided deadline flexibility of each appliance before shifting it to another time interval. Moreover, temperature flexibility is a feature of thermostatically-controllable smart appliances, e.g., HVAC. It is responsible for providing thermal comfort and appropriate indoor air quality with a thermostat which operates in an “on-off” mode and simply runs at its rated power when it is turned on [16]. Customers are able to provide their interested set temperature (= ost) and temperature flexibility ofl in each time interval.

While executing the aforementioned procedure, Electricity Consumption Threshold (ECT) is also updated based on the electricity consumption of smart appliances. ECT is a set of grid constraints which are imposed by DSOs to prevent the electrical grid from any unforeseen circumstances, e.g., electricity blackouts. This set is composed of *feder* and *household* thresholds. For the former, let us assume the scheduler is scheduling events of day \mathbb{D}_η . Obviously, it knows (or at least can predict) the aggregated load consumption of customers in day $\mathbb{D}_{\eta-1}$. Therefore, it

Algorithm 3.1: Load scheduling

Input : Events.
Output: Schedule of events.

- 1 Immediately forward the incoming events to buffer iw
- 2 **while** $iw \neq \{\}$ **do**
- 3 **for** $i = 1$ to $|iw|$ **do**
- 4 **if** $ev_i.shift = 0$ **then**
- 5 $is \leftarrow is \cup ev_i$
- 6 $ect \leftarrow ect - ev_i.lp$
- 7 **else**
- 8 **if** $ev_i.intr = 0$ **then**
- 9 **if** $\exists ev_j \in do \rightarrow sa_j = sa_i$ **then**
- 10 $do \leftarrow do \cup ev_i$
- 11 $ect \leftarrow ect - ev_i.lp$
- 12 **else**
- 13 **if** $ev_i.est + ev_i.ept > ev_i.ofl$ **then**
- 14 $do \leftarrow do \cup ev_i$
- 15 $ect \leftarrow ect - ev_i.lp$
- 16 **end**
- 17 **end**
- 18 **else**
- 19 **if** $ev_i.est + ev_i.ept > ev_i.ofl$ **then**
- 20 $do \leftarrow do \cup ev_i$
- 21 $ect \leftarrow ect - ev_i.lp$
- 22 **end**
- 23 **end**
- 24 **end**
- 25 **end**
- 26 **if** $ect > 0$ **then**
- 27 **if** $\sum_{i=1}^{|iw|} (ev_i.ec) > ect$ **then**
- 28 $ev_{\{1,2,\dots\}} \leftarrow$ Run the event selection procedure
- 29 $do \leftarrow do \cup ev_{\{1,2,\dots\}}$
- 30 $ect \leftarrow ect - \sum_{m=1}^{|ev_{\{1,2,\dots\}}|} ev_m.ec$
- 31 $dw \leftarrow dw \cup iw \setminus ev_{\{1,2,\dots\}}$
- 32 **else**
- 33 $do \leftarrow do \cup iw$
- 34 $ect \leftarrow ect - \sum_{i=1}^{|iw|} (ev_i.ec)$
- 35 **end**
- 36 **else**
- 37 $dw \leftarrow dw \cup iw$
- 38 $ect \leftarrow ect - \sum_{i=1}^{|iw|} ev_i.ec$
- 39 **end**
- 40 **end**

is possible to calculate the *aggregated peak consumption* and *peak consumption of each customer* in that day. As a result, Equations (3.6) and (3.7) formulates feeders and households thresholds, respectively.

$$\begin{aligned}
 ect_{F_{1:\kappa}}^H(\Delta \mathbb{T}) &= \max_{c \in \mathbb{C}} \sum_{sa \in SA} lp & ect_{c_{1:h}}^H(\Delta \mathbb{T}) &= \max_{sa \in SA} lp \\
 ect_{F_{1:\kappa}}^S(\Delta \mathbb{T}) &= \left(\max_{c \in \mathbb{C}} \sum_{sa \in SA} lp \right) \times \mathbb{K}_F(\Delta \mathbb{T}) & ect_{c_{1:h}}^S(\Delta \mathbb{T}) &= \left(\max_{sa \in SA} lp \right) \times \mathbb{K}_c(\Delta \mathbb{T})
 \end{aligned}
 \tag{3.6} \tag{3.7}$$

Here, $ect_{F_{1:\kappa}}^H(\Delta \mathbb{T})$ and $ect_{F_{1:\kappa}}^S(\Delta \mathbb{T})$ are equal to hard and soft ECTs of κ feeders $F_{1:\kappa}$ in each time interval $\Delta \mathbb{T}$, respectively. Notation $\mathbb{K}_F(\Delta \mathbb{T})$ corresponds to the time-dependent feeder factor which can reflect e.g., the variations of market prices or a constant ratio. Correspondingly, $ect_{c_{1:h}}^H(\Delta \mathbb{T})$ and $ect_{c_{1:h}}^S(\Delta \mathbb{T})$ are hard and soft ECTs of h households $c_{1:h}$ in each time interval $\Delta \mathbb{T}$, respectively. Notation $\mathbb{K}_c(\Delta \mathbb{T})$ means the same as $\mathbb{K}_F(\Delta \mathbb{T})$. This PhD project assumes that the thresholds for day \mathbb{D}_η are calculated based on aggregated load consumptions in day $\mathbb{D}_{\eta-1}$ and relevant time-dependent factors $\mathbb{K}_F(\Delta \mathbb{T})$ and $\mathbb{K}_c(\Delta \mathbb{T})$ in day \mathbb{D}_η . Afterwards, the scheduler checks if there is any ect_F^S remained or not (line 26 in Algorithm 3.1). If so, it commence investigating whether the aggregated load consumption of remaining events are greater that ect_F^S . Otherwise, all remaining events are moved to decided to wait buffer dw . If $\sum_{i=1}^{|iw|} (ev_i.ec) > ect$, the scheduler calls an *event selection procedure*. Otherwise, it decides to move all remaining events to decided to operate buffer *do*.

3.2.1 Event Selection Problem

The event selection procedure runs an optimization procedure, to decide which events should be selected, taking customers' and DSO's objectives into account combined with the remaining thresholds as its constraints. This results in an NP-complete problem since it is reducible to the *Knapsack Problem* [17, 6]. By reduction, we mean if an algorithm for solving the knapsack problem efficiently (if it exists) could also be used as a subroutine to solve the event selection problem efficiently. The Knapsack problem is a traditional problem of Computer Science in combinatorial optimization literature. Given a finite number of items, the Knapsack attempts to pack the items to get the maximum total value considering its specific capacity. Each indivisible item has a weight and a value. Hence, it is an NP-Complete problem since the time complexity of solving it in a brute-force manner, i.e., calculating all feasible subsets in order to find the optimal one, is intractable. Referring to the event selection problem, the objective set can be a subset of customers' and DSO's objectives while the knapsack capacity equals to the remaining ECTs. On the one side, from the customers' perspective, the scheduler is able to postpone the operation of smart appliances to time periods with cheap electricity prices, i.e., Electricity Cost Minimization (ECM). Afterwards, the scheduler is responsible for operating the smart appliances as close to the preferred operating start times as possible, i.e., Comfort Level Maximization (CLM). On the other side, from the DSO's perspective, the scheduler aims at flattening the aggregated consumption, i.e., Peak Demand Flattening (PDF). In addition, it is interested in reducing the amount of CO₂ emission, i.e., CO₂ Emission Minimization (CEM). Indeed, these objectives come into conflict with each other. As a result, we confront a multi-objective constrained load scheduling problem which is also NP-complete [18]. However, it is possible for the scheduler to choose a single-objective or a multi-objective optimization technique based on the number of objectives required to be optimized in each time interval.

3.2.1.1 Single-objective Optimization: Dynamic Programming

Dynamic programming is a method for solving optimization problems. The idea is dividing the problem into sub-problems, solving each sub-problem once, and storing the solutions in a table to reuse them repeatedly. Dynamic programming has some principals as follows:

- Structure: Characterizing the structure of an optimal solution.
 - Decomposing the problem into smaller problems.
 - Finding a relation between the structure of the optimal solution of the original problem and solutions of the smaller problems.
- Optimality: Recursively defining the value of an optimal solution.
 - Expressing the solution of the original problem in terms of optimal solutions for smaller problems.
 - Bottom-up computation: Computing the value of an optimal solution in a bottom-up manner using a table structure.

3.2.1.2 Multi-Objective Optimization: Evolutionary Algorithm

Evolutionary algorithms are among the most well-known meta-heuristic search mechanisms utilized to generate feasible solutions to a multi-objective optimization problem. Being free of the objective search space is their unique feature [19]. Basically, an evolutionary algorithm randomly generates a population including a set of feasible solutions. Then, it executes an exploitation and then, exploration procedures on the population to choose best individuals for the next generation. We choose Non-dominated Sorting Genetic Algorithm-II (NSGA2) due to its fast non-dominated sorting approach and ability to find much better spread and convergence of solutions near the true Pareto-front [20], as Figure 3.2 shows its flowchart. The scheduler aims at finding a Pareto-front in the objective space including a set of *non-dominant* Pareto-optimal solutions as diverse as possible (see Figure 3.3). In this space, solution one *dominates* solution two, if it is better than solution two in some objectives and perhaps equal in others. A Pareto-optimal solution does not improve for one objective unless it satisfies other objective(s). In the load scheduling problem, each solution is represented as an admissible subset of the remaining events in each time interval. To be more precise, a Pareto-optimal solution specifies “which events” are selected to move into decided to operate buffer *do*.

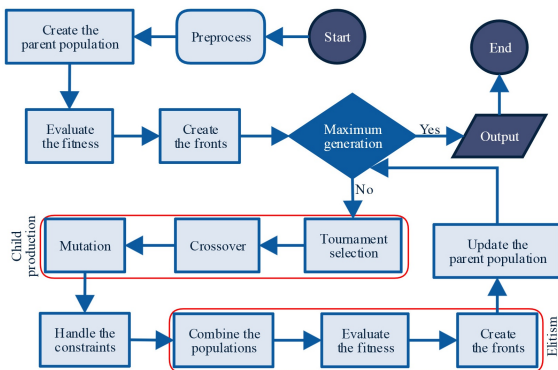


Figure 3.2: Flowchart of NSGA2

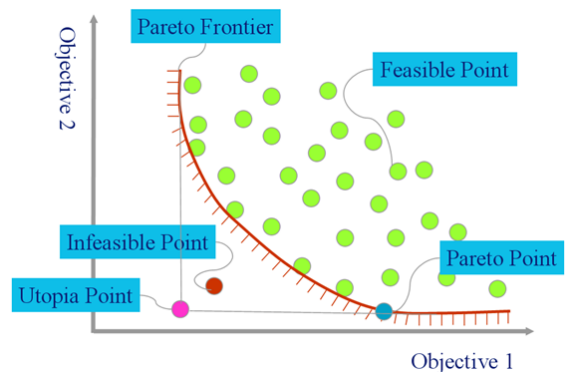


Figure 3.3: A Pareto-front

Simulation Analysis and Future Plans

This chapter analyzes the simulations done in the first part of the PhD study and describes the activities planned for completion of the project.

4.1 Simulation Analysis

This section first describes the simulation setup demonstrating how to instantiate different objects of the formal framework with respect to the load scheduling's requirements. Subsequently, the simulation results and corresponding analyses will be clarified precisely.

4.1.1 Simulation Setup

The formal framework and proposed algorithms have been implemented with MATLAB R2015b on a personal computer with an Intel Core i7-2.0GHz CPU and 6GB of memory. Fig. 4.1 pictures the conceptual view of the framework wrapped as a load scheduling problem. We assume $h = 100$ connected to one feeder, where each has an *ems* and three smart appliances, i.e., one washing machine $sa_1 = \text{WM}$, one tumble dryer $sa_2 = \text{TD}$, and one dish washer $sa_3 = \text{DW}$, operating one time per day. To capture load profiles of smart appliances used in the simulation, we design a co-simulation interface to the BehavSim software to specify a random scenario for each customer in one minute time resolution [21]. It has been particularly developed as SEMIAH consumption simulation tool providing statistical consumption of households, presenting a user interface to define behavior of householders, and generating data consumption of households. As a final note, in accordance with Equation (3.4), event pooling times *ept* of WM, TD, and DW are two, three, and four minutes, respectively. The event sent time *est* of each smart appliance equals to its operating start time *ost*. Table 4.1 presents the preliminary information needed to perform the load scheduling algorithm. Here, AFO and CLM stand for Appliance Full Operation and Comfort Level Maximization, respectively. Afterwards, the DSO is interested in flattening the peak demand periods (PDF). To this end, hard feeder threshold $ect_{F_{1:\kappa}}^H$ prevents the grid from getting overload while soft feeder threshold $ect_{F_{1:\kappa}}^S$ is a target for the DSO to keep the aggregated load consumption below it. The set of electricity prices *ep* is captured from the Nord Pool Spot (NPS) [22]. To sum up, we utilize a Multi-Objective Evolutionary Algorithm (MOEA) as the scheduling approach, where it gets inspirations from NSGA2 [20].

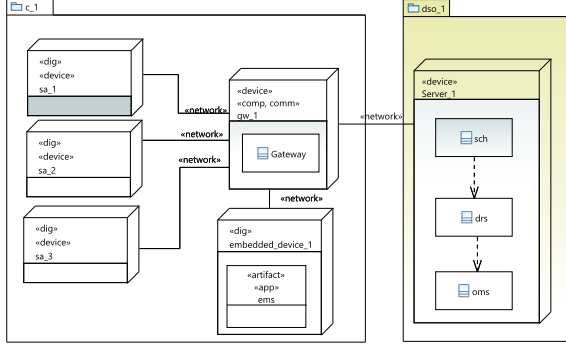


Figure 4.1: Conceptual view of the framework as a load scheduling problem

Table 4.1: Setting of framework elements

$c1:100$				
sa		sf		ems
		$shift$	$intr$	obj_C
$sa_1 = WM$		1	1	
$sa_2 = TD$		1	1	AFO
$sa_3 = DW$		1	1	CLM
oms				
obj_O	ct_{dso}		ep	λ
PDF	$ect_{F_1}^H$	$ect_{F_1}^S$	NPS	MOEA

4.1.2 Simulation Analysis

Figure 4.2 pictures the aggregated load consumption of 100 customers before scheduling. According to the load profiles generated with BehavSim, events arrive in the period from 07:00 to 24:00, in which ECTs are approached. Maximum Electricity Consumption (MEC) occurs at 09:43 with 62.352 watts.

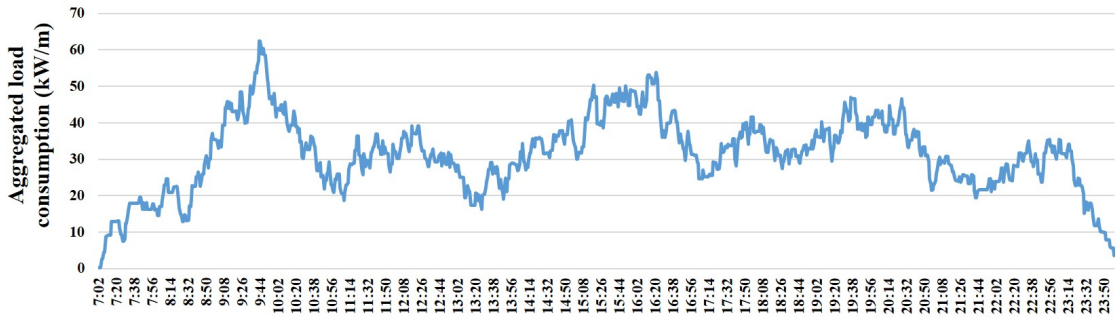


Figure 4.2: Aggregated load consumption of 100 customers in one minute time resolution

For the flexibility, customers can provide *maximum possible* (MaxFlex), *minimum possible* (MinFlex), and *random possible* (RandFlex) values as three different flexibility scenarios. For simplicity, we set $ect_{F_1}^H = MEC$ as households' thresholds in each time interval. According to DSO's objective, i.e., PDF, we enable the shifting and interruption by assigning $\mathbb{K}_F = (10 : 90)\%$, where $ect_{F_1}^S = \mathbb{K}_F \times ect_{F_1}^H$. We change \mathbb{K}_F to analyze its influence on:

- Peak Demand Reduction (PDR): $\frac{MEC - MEC_{sch}}{MEC} \times 100$
 – MEC_{sch} means the Maximum Electricity Consumption after scheduling the events.
- Peak-to-Average Ratio (PAR): $\frac{MEC_{sch}}{\left(\sum_c \sum_{sa} lp\right) / 1440}$
- Appliance Serving Delay (ASD): $\frac{\sum_c \sum_{sa} (OFT_{sch} - OFT_c)}{100 \times 3}$
 – OFT_{sch} is the time at which the scheduler finishes operating the smart appliance. OFT_c is the time at which the smart appliance finishes operating without any scheduling.
- Customer Flexibility Usage (CFU): $\frac{\sum_c \sum_{sa} ((OFT_{sch} - OFT_c) / (fl - OFT_c))}{c \times sa} \times 100$

- Customer Electricity Cost (CEC)
- Scheduling Computation Time (SCT)

Figure 4.3 shows PDR percentage in three different flexibility scenarios MaxFlex, MinFlex, and RandFlex. Starting from 10%, since the assigned $ect_{F_1}^S$ is very low comparing with MEC, the scheduler is unable to operate more smart appliances. In MaxFlex, since all customers provide the maximum possible flexibility, i.e., $fl = 1440$, most of smart appliances are postponed to be allowed in future. This causes a big rebound peak occurred at the end of the day. Obviously, the electrical grid cannot tolerate this situation. In MinFlex, since fl is just one minute after the finishing time of the smart appliances, the scheduler has to allow all smart appliances to operate at the time customers are interested. In RandFlex, we still confront the rebound peak situation, however, relatively in lower percentage comparing with MaxFlex situation. This problem continues to be alleviated while we increase $ect_{F_1}^S$. When $ect_{F_1}^S = 60\% \times ect_{F_1}^H$, we observe a significant PDR in both MaxFlex and RandFlex scenarios. However, while we again continue to increase $ect_{F_1}^S$, PDR starts decreasing. The reason is that when the scheduler is able to operate more appliances in each time interval, then, there is no need to use most of the flexibilities. The interesting point is that when $ect_{F_1}^S = 80\% \times ect_{F_1}^H$ and $ect_{F_1}^S = 90\% \times ect_{F_1}^H$, the scheduler is able to use a small portion of flexibilities. This is due to the fact that events density in peak times are low. Indeed, applying $ect_{F_1}^S = 100\% \times ect_{F_1}^H$ gives the scheduler the permission to allow all smart appliances to operate at the time they have been set. Table 4.2 lists corresponding PAR values. Finally, Figure 4.4 shows the aggregated load consumption before and after scheduling with different flexibility scenarios. In the same context, Figure 4.5 pictures the load consumption of a random household before and after scheduling.

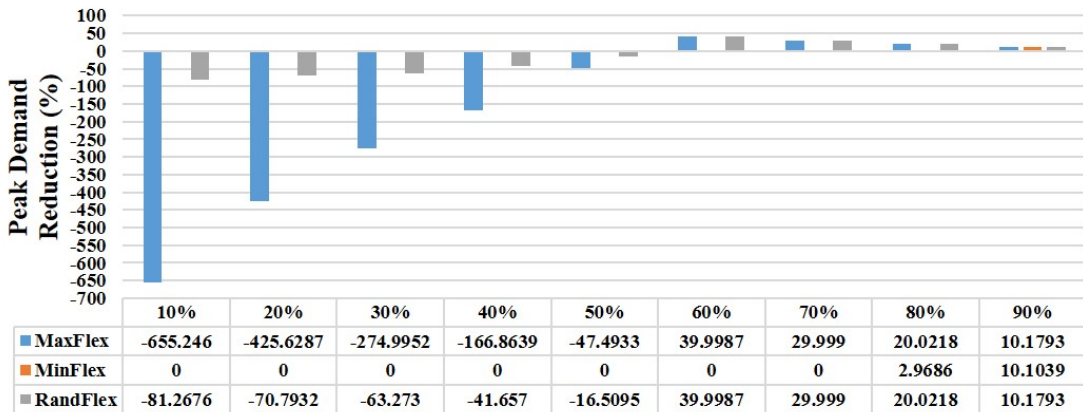


Figure 4.3: Peak Demand Reduction percentage in three different flexibility scenarios

Table 4.2: Peak-to-Average Ratio in three different flexibility scenarios

	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
MaxFlex	21.17	14.73	10.51	7.48	4.13	1.68	1.96	2.24	2.51	2.80
RandFlex	5.08	4.78	4.57	3.97	3.26	1.68	1.96	2.24	2.51	2.80
MinFlex	2.80	2.80	2.80	2.80	2.80	2.80	2.80	2.72	2.52	2.80

Figure 4.6 pictures CFU percentage while Figure 4.7 shows average ASD in three different flexibility scenarios MaxFlex, MinFlex, and RandFlex. When $ect_{F_1}^S = 10\% \times ect_{F_1}^H$, the scheduler gets benefit of flexibilities as much as possible. This reflects on ASD accordingly. In MaxFlex scenario, the scheduler uses 46.73% of flexibilities which causes customers to wait averagely 543

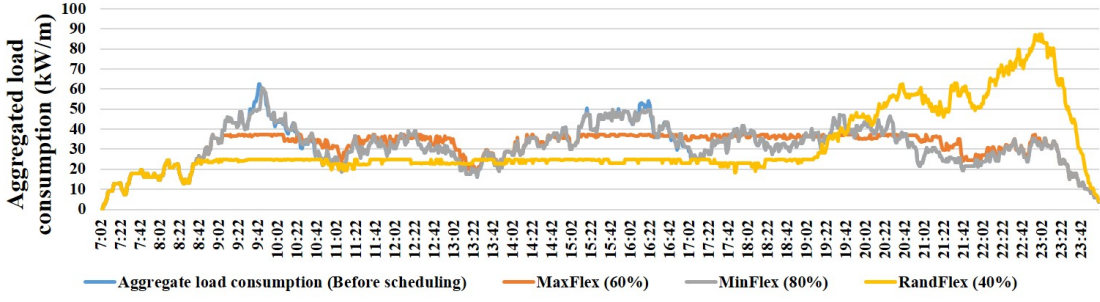


Figure 4.4: Aggregated load consumption before and after scheduling

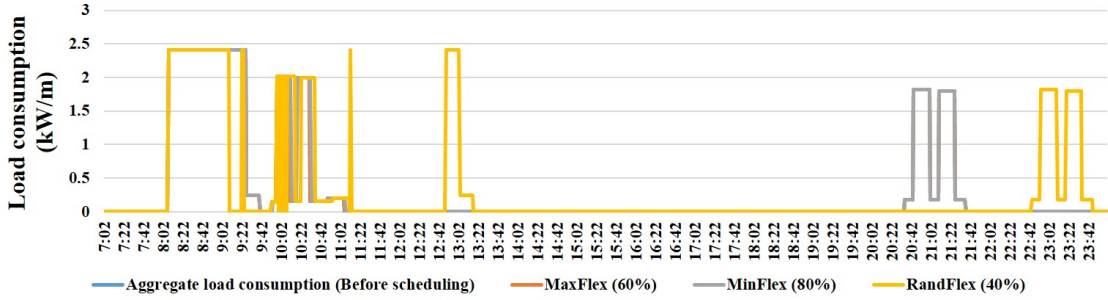


Figure 4.5: load consumption of a random household before and after scheduling

minutes to receive their smart appliances in finished status. Although the scheduler uses 59.3% of flexibilities in MinFlex Scenario, however, since customers have provided minimum possible flexibility, the average ASD is very low, i.e., 1.46 minutes. Both CFU and ASD decrease while ECT increases. Here, there is a trade-off between PDR and ASD. The DSO is inherently interested in flattening the aggregated peak consumption while customers concern to operate their smart appliance as they desire. This proves the DR potentiality for the aggregator to trade the flexibility in the electricity market. According to Figure 4.3, when $ect_{F_1}^S = 60\% \times ect_{F_1}^H$, the DSO is able to reduce the peak by almost 40%. In this situation, customers should just wait averagely for 14 minutes. It should be noted that when we speak about maximum possible flexibility, we consider one-day scheduling scheme. It means that all smart appliances are operated until 24:00. This assumption opens another challenge: “How can we run a consecutive load scheduling?” Here, the main concern is how to intelligently assign various values to $ect_{F_1}^S$ in consecutive days to reach an almost aggregated load consumption.

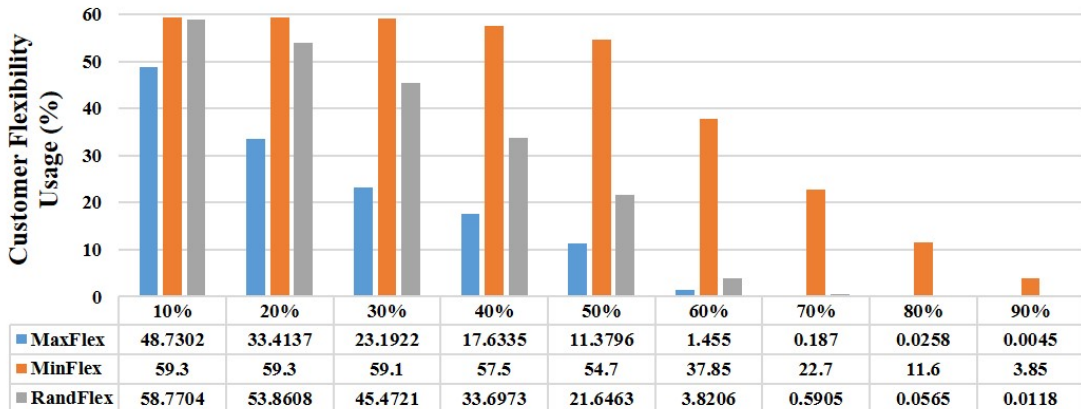


Figure 4.6: Customer Flexibility Usage percentage in three different flexibility scenarios

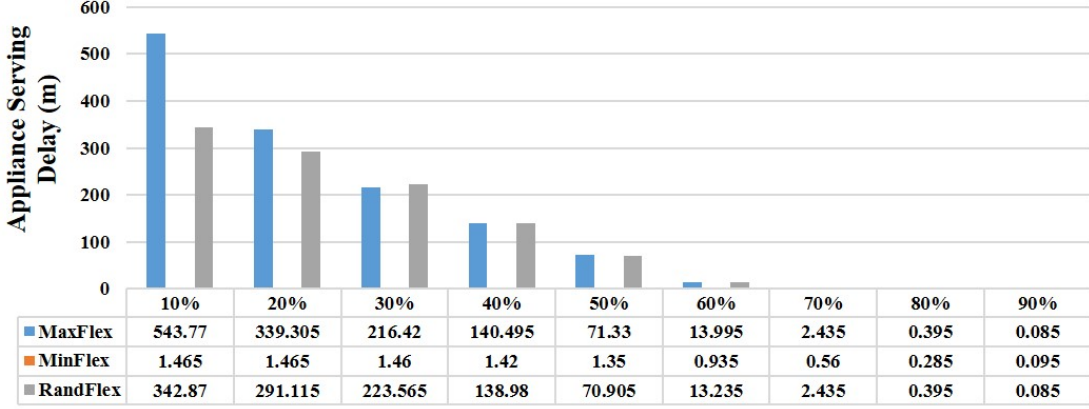


Figure 4.7: Appliance Serving Delay percentage in three different flexibility scenarios

Figure 4.8 demonstrates the daily average Customer Electricity Cost (CEC) in three different flexibility scenarios when we fluctuate the $ect_{F_1}^S$. For MaxFlex scenario, since electricity prices are relatively low during off-peak period and customers provide maximum possible flexibility, the scheduler is able to postpone the operating time of majority of smart appliances to the end of day. This results in decreasing the daily average CEC. Therefore, while we increase the $ect_{F_1}^S$, daily average CEC also increases. In contrast, in MinFlex scenario, since customers are not willing to offer any considerable flexibility, their daily average CEC does not differ when $ect_{F_1}^S$ is changed. The reason is the scheduler's inability in scheduling the smart appliances. The behavior of daily average CEC in RandFlex scenario is similar to the situation in MaxFlex scenario. Here, because of the randomness, values for the daily average CEC are relatively high.

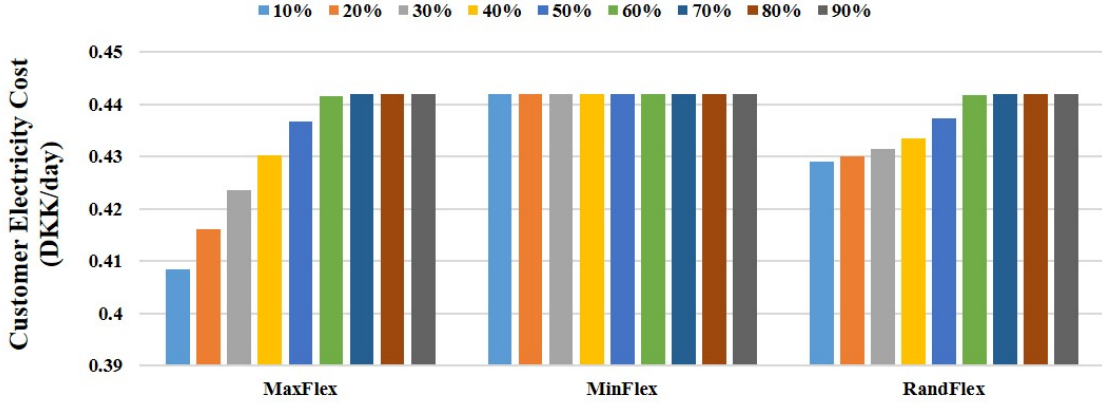


Figure 4.8: Daily average Customer Electricity Cost in three different flexibility scenarios

As a final analysis, Figure 4.9 displays the average SCT in each time interval of three different flexibility scenarios. For MaxFlex scenario, while we increase the $ect_{F_1}^S$ from 10% to 50% of the $ect_{F_1}^H$, the average SCT also increases. This is due to the fact the aggregated load consumption of majority of smart appliances is greater than $ect_{F_1}^S$. Therefore, higher average SCT comes from high number of calls to *event selection procedure* and the number of remaining events inside each call to *event selection procedure*. Nevertheless, while $ect_{F_1}^S \geq 60\% ect_{F_1}^H$, the average SCT decreases. The main reason is decreasing the number of calls to the *event selection procedure* and thus, having a few number of remaining events inside each call. For MinFlex, the situation is completely different since there is no need to call the *event selection procedure* because almost all of smart appliances should be operated at the time the customers are interested. Therefore, the average SCT has a quite constant slope. Finally, when customers provide random flexibilities, the situation is similar

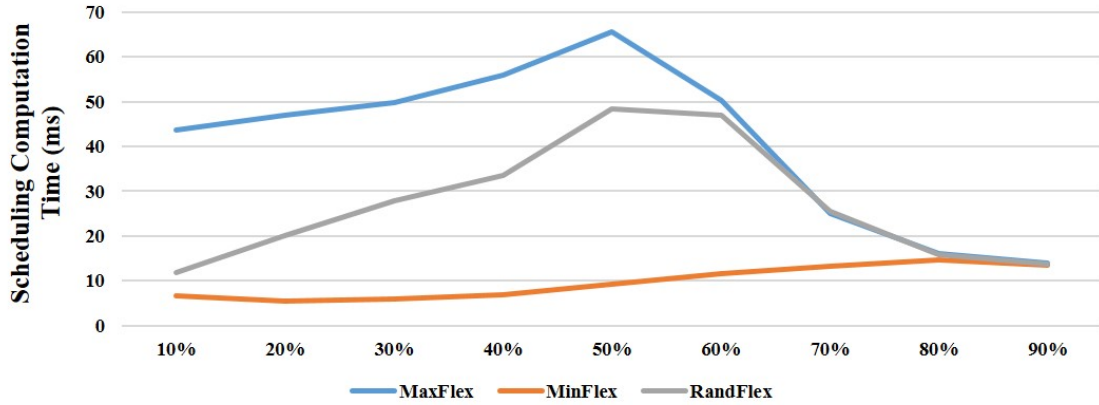


Figure 4.9: Average Scheduling Computation Time in three different flexibility scenarios

to MaxFlex scenario, however, with a different slope. It also follows the same reason as emphasized for MaxFlex scenario.

4.2 Future Plans

The future plans for completion of the PhD project are categorized as follows. The publications planned to submit under the future plans are displayed in Appendix A.

4.2.1 Intelligent Grid Constraints

Inherently, DSOs are interested in continuously maintaining the electrical grid to prevent it from unforeseen hazardous circumstances. Therefore, it is essential to propose *Intelligent Grid Constraints* in both feeder and household levels. Here, being intelligent is defined as a procedure which periodically analyzes aggregated load consumptions and behaviors of customers before, during, and after scheduling. These analyses lead to first learn, then, fluctuate the ECTs in both feeder and households levels. Finally, it has been planned to deploy these intelligent grid constraints in load scheduling algorithms to interact automatically over time.

4.2.2 Load Scheduling Algorithms in the Pilot-Testing Households

This PhD project intends to verify and validate the load scheduling algorithms in two phases. First phase aims to test the whole SEMIAH DR framework, including proposed load scheduling algorithms, in a lab before installing it in the pilot-testing households. The goal of this phase is twofold: on one hand, it validates the technical feasibility of controlling flexible appliances in households. Technical validation will include errors and delays assessment, system failures records, etc. On the other hand, it studies customers' behavior, to serve as an anchorage for a large scale simulation. The second phase, studies 200 customers' interaction with the system which will allow the PhD project to evaluate their acceptance rate toward such a system. Its results will lead to announcing a simulator used to evaluate the energy shifting potential of 200,000 households.

4.2.3 Flexibility Trading Potential in the Electricity Market

The research done has led to a define of flexibility concept used in load scheduling algorithms. Beside having deadline and temperature flexibilities described before, there is an interested to define a new type named *curtailability*. This will potentially decreases the *consumption time* or

the *consumption percentage*, for instance brightness setting. Apparently, it cannot have so much influence on DR programs, however, from an aggregator point of view, it is not negligible, especially when number of DR participant increases. Now, the main challenge appears when a customer wants to offer these flexibilities to the intra-day market. If the offer gets accepted, is there any guarantee (how much percentage) to rely on and use them appropriately? Therefore, this PhD project tries to interpret the aggregated flexibility to quality of DR to be able to make it tradable. This requires an additional research on the sensitivity of the scheduling on the flexibility provided by customers.

4.2.4 Decentralizing Load Scheduling Algorithms

The concept of a decentralized load scheduling algorithm presents a powerful counterpoint to the more conventional centralized one. Decentralization provides a number of significant advantages over closed systems, such as robustness, adaptability, flexibility, innovation, and distributed intelligence. The key to this compelling architecture is the impressive ability of a decentralized system to react or grow in response to the challenge in increasing the number DR participants. Multi-agent systems use distributed agents to either model or solve such challenging problem. An agent is an entity which matches some real-world object, e.g., a customer. The agents can both act independently or interact with each other in their neighborhood to form some coalitions. This PhD project will strive to decentralize the load scheduling algorithms in such a way that each customer be responsible for managing and controlling the corresponding smart appliances. This can result in employing various game theory concepts. The initial intention is to carry out this research in the stay abroad, aiming to find a research institution with relevant expertise in this research area.

4.2.5 Time planning

In the Gantt diagram displayed below the future plans sketched above are framed together with other relevant milestones and tasks like staying abroad and dissertation writing. Staying abroad is planned to last around five months starting on Fall 2016.

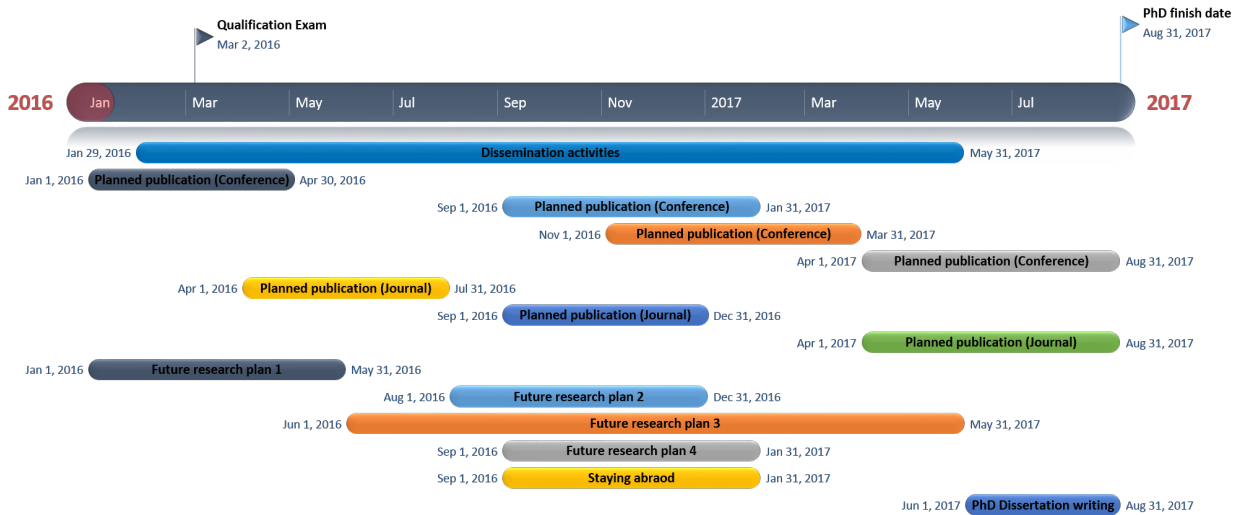


Figure 4.10: Gantt diagram of the future plans

Bibliography

- [1] H. Farhangi, “The Path of the Smart Grid,” *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, 2010.
- [2] Universal Smart Energy Framework (USEF). [Online]. Available: <http://www.usef.info>
- [3] A. Soares, Á. Gomes, and C. H. Antunes, “Categorization of Residential Electricity Consumption as a Basis for the Assessment of the Impacts of Demand Response Actions,” *Renewable and Sustainable Energy Reviews*, vol. 30, pp. 490–503, 2014.
- [4] J. Momoh, *Smart Grid: Fundamentals of Design and Analysis*. John Wiley & Sons, 2012, vol. 63.
- [5] R. H. Jacobsen, D. Gabioud, G. Basso, P.-J. Alet, A. G. Azar, and E. S. M. Ebeid, “SEMIAH: An Aggregator Framework for European Demand Response Programs,” in *IEEE Euromicro Conference on Digital System Design (DSD)*, 2015, pp. 470–477.
- [6] A. Ghasem Azar, R. Hylsberg Jacobsen, and Q. Zhang, “Aggregated Load Scheduling for Residential Multi-Class Appliances: Peak Demand Reduction,” in *Proceedings of the 12th International IEEE Conference on the European Energy Market-EEM, 2015*.
- [7] GridWise® Architecture Council. [Online]. Available: <http://www.gridwiseac.org/>
- [8] Bruinenberg, J. and Colton, L. and et al., “Smart Grid Coordination Group Technical Report Reference Architecture for the Smart Grid Version 1.0 (draft),” Tech. Rep., 2012.
- [9] V. Alagar and K. Periyasamy, “Extended Finite State Machine,” in *Specification of Software Systems*, ser. Texts in Computer Science. Springer London, 2011, pp. 105–128.
- [10] N. Framework, “Roadmap for Smart Grid Interoperability Standards,” *NIST special publication*, vol. 1108, 2010.
- [11] K. C. Sou, J. Weimer, H. Sandberg, and K. H. Johansson, “Scheduling Smart Home Appliances Using Mixed Integer Linear Programming,” in *Proceedings of 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011, pp. 5144–5149.
- [12] Smart Energy Profile 2.0 Application Protocol Standard, “ZigBee Alliance, Tech. Rep. Document 13-0200-00, 2013.”
- [13] E. Sameul Malki Ebeid, S. Rotger Griful, S. Aagaard Mikkelsen, and R. Hylsberg Jacobsen, “A Methodology to Evaluate Demand Response Communication Protocols for the Smart Grid,” in *Proceedings of IEEE International Conference on Communications (ICC) 2015*.
- [14] “IEC 61970: Energy Management System Application Program Interface (EMS-API) - Part 301: Common Information Model (CIM) base, Third Edition,” Tech. Rep., 2009.
- [15] K. Zhou, J. Pan, and L. Cai, “Optimal Combined Heat and Power System Scheduling in Smart Grid,” in *Proceedings of IEEE INFOCOM*, 2014, pp. 2831–2839.

Bibliography

- [16] S. Shao, M. Pipattanasomporn, and S. Rahman, “Development of Physical-Based Demand Response-Enabled Residential Load Models,” *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 607–614, 2013.
- [17] H. Kellerer, U. Pferschy, and D. Pisinger, *Introduction to NP-Completeness of Knapsack Problems*. Springer, 2004.
- [18] T. Lust and J. Teghem, “The Multiobjective Multidimensional Knapsack Problem: A Survey and a New Approach,” *International Transactions in Operational Research*, vol. 19, no. 4, pp. 495–520, 2012.
- [19] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001, vol. 16.
- [20] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II,” *Springer Lecture notes in Computer Science*, vol. 1917, pp. 849–858, 2000.
- [21] G. Basso, P. Ferrez, D. Gabioud, and P. Roduit, “An Extensible Simulator for Dynamic Control of Residential Area: Case Study on Heating Control,” in *IEEE Euromicro Conference on Digital System Design (DSD)*, 2015, pp. 486–493.
- [22] Nord Pool Spot. [Online]. Available: <http://www.nordpoolspot.com>



Publications

This appendix outlines the published, submitted, and planned contributions of the PhD project in terms of scientific publications and project deliverables. For each planned publication the expected time of submission and the target publisher (e.g., conference or journal) are shown.

A.1 Scientific Publications

A.1.1 Published

A.1.1.1 Conferences

- Rune Hylsberg Jacobsen, Dominique Gabioud, Gillian Basso, Pierre-Jean Alet, **Armin Ghasem Azar**, Emad Samuel Malki Ebeid, *SEMIAH: An Aggregator Framework for European Demand Response Programs*, Euromicro Conference on Digital Systems Design (DSD), Madeira, Portugal, August 2015.
- Rune Hylsberg Jacobsen, **Armin Ghasem Azar**, Qi Zhang, Emad Samuel Malki Ebeid, *Home Appliance Load Scheduling with SEMIAH*, International Conference on Smart Systems, Devices, and Technologies (SMART), Brussels, Belgium, June 2015, *The Best Paper Award*.
- **Armin Ghasem Azar**, Rune Hylsberg Jacobsen, Qi Zhang, *Aggregated Load Scheduling for Residential Multi-Class Appliances: Peak Demand Reduction*, International Conference on European Energy Market (EEM), Lisbon, Portugal, May 2015.

A.1.2 Submitted

A.1.2.1 Journals

- **Armin Ghasem Azar**, Rune Hylsberg Jacobsen, Qi Zhang, *Multi-Objective Residential Load Scheduling Optimization*, International Journal on Advances in Intelligent Systems. [Submitted on March 2, 2016].

A.1.2.2 Conferences

- **Armin Ghasem Azar**, Emad Samuel Malki Ebeid, Rune Hylsberg Jacobsen, *A Formal Framework for Modeling Smart Grid Applications: Demand Response Case Study*, ACM

International Conference on Future Energy Systems (e-Energy), Waterloo, Canada, June 2016. [Submitted on February 12, 2016].

A.1.3 Planned

A.1.3.1 Journals

- *A Robust Load Scheduling Framework for Flexible Aggregation of Multi-Class Smart Appliances*, IEEE Transactions on Smart Grid.
- *A High-Level Co-Simulator for Smart House Appliance Load Scheduling in the Smart Grid*, Elsevier Applied Energy.
- *Demand Side Management and Load Scheduling in the Smart Grid*, IEEE Transactions on Smart Grid.

A.1.3.2 Conferences

- *Intelligent Load Scheduling Policy for the Scalable Aggregation of Households*, IEEE ISGT USA, 2016.
- *Autonomous Learning-based Smart Appliance Load Scheduling in the Smart Grid*, EnergyCon 2017.
- *Stochastic scheduling of defferable and interruptible appliances in the smart grid*, IEEE Smart-GridComm 2017.
- *A distributed agent-based load scheduling flexibility in the demand side management*, ACM e-Energy 2017.

A.2 SEMIAH Project Deliverables

A.2.1 Published

- D5.1 - Algorithms for Demand Response and Load Control, March 2015, 30 pages, [*Main editor*].

A.2.2 Submitted

- D4.3 - Demand Response Prototype, March 2016.
- D5.4 - Back-End System Release, March 2016.



Courses

This section contains a list of all the courses taken during the first 1.5 years of the PhD project. The sum of all the credits leads to 30 ECTS, thus, fulfilling the requirements from the Graduate School of Science and Technology (GSST) at AU.

B.1 Scientific Courses

Groningen Energy Summer School <i>University of Groningen, PhD Course, 5 ECTS</i>	Q1 2015
Winter School of the Munich School of Engineering <i>Technical University of Munich, PhD Course, 3 ECTS</i>	Q3 2014
Wireless IP and Internet of Things <i>Aarhus University, MSc Course, 5 ECTS</i>	Q2 2014
Communications for MicroGrids <i>Aalborg University, PhD Course, 2 ECTS</i>	Q1 2014
Energy Management Systems and Optimization in MicroGrids <i>Aalborg University, PhD Course, 3 ECTS</i>	Q1 2014

B.2 Transferable Skill Courses

Science Teaching: Introduction to Science Teaching <i>Aarhus University, PhD Course, 5 ECTS</i>	Q1 2016
Scientific Writing and Communication <i>Aarhus University, PhD Course, 4 ECTS</i>	Q1 2015
Academic English for non-Danish PhD Students <i>Aarhus University, PhD Course, 3 ECTS</i>	Q3 2014



Dissemination Activities

It is expected that the PhD student performs dissemination activities according to 280h/year. A total amount of 560 hours of dissemination activities have already been categorized into performed and planned parts.

C.1 Performed

In the first part of the PhD project 536 hours (63%) of obligatory dissemination activities have been performed as follows:

- | | |
|------------------------------|-----------|
| • Teaching activities | 190 hours |
| • Writing deliverables | 290 hours |
| • Project presentations | 46 hours |
| • Extracurricular activities | 10 hours |

C.2 Planned

The following lists dissemination activities planned for the remaining period of the PhD project.

- | | |
|----------------------------|-----------|
| • Supervision activities | 50 hours |
| • Staying abroad | 114 hours |
| • Writing PhD dissertation | 140 hours |

Armin Ghasem Azar, Demand Response Driven Load Scheduling in Formal Smart Grid Framework, 2016