AARHUS
UNIVERSITY
SCIENCE AND TECHNOLOGY

SALSA: A Formal Hierarchical Optimization Framework for Smart Grid

Department of Engineering

Armin Ghasem Azar

PhD Dissertation

SALSA: A Formal Hierarchical Optimization Framework for Smart Grid

PhD Dissertation

Armin Ghasem Azar

August, 2017

AARHUS
UNIVERSITY
DEPARTMENT OF ENGINEERING

Dissertation submitted:	August 14, 2017
PhD Supervisor:	Associate Prof. Rune Hylsberg Jacobsen Department of Engineering, Aarhus University
PhD Committee:	Prof. Peter Palensky, Department of Electrical Sustainable Energy, Delft University of Technology Prof. Bo Nørregaard Jørgensen, Center for Energy Informatics, University of Southern Denmark (SDU) Associate Prof. Stefan Hallerstede (chair), Department of Engineering, Aarhus University
PhD Series:	Faculty of Science and Technology, Aarhus University

Published by:
River Publishers
Niels Jernes Vej 10
9220 Aalborg
Denmark
http://riverpublishers.com/

Printed in Denmark by Rosendahls, 2017

Aarhus University Library

ISBN: 978-87-7507-408-2 (online)
DOI: 10.7146/aul.231.166

Abstract

The smart grid, by the integration of advanced control and optimization technologies, provides the traditional grid with an indisputable opportunity to deliver and utilize the electricity more efficiently. Building smart grid applications is a challenging task, which requires a formal modeling, integration, and validation framework for various smart grid domains. The design flow of such applications must adapt to the grid requirements and ensure the security of supply and demand. This dissertation, by proposing a formal framework for customers and operations domains in the smart grid, aims at delivering a smooth way for: i) formalizing their interactions and functionalities, ii) upgrading their components independently, and iii) evaluating their performance quantitatively and qualitatively.

The framework follows an event-driven demand response program taking no historical data and forecasting service into account. A scalable neighborhood of prosumers (inside the customers domain), which are equipped with smart appliances, photovoltaics, and battery energy storage systems, are considered. They individually schedule their appliances and sell/purchase their surplus/demand to/from the grid with the purposes of maximizing their comfort and profit at each instant of time. To orchestrate such trade relations, a bilateral multi-issue negotiation approach between a virtual power plant (on behalf of prosumers) and an aggregator (inside the operations domain) in a non-cooperative environment is employed. The aggregator, with the objectives of maximizing its profit and minimizing the grid purchase, intends to match prosumers' supply with demand. As a result, this framework particularly addresses the challenges of: i) scalable and hierarchical load demand scheduling, and ii) the match between the large penetration of renewable energy sources being produced and consumed. It is comprised of two generic multi-objective mixed integer nonlinear programming models for prosumers and the aggregator. These models support different scheduling mechanisms and electricity consumption threshold policies.

The effectiveness of the framework is evaluated through various case studies based on economic and environmental assessment metrics. An interactive web service for the framework has also been developed and demonstrated.

Resumé

Et smart grid, ved at integrere avancerede styrings- og optimeringsteknologier, giver det traditionelle net en ubestridelig mulighed for at levere og udnytte el mere effektivt. Opbygning af smart grid applikationer er en udfordrende opgave, som kræver et formelt rammeværk for modellering-, integration og validering af forskellige domæner. Designflowet af sådanne applikationer skal tilpasse sig netkravene og sikre forsynings- og efterspørgselssikkerheden. Denne afhandling, ved at foreslå et formelt rammeværk for kunde- og driftdomæner i smart grid, sigter mod at levere en smidig måde at: i) formalisere deres interaktioner og funktionaliteter, ii) opgradere deres komponenter uafhængigt og iii) vurdere deres ydelse kvantitativt og kvalitativt.

Rammen følger et event-drevet demand response-program, hvor der ikke tages hensyn til historiske data og prognoser. Et skalerbart nabolag af (inden for kundedomænet), der er udstyret med intelligente apparater, solceller og batteriladere, betragtes. Disse planlægger individuelt deres apparater og sælger/køber deres overskud/efterspørgsel til/fra nettet med det formål at maksimere deres komfort og fortjeneste hvert øjeblik. For at orkestrere sådanne handelsmuligheder er der brugt en bilateral multilagsforhandlingsstrategi mellem et virtuelt kraftværk (på vegne af prosumere) og en aggregator (inden for driftssdomænet) i et ikke-kooperativt miljø. Aggregatoren, der har til formål at maksimere sin fortjeneste og minimere køb fra nettet, har til hensigt at matche prosumers forsyning med efterspørgslen. Som følge heraf behandler dette rammeværk især udfordringerne ved: i) skalerbar og hierarkisk efterspørgselsplanlægning, og ii) matchet mellem produktion og forbrug fra den store indtrængning af vedvarende energikilder. Den består af to generiske multi-objectiv mixed integer ikke-lineære programmeringsmodeller til prosumere og aggregatoren. Disse modeller understøtter forskellige skeduleringsmekanismer og politikker for en elforbrugstærskel.

Rammeværkets effektivitet evalueres gennem forskellige casestudier baseret på økonomiske og miljømæssige metrikker til vurdering. En interaktiv webtjeneste for rammeværket er ligeledes udviklet og demonstreret.

Contents

Dissertation Details

Dissertation Title:	SALSA: A Formal Hierarchical Optimization Framework for Smart Grid
PhD Student:	Armin Ghasem Azar
Supervisor:	Associate Prof. Rune Hylsberg Jacobsen, Aarhus University

The body of this dissertation consists of the following papers.

[1] Armin Ghasem Azar, Rune Hylsberg Jacobsen, and Qi Zhang, "Aggregated Load Scheduling for Residential Multi-Class Appliances: Peak Demand Reduction," In *IEEE International Conference on the European Energy Market (EEM)*, 2015, pages 1-6, doi: 10.1109/EEM.2015.7216702

[2] Rune Hylsberg Jacobsen, Armin Ghasem Azar, Qi Zhang, and Emad Samuel Malki Ebeid, "Home Appliance Load Scheduling With SEMIAH," In *Fourth International Conference on Smart Systems, Devices, and Technologies (SMART)*, 2015, pages 1-2, Link to paper

[3] Rune Hylsberg Jacobsen, Dominique Gabioud, Gillian Basso, Pierre-Jean Alet, Armin Ghasem Azar, and Emad Samuel Malki Ebeid, "SEMIAH: An Aggregator Framework for European Demand Response Programs," In *IEEE Euromicro Conference on Digital System Design (DSD)*, 2015, pages 470-477, doi: 10.1109/DSD.2015.96

[4] Armin Ghasem Azar, "Demand Response Driven Load Scheduling in Formal Smart Grid Framework," *Technical Report Electronics and Computer Engineering, Aarhus University*, vol. 4, no 24, 2016, pages 1-35, ISSN: 2245-2087, Link to report

[5] Armin Ghasem Azar and Rune Hylsberg Jacobsen, "Appliance Scheduling Optimization for Demand Response," *International Journal on Advances in Intelligent Systems*, vol. 2, no 1&2, 2016, pages 50-64, Link to paper

[6] Rune Hylsberg Jacobsen, Armin Ghasem Azar, and Emad Samuel Malki Ebeid, "Design of an Event-Driven Residential Demand Response In-

frastructure," In *IEEE Euromicro Conference on Digital System Design (DSD)*, 2016, pages 38-45, doi: 10.1109/DSD.2016.105

[7] Armin Ghasem Azar, Emad Samuel Malki Ebeid, and Rune Hylsberg Jacobsen, "A Formal Framework for Modeling Smart Grid Applications: Demand Response Case Study," In *IEEE Euromicro Conference on Digital System Design (DSD)*, 2016, pages 46-54, doi: 10.1109/DSD.2016.61

[8] Armin Ghasem Azar and Rune Hylsberg Jacobsen, "Agent-Based Charging Scheduling of Electric Vehicles," In *IEEE Online Conference on Green Communications (OnlineGreenComm)*, 2016, pages 64-69, doi: 10.1109/OnlineGreenCom.2016.7805408

[9] Armin Ghasem Azar, Hamidreza Nazaripouya, Behnam Khaki, Chi-Cheng Chu, Rajit Gadh, and Rune Hylsberg Jacobsen, "A Non-Cooperative Framework for Coordinating a Neighborhood of Distributed Prosumers," Submitted to *IEEE Transactions on Smart Grid*, 2017

In addition to the published articles and submitted manuscript, the following manuscripts are currently under preparation.

- Armin Ghasem Azar and Rune Hylsberg Jacobsen, “SALSA: A Formal Hierarchical Optimization Framework for Smart Grid"
- Rune Hylsberg Jacobsen and Armin Ghasem Azar, “SEMIAH: Scalable Energy Management Infrastructure for Aggregation of Households"
- Aisha Umair and Armin Ghasem Azar, “An Agent-Based Coordination Approach for Large-Scale Demand Response Optimization"

This dissertation has been submitted for assessment in partial fulfillment of the PhD degree. The dissertation is based on the submitted or published scientific papers which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the dissertation. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty. The dissertation is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Associate Prof. Rune Hylsberg Jacobsen, for his continuous support, patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this dissertation. I could not have imagined having a better supervisor for my PhD study. Additionally, I would like to thank the PhD committee, Profs. Peter Palensky, Bo Nørregaard Jørgensen, and Associate Prof. Stefan Hallerstede, for their interest in my work. This PhD study has been funded by European Union FP7 program under grant agreement n^{o} 619560 (SEMIAH).

My sincere thanks also goes to Professor Rajit Gadh and Dr. Chi-Cheng (Peter) Chu, who provided me an opportunity to join SMERC at UCLA as a visiting PhD student. Without their precious support it would not be possible to conduct this research. I would particularly like to single out Hamidreza and Behnam for the stimulating discussions, sleepless nights we were working together before deadlines, and for all the fun we have had during my six-month stay at UCLA.

My experience at Aarhus University would had definitely not been the same without my dear colleagues, Qi Zhang, Peter Lykke, Emad Ebeid, Sergi Griful, Søren Mikkelsen, Femina Beevi, Ubbe Welling, Egon Kidmose, Jacob Jeppesen, and Jorge Miranda. Not only were we able to support each other by deliberating over our problems and findings, but also happily by talking about things other than just our research.

I am grateful to my parents, Behnaz and Mohammad Hossein, and sister, Ayda, for allowing me to realize my own potential. All the support they have provided me over the years was the greatest gift anyone has ever given me. You are always there for me. Finally, I would like to thank my wife, Zohreh. She was always cheering me up and stood by me through the good times and bad. Her support and encouragement have seen me through tumultuous times. I thank her for bringing joy to my life in so many different ways.

Armin Ghasem Azar
Aarhus University, August, 2017

List of Figures

List of Tables

List of Algorithms

Acronyms

ALC	Application Level Contribution
AOD	Average Appliance Operation Delay
BESS	Battery Energy Storage System
CIM	Common Information Model
DER	Distributed Energy Resource
DO	Decided to Operate
DR	Demand Response
DRS	Demand Response System
DSO	Distribution System Operator
DW	Decided to Wait
ECT	Electricity Consumption Threshold
EDF	Earliest Deadline First
EFSM	Extended Finite State Machine
EMS	Energy Management System
EV	Electric Vehicle
EVSE	Electric Vehicle Supply Equipment
FIFO	First In First Out
FUR	Average Flexibility Usage Rate
GUI	Graphical User Interface
GVPP	Generic Virtual Power Plant
HEMG	Home Energy Management Gateway

HVAC	Heating, Ventilation, and Air Conditioning
ICT	Information and Communication Technology
IEC	International Electrotechnical Commission
IS	Immediately Start
IW	Immediately Wait
LDS	Load Demand Scheduling
LRT	Latest Release Time
LST	Least Slack Time
MAS	Multi-Agent System
MCT	Maximum Computation Time
MO-MINLP	Multi-Objective Mixed Integer Nonlinear Programming
MOEA	Multi-Objective Evolutionary Algorithm
MOO	Multi-Objective Optimization
NIST	National Institute of Standards and Technology
NMS	Network Management System
NSGA-III	Non-dominated Sorting Genetic Algorithm-III
OMS	Operating Management System
OpenDSS	Open Distribution System Simulator
PAR	Peak-to-Average Ratio
PCB	Average Prosumer Cost-Benefit
PDR	Peak Demand Reduction
PV	Photovoltaic
QoS	Quality of Service
RES	Renewable Energy Source
RMS	Rate-Monotonic Scheduling
SALSA	Scalable Aggregation of Load Schedulable Appliances
SEMIAH	Scalable Energy Management Infrastructure for Aggregation of Households

SEP2	Smart Energy Profile 2.0
SGAM	Smart Grid Architecture Model
SLA	Service Level Agreement
SLC	System Level Contribution
SLR	Average Self Load-Satisfaction Rate
SOC	State of Charge
SSR	Average Self Sufficiency Rate
UML	Unified Modeling Language
USEF	Universal Smart Energy Framework
VPP	Virtual Power Plant

Nomenclature

Notes:

- Words "Customer," "Consumer," "Prosumer," "Household," and "Dwelling" are used interchangeably in this document.
- All notations in this document follow this rule: $①_{②}^{③}(④)$, where ① refers to an entity, ② is the index of the entity owner, ③ defines a feature of the entity, and ④ is the time/iteration index.

Constants

$\mathcal{A}$	Aggregator
$\widetilde{K}$	Number of feasible behavior matrices of the aggregator
ana	Analog component of a device
$sa_{j,i}$	j-th appliance of prosumer ρ_i
$\varepsilon_{j,i}$	Exact operating end time of appliance $sa_{j,i}$
$\beta_{j,i}$	Desired operating end time of appliance $sa_{j,i}$
$ofl_{j,i}$	Desired operating flexibility of appliance $sa_{j,i}$
$\mathfrak{M}_i$	Number of appliances of prosumer ρ_i
$ost_{j,i}$	Desired operating start time of appliance $sa_{j,i}$
app	Software application
B_i	BESS of prosumer ρ_i
B_i^{cap}	BESS capacity of prosumer ρ_i (kWh)
$B_i^{\overline{chg}}$	Maximum charging power of the BESS of prosumer ρ_i (kW)

$B_i^{\overline{d\text{-}chg}}$	Maximum discharging power of the BESS of prosumer ρ_i (kW)
$B_i^{\overline{\text{SOC}}}$	Maximum SOC for the BESS of prosumer ρ_i
$B_i^{\underline{\text{SOC}}}$	Minimum SOC for the BESS of prosumer ρ_i
bv	Behavioral view of a software application
$comm$	Communicational part of a device
e	Communciation overhead element
v	Number of communication overhead elements
$comp$	Computational part of a device
n	Number of computational functions
m	Number of computational overhead elements
f	Computational function
ζ	Constraint element of a smart appliance
χ	Constraint element of an OMS
a	Number of constraint elements of a smart appliance
l	Number of constraint elements of an OMS
∇	Negotiation convergence tolerance
$\mathscr{C}$	Number of solutions crossovered in the NSGA-III
pc	Crossover probability in the NSGA-III
$\mathbb{C}$	Customers domain
h	Number of customers
ϵ	Decay rate controller for time-dependent concession values
dev	Device
dig	Digital component of a device
$dist$	Topological distance value between two connected devices
drs	Demand response system
$elec$	Electrical element
ems	Energy management system of a customer
en	Structural entitiy
ev	Event

ept	Event pooling time
est	Event sent time
gw	Gateway
$\mathcal{W}$	Number of generations in the NSGA-III
$\mathcal{G}$	The grid
is	Immediately Start buffer
iw	Immediately Wait buffer
info	Informational element
intr	Interruptibility feature of a smart appliance
mech	Mechanical part of a device
mob	Mobility type of a device
$\mathscr{M}$	Number of solutions mutated in the NSGA-III
pm	Mutation probability in the NSGA-III
$\mathcal{T}$	Number of negotiation iterations
nw	Network component
nms	Network management system of a DSO
σ	Objective of a customer
$\jmath$	Number of objectives of a customer
g	Number of objectives of a DSO
ς	Objective of a DSO
oms	Operation management system of a DSO
opr	Operating program of a smart appliance
$\mathbb{O}$	Operations domain
o	Overhead element
phy	Physical part of a device
pref	Tuple including appliance operating start time, program, and flexibility
ϑ	Projection operator
ρ_i	i-th prosumer

K	Number of feasible behavior pairs of prosumers
PV_i	PV system of prosumer ρ_i
PV_i^{cap}	Power generation capacity of the PV of prosumer ρ_i (kW)
y	Number of quality of service elements
x	Quality of service element
re	Structural relationship
rsp	Response
rst	Response sent time
do	Decided to Operate buffer
dw	Decided to Wait buffer
sch	Scheduler
λ	Scheduling strategy
$shift$	Shiftability feature of a smart appliance
sg	Smart grid entity
w	Smart grid domain
sm	Smart meter
sv	Structural view of a software application
$\widetilde{\hbar}_i$	Temperature flexibility degree of prosumer ρ_i (°C)
$\Delta\hbar_i$	Temperature fluctuation degree of prosumer ρ_i (°C)
tsp_i	Set point temperature of prosumer ρ_i (°C)
$\Delta\tau$	Time interval resolution
T	Number of time intervals
$\mathscr{U}$	Number of solutions selected by the tournament selection procedure in the NSGA-III
$U_{\mathcal{A}}$	Utility function of the aggregator
$U_{\mathcal{V}}$	Utility function of the VPP
$\mathcal{V}$	VPP

Indexes

$\widetilde{k}$	Behavior matrix index

j	Appliance index
d	Electricity price index
$\mathscr{A}$	Generation index
ϱ	Negotiation iteration index
i	Prosumer index
k, k'	Behavior pair indexes
t	Time interval index

Sets

SA_i	Set of appliances of prosumer ρ_i
$Apps$	Multiset of software applications
BV	Multiset of behavioral views
buf	Set of buffers
z	Vector of communication overheads
b	Vector of communication interfaces
r	Computational vector
q	Communicational vector
ct_{sa}	Set of constraints of a smart appliance
$\mathbb{D}$	Multiset of devices
ep	Set of electricity prices (over a day)
$\emptyset$	Empty set
lp	Load profile of a smart appliance
$lp_{j,i}$	Load profile of appliance $sa_{j,i}$
$\mathbb{B}$	Binary numbers
$\mathbb{N}$	Natural numbers
$\mathbb{R}$	Real numbers
NW	Multiset of network components
$\mathcal{NS}_i$	Set of non-shiftable appliances of prosumer ρ_i
obj_{C}	Set of objectives of a customer
obj_{O}	Set of objectives of a DSO

$\mathcal{P}$	Set of prosumers
QoS	Vector of quality of service elements
$\mathcal{S}_i$	Set of shiftable appliances of prosumer ρ_i
$\mathbb{A}$	Multiset of smart appliances
sf	Vector of smart features of a smart appliance
ω	Set of smart grid domains
Ψ	Multiset of smart grids
s	Set of behavioral states
SV	Multiset of structural views
ι	Set of transitions between behavioral states

Variables

$\widetilde{\ell}_{\mathcal{A}}(t)$	Set of feasible behavior matrices of the aggregator
$\mathscr{Z}(t)$	The zone of agreement in the negotiation
$ec_{j,i}(t)$	Load demand of appliance $sa_{j,i}$ (kW)
$B_i^{chg}(t)$	Charging power of the BESS of prosumer ρ_i (kW)
$B_i^{d\text{-}chg}(t)$	Discharging power of the BESS of prosumer ρ_i (kW)
$B_i^{\circledast}(t)$	Amount of energy stored in the BESS of prosumer ρ_i until time interval t (kWh)
$B_i^{\text{SOC}}(t)$	SOC value of the BESS of prosumer ρ_i
$v_i^{chg}(t)$	Binary charging status of the BESS of prosumer ρ_i
$v_i^{d\text{-}chg}(t)$	Binary discharging status of the BESS of prosumer ρ_i
$\widetilde{\mathcal{BM}}_{\mathcal{A}}^{\widetilde{k}}(t)$	$\widetilde{k}$-th behavior matrix of the aggregator
$\mathcal{BP}_i^k(t)$	k-th behavior pair of prosumer ρ_i
$\mathcal{BP}_i^{res}(t)$	Reservation behavior pair of prosumer ρ_i
$\widetilde{\mathcal{BP}}_i^{res}(t)$	Reservation behavior pair of the aggregator assumed for prosumer ρ_i
$dec_{j,i}(t)$	Binary decision variable of the operating status of appliance $sa_{j,i}$
$\kappa_{\mathcal{A}}(\varrho)$	Reactive concession value of the aggregator
$\kappa_{\mathcal{V}}(\varrho)$	Reactive concession value of the VPP

$\Pi_{\mathcal{A}}(\varrho)$	Desired utility value of the aggregator
$\Pi_{\mathcal{V}}(\varrho)$	Desired utility value of the VPP
$ect_{\mathcal{A}}^{\mathscr{H}}(t)$	Hard ECT of the aggregator (kW)
$ect_{i}^{\mathscr{S}}(t)$	Soft ECT of prosumer ρ_i (kW)
$P^{\mathfrak{u}}(t)$	Maximum offerable price for prosumers to trade electric energy (\$/kWh)
$P^{\mathbb{l}}(t)$	Minimum offerable price for prosumers to trade electric energy (\$/kWh)
$P_{\mathcal{G}}^{\mathfrak{u}}(t)$	Price of the grid to sell electric energy to the aggregator (\$/kWh)
$P_{\mathcal{G}}^{\mathbb{l}}(t)$	Price of the grid to buy electric energy from the aggregator (\$/kWh)
$P_i(t)$	Price of prosumer ρ_i offered to the aggregator via the VPP (\$/kWh)
$P_i^{res}(t)$	Reservation price offer of prosumer ρ_i (\$/kWh)
$\widetilde{P}_i(t)$	Price of the aggregator offered to prosumer ρ_i via the VPP (\$/kWh)
$Z_{\mathcal{A}}(t)$	Set of feasible desired offer packages of the aggregator
$Z_{\mathcal{V}}(t)$	Set of feasible desired offer packages of the VPP
$\overline{Z}_{\mathcal{A}}(\varrho)$	Subset of feasible desired offer packages of the aggregator
$\overline{Z}_{\mathcal{V}}(\varrho)$	Subset of feasible desired offer packages of the VPP
$flex_{j,i}(t)$	Binary flexibility status of appliance $sa_{j,i}$
$\mathcal{F}_i(t)$	Flexibility point (a triple set) of prosumer ρ_i
$\Im_i(t)$	Flexibility zone of prosumer ρ_i
$\mathcal{O}_{\mathcal{A}}(\varrho)$	Offer package sent from the aggregator to the VPP
$\mathcal{O}_{\mathcal{A}}^{res}(t)$	Reservation offer package of the aggregator
$\mathcal{O}_{\mathcal{V}}(\varrho)$	Offer package sent from the VPP to the aggregator
$\mathcal{O}_{\mathcal{V}}^{res}(t)$	Reservation offer package of the VPP
$\aleph_i(t)$	Power exchanged between prosumer ρ_i and the grid (kW)
$\aleph_i^{\mathrm{B2G}}(t)$	Power transferred from the BESS of prosumer ρ_i to the grid (kW)
$\aleph_i^{\mathrm{B2L}}(t)$	Power transferred from the BESS to appliances of prosumer ρ_i (kW)
$\aleph_i^{\mathrm{G2B}}(t)$	Power transferred from the grid to the BESS of prosumer ρ_i (kW)
$\aleph_i^{\mathrm{G2L}}(t)$	Power trasnferred from the grid to appliances of prosumer ρ_i (kW)
$\aleph_i^{\mathrm{P2B}}(t)$	Power transferred from the PV to the BESS of prosumer ρ_i (kW)

$\aleph_i^{\mathrm{P2G}}(t)$	Power transferred from the PV of prosumer ρ_i to the grid (kW)
$\aleph_i^{\mathrm{P2L}}(t)$	Power transferred from the PV to appliances of prosumer ρ_i (kW)
$\aleph_i^{res}(t)$	Reservation power offer of prosumer ρ_i to exchange (kW)
$\widetilde{\aleph}_{\mathcal{A}}(t)$	Power exchanged between the aggregator and the grid (kW)
$\widetilde{\aleph}_i(t)$	Power exchanged between the aggregator and prosumer ρ_i (kW)
$\overline{\aleph}_i(t)$	Extreme value of maximizing the comfort level of prosumer ρ_i
$\underline{\aleph}_i(t)$	Extreme value of maximizing the benefit of prosumer ρ_i
$\varphi_{\mathcal{A}}(\varrho)$	Offer package projection weight of the aggregator imposed on the offer package received from the VPP
$\varphi_{\mathcal{V}}(\varrho)$	Offer package projection weight of the VPP imposed on the offer package received from the aggregator
$\ell_i(t)$	Set of feasible behavior pairs of prosumer ρ_i
$PV_i^G(t)$	PV generation of prosumer ρ_i (kW)
$\mathcal{SI}_{\mathcal{A}}^{\tilde{k}}(t)$	Satisfaction index of offer package $\mathcal{O}_{\mathcal{A}}^{\tilde{k}}(\varrho)$ sent from the aggregator to the VPP
$\mathcal{SI}_{\mathcal{A}}^{res}(t)$	Satisfaction index of reservation offer package $\mathcal{O}_{\mathcal{A}}^{res}(t)$ of the aggregator
$\mathcal{SI}_i^k(t)$	Satisfaction index of behavior pair $\mathcal{BP}_i^k(t)$ of prosumer ρ_i
$\mathcal{SI}_i^{res}(t)$	Satisfaction index of reservation behavior pair $\mathcal{BP}_i^{res}(t)$ of prosumer ρ_i
$\mathcal{SI}_{\mathcal{V}}(\varrho)$	Satisfaction index of offer package sent from the VPP to the aggregator
$\mathcal{SI}_{\mathcal{V}}^{res}(t)$	Satisfaction index of reservation offer package $\mathcal{O}_{\mathcal{V}}^{res}(t)$ of the VPP
$\hbar_i(t)$	Household temperature of prosumer ρ_i (°C)
$\mathcal{X}_i(t)$	Percentage for the ECT of prosumer ρ_i (only ECT-P1 & 2)
$\mu_{\mathcal{A}}(\varrho)$	Time-dependent concession value of the aggregator
$\mu_{\mathcal{V}}(\varrho)$	Time-dependent concession value of the VPP
$\xi(\varrho)$	Weighted offer package

Part I

Summary

Chapter 1

Introduction

This chapter introduces the reader to the dissertation. This is done by first motivating the need of investigation in this research field. Afterwards, research hypotheses defined and methods followed during the PhD studies are introduced. Next section introduces the EU FP7 SEMIAH project followed by the list of the scientific publications. The chapter is finalized by describing the reading guidelines and by outlining the structure of the document.

1.1 Research Motivation

In the last two decades, the demand for electricity has risen exponentially, and it will continue to grow remarkably. In Europe, the electricity demand in the residential sector is expected to rise on average 56% from its 2000 level until 2050 with an annual growth rate of 1.1% [12]. Advancement in technologies and new services for the smart grid enable novel solutions for energy system integration while respecting the stability and security needed in the context of an increasing share of Renewable Energy Source (RES) in the electricity grid. European countries are progressing towards the development of smart grid concepts for the establishment of an efficient market for trading flexibility in electricity consumption and production [13]. The concept is based on a wholesale model, which determines the future roles of actors in the electricity market.

With the advent of smart grids, new solutions are becoming available. Major manufacturers have focused on the development of smart appliances, but their large market uptake is not expected to occur in the short-term. Demand Response (DR), defined as changes in electricity usage by customers from their normal consumption patterns in response to price or other signals, intends to improve energy efficiency and reduce peak demand [14]. However, 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[1] and were responsible for 10% of the CO_2 emissions in 2007[2]. Currently, DR is in its nascent stage in Europe with the existing programs essentially aimed at large industrial customers, which are easier to manage as one large client represents hundreds of households in terms of energy consumption. Until now, DR implementation has been strongly inhibited by the following barriers:

1. **System complexity**: The management systems for customers and utilities are proprietary and do not host a platform for 3^{rd} party applications such as DR. Hence, for the large deployment of smart grids, a single and robust formal framework for formalizing equipment inside each household as well as the outside world is required;

2. **Lack of an interoperable energy optimization system and aggregators**: Aggregation services are fundamental to efficiently balance supply and demand, and to ensure that both utility companies as well as customers derive benefits. In order to support the large implementation of DR systems, a significant deployment of advanced load demand optimization and control systems, including advanced aggregation, scheduling, and matching components, is required. The system infrastructure must be scalable, easily expandable, provide functionalities in real-time, adapt to current hierarchy of power grid, integrate large penetration of RESs, such as Photovoltaics (PVs), and Battery Energy Storage Systems (BESSs) as well;

3. **Lack of efficient negotiation models for DR**: Currently, clear business models for DR have not been defined yet, particularly due to a lack of methodologies for the quantification of costs and benefits for energy utilities as well as customers. In fact, the amount of flexibility of single household's appliances is too small for trading transactions of energy and ancillary services. Therefore, the suggested aggregation and disaggregation of a huge number of devices is crucial and equally important. Tools to estimate the financial benefits are also needed. Therefore, developing an approach, which is integrated with the DR, to enable the negotiation between customers and aggregators [15] within a reasonable time is inevitable.

1.2 Research Hypotheses

The aforementioned challenges must be overcome to: i) ensure the deployment of technologies to efficiently and securely manage energy consumption in households; ii) significantly increase the substitution of conventional generation (fossil fuels-based) with RESs, and iii) reduce/shift peak loads. This

[1] Eurostat-Final energy consumption, by sector (see this link for more information)
[2] European Commission (2010) EU Energy Figures in 2010. CO_2 emissions by sector

dissertation, therefore, addresses the following research hypotheses:

Hypotheses:

- **RH-1**: Develop a generic and robust formal framework for smart grid, to be automatically adaptable to various grid domains, actors, and their interior entities, based on design principles in software engineering;
- **RH-2**: Introduce a novel Scalable Aggregation of Load Schedulable Appliances (SALSA) system, derived from the formal framework, including the implementation of: i) a decentralized Energy Management System (EMS) for prosumers, to individually coordinate and control their smart appliances, PVs, BESSs, and to trade their heterogeneous flexibilities in the market through an aggregator; and ii) an efficient Operating Management System (OMS) for the aggregator to match a scalable number of prosumers' demand with their supply in real-time;
- **RH-3**: Proposing a bilateral multi-issue negotiation approach, to enable energy trading and matching between prosumers and the aggregator in a non-cooperative environment, with guaranteed convergence to a solution without sharing private information.

1.3 Research Methods

The research conducted in this dissertation has been carried out as an engineering PhD study, where a significant amount of work have been performed in collaboration with the EU FP7 Scalable Energy Management Infrastructure for Aggregation of Households (SEMIAH) project. The aim of the research has been to create usable solutions that address specific scopes and needs (see the next section for more information). The hypotheses of this dissertation have been broken down into several sub-hypotheses, which have been addressed by two main scientific contribution levels: i) System Level Contributions (SLCs) to enable a formal framework for smart grid application, and ii) Application Level Contributions (ALCs) to develop a smart grid application based on SLCs. The research method includes the following phases, as Fig. 1.1 shows its design model:

1. **Problem identification**: This phase includes identification and formulation of a research gap under a particular area of study. The research area is typically triggered by a challenge identified in the SEMIAH project.
2. **Background research**: This phase carries out a literature survey on how other researchers have tried to solve similar problems.
3. **Sub-hypothesis construction**: This phase builds a comprehensive and

yet simple sub-hypothesis on how the identified problem can potentially be solved. It specifies solution design goals and appropriate approaches to solve the problem. Existing state of the art technologies in the area of study are examined and potential enhancements are investigated.

4. **Implementation**: This phase materializes the sub-hypothesis through different means, such as simulations, co-simulations, discussions, etc. The implementation also makes a system level verification and validation possible. For instance, without implementation it would be hard to quantify the performance of e.g, different scheduling mechanisms.

5. **Evaluation**: This phase evaluates the results obtained from the implementations done against the constructed sub-hypothesis.

6. **Dissemination**: This phases, by publishing the findings, aims at sharing the results with the scientific community.

It should be noted that the method followed has not been sequential, but rather recursive: the completion of one phase may lead to go to a prior phase (e.g., literature survey may lead to identification of new problems).

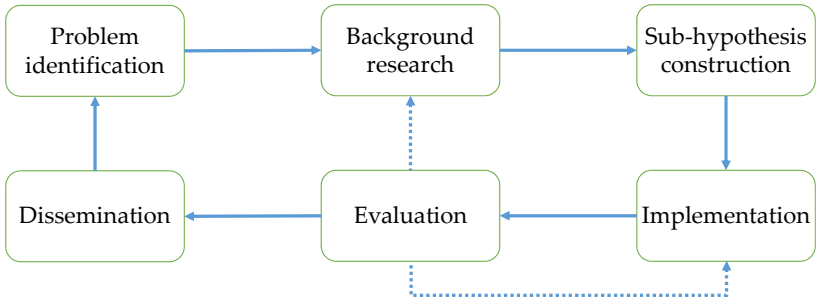


Fig. 1.1: Design model of the used research method.

1.4 The SEMIAH Project

The consortium behind SEMIAH project is interested in pursuing a major technological and scientific breakthrough by developing a novel Information and Communication Technology (ICT) infrastructure for the implementation of DR programs in households [3]. The SEMIAH infrastructure enables the shifting of energy consumption of high energy-consuming loads to off-peak periods with high generation of electricity from 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)). This will imply a step-change innovation in the field where there are currently no similar solutions. To validate this new infrastructure and for assessing the potential impacts, a large-scale simulation of up to 200.000 households will be performed. 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.

The flexibility concept of SEMIAH aligns with the European mandate M/490 [16]: "The flexibility [offering] concept assumes that parties connected to the grid produce offerings of flexibility in load and (distributed) generation. Thereby, so-called flex-offers are issued indicating these power profile flexibilities, e.g., shifting in time or changing the energy amount. In the flex-offer approach, consumers and producers directly specify their demand and supply power profile flexibility in a fine-grained manner (household and SME level)." In SEMIAH, flexibility from home appliances are aggregated in a coherent way to produce flex-offers that can be traded in the electricity markets. 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 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 Distribution System Operator (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 [1].

Fig. 1.2 demonstrates the SEMIAH technical architecture, which is characterized by two back-end and front-end systems. This PhD project has contributed to "SEMIAH Intelligence" and "DSO Grid Constraints" parts as components of the back-end system [3]. The SEMIAH Intelligence aims at scheduling the load requests received from the Home Energy Management

Gateways (HEMGs) subject to grid constraints imposed by the DSO. It finally forwards the scheduling decisions to the Generic Virtual Power Plant (GVPP). This dissertation has contributed to the SEMIAH intelligence and DSO grid constraints parts, which are reflected in various project deliverables.[1][2][3]

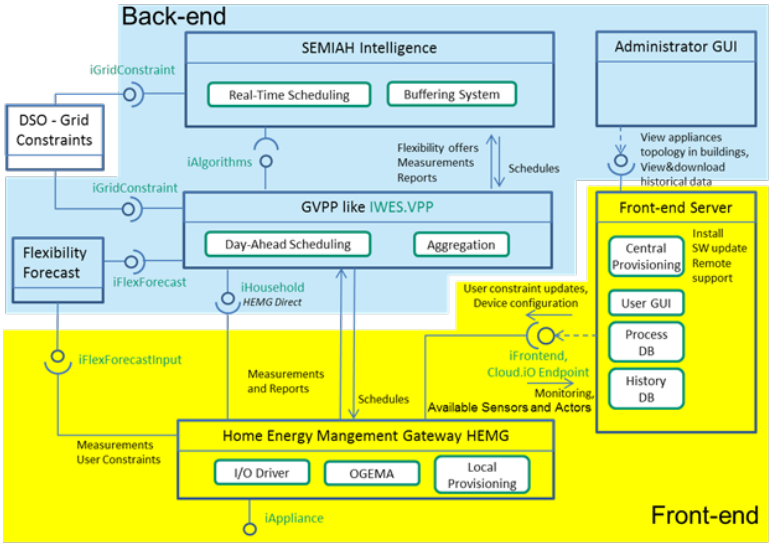


Fig. 1.2: SEMIAH technical architecture.

1.5 Scientific Publications

The research done during the three-year PhD study has led to nine publications: one journal article, seven peer-reviewed conference papers, and one technical report. At the time of writing this dissertation, one journal manuscript is under review. Furthermore, two journal and one conference manuscripts are under preparation.

1.5.1 Published

[1] Armin Ghasem Azar, Rune Hylsberg Jacobsen, and Qi Zhang, "Aggregated Load Scheduling for Residential Multi-Class Appliances: Peak Demand Reduction," In *IEEE International Conference on the European Energy Market (EEM)*, 2015, pages 1-6, doi: 10.1109/EEM.2015.7216702

[1] D4.3: Demand response prototypes (not public)
[2] D5.1: Algorithms for demand response and load control (link to the deliverable)
[3] D5.4: Back-end system release (not public)

[2] Rune Hylsberg Jacobsen, Armin Ghasem Azar, Qi Zhang, and Emad Samuel Malki Ebeid, "Home Appliance Load Scheduling With SEMIAH," In *Fourth International Conference on Smart Systems, Devices, and Technologies (SMART)*, 2015, pages 1-2, Link to paper

[3] Rune Hylsberg Jacobsen, Dominique Gabioud, Gillian Basso, Pierre-Jean Alet, Armin Ghasem Azar, and Emad Samuel Malki Ebeid, "SEMIAH: An Aggregator Framework for European Demand Response Programs," In *IEEE Euromicro Conference on Digital System Design (DSD)*, 2015, pages 470-477, doi: 10.1109/DSD.2015.96

[4] Armin Ghasem Azar, "Demand Response Driven Load Scheduling in Formal Smart Grid Framework," *Technical Report Electronics and Computer Engineering, Aarhus University*, vol. 4, no 24, 2016, pages 1-35, ISSN: 2245-2087, Link to report

[5] Armin Ghasem Azar and Rune Hylsberg Jacobsen, "Appliance Scheduling Optimization for Demand Response," *International Journal on Advances in Intelligent Systems*, vol. 2, no 1&2, 2016, pages 50-64, Link to paper

[6] Rune Hylsberg Jacobsen, Armin Ghasem Azar, and Emad Samuel Malki Ebeid, "Design of an Event-Driven Residential Demand Response Infrastructure," In *IEEE Euromicro Conference on Digital System Design (DSD)*, 2016, pages 38-45, doi: 10.1109/DSD.2016.105

[7] Armin Ghasem Azar, Emad Samuel Malki Ebeid, and Rune Hylsberg Jacobsen, "A Formal Framework for Modeling Smart Grid Applications: Demand Response Case Study," In *IEEE Euromicro Conference on Digital System Design (DSD)*, 2016, pages 46-54, doi: 10.1109/DSD.2016.61

[8] Armin Ghasem Azar and Rune Hylsberg Jacobsen, "Agent-Based Charging Scheduling of Electric Vehicles," In *IEEE Online Conference on Green Communications (OnlineGreenComm)*, 2016, pages 64-69, doi: 10.1109/OnlineGreenCom.2016.7805408

1.5.2 Under Review

[9] Armin Ghasem Azar, Hamidreza Nazaripouya, Behnam Khaki, Chi-Cheng Chu, Rajit Gadh, and Rune Hylsberg Jacobsen, "A Non-Cooperative Framework for Coordinating a Neighborhood of Distributed Prosumers," Submitted to *IEEE Transactions on Smart Grid*, 2017

1.5.3 In Preparation

- Armin Ghasem Azar and Rune Hylsberg Jacobsen, “SALSA: A Formal Hierarchical Optimization Framework for Smart Grid"

- Rune Hylsberg Jacobsen and Armin Ghasem Azar, “SEMIAH: Scalable Energy Management Infrastructure for Aggregation of Households"
- Aisha Umair and Armin Ghasem Azar, “An Agent-Based Coordination Approach for Large-Scale Demand Response Optimization"

1.6 Document Structure

This dissertation has two parts. Part I contains main contributions. Part II consists of a chapter for each scientific publication.

Part I is divided into eight chapters. Each chapter in Part I starts with a paragraph describing the content. This document should be read sequentially as the comprehension of each chapter depends on the previous ones. Reading these chapters, the scientific contributions[1] of the dissertation are described in a rectangular box as follows:

Contribution XXX-#. This is a contribution.

To ease the traceability, each contribution is know with "three letters" (abbreviation for the contribution level) followed by a number. One contribution can be related to one or several publications. Overview of contributions and their connections are provided in Chapter 8. This dissertation makes various contributions to smart grid through SLCs and ALCs. The former targets research hypothesis RH-1 while the latter addresses research hypotheses RH-2 and RH-3.

Chapter 1 introduces the reader to the dissertation. Chapter 2 provides the common background information required for the comprehension of the remaining of the document. All scientific contributions and discussions of this dissertation are provided in Chapters 3-7. Chapter 3 describes a high-level view of the proposed formal framework. Chapter 4 details all optimization models and algorithms incorporated in the proposed SALSA system. Chapter 5, using the optimization models proposed in the previous chapter, defines the agent-based negotiation approach. Chapter 6 describes the simulation setup and results. Chapter 7 presents the interactive web service developed for the SALSA. Finally, Chapter 8 concludes the dissertation and provides future work.

Part II is also divided into eight different chapters each of them containing the details of each publication. Full articles have been included when re-print copyrights have been obtained. In each chapter, there is a brief description of the current state of the publication (published or under review), a full citation and also a statement of the contribution of each co-author. The authors’ contribution describes the work that each individual has done in each manuscript.

[1] A contribution is seen as an achievement presented to the scientific community.

Chapter 2

Background

This chapter, with respect to research hypotheses defined in Section 1.2, categorizes the state of the art into three sections followed by positioning the contributions of this PhD dissertation accordingly.

2.1 Formalization of Smart Grid

In recent years, there has been an increasing interest in investigating concerns about the inefficient structure of the current electrical grid for responding the growing electricity demand [17,18]. Farhangi [17] examined different impacts of transforming the current electrical grid to a complex system of systems named the smart grid. Fang *et al.* [18] discussed that engaging ICT with the smart grid could play a major role in enabling technologies for smart grid data communications.

To handle these data communications, Godfrey *et al.* [19] presented a co-simulation framework to model both the communication network and the power system. They employed a baseline scenario and demonstrated the responses to power fluctuations subject to considering any communication efficiency (e.g., Quality of Service (QoS)). Afterwards, Schutte *et al.* [20] delivered the Mosaik framework for a co-simulation of various scenario descriptions, grid topology, and control strategies using their semantic information. The framework lacks precise details in the model description since it has just considered a single naive life entity while current electrical infrastructure includes different varieties of entities such as cables, step-down transformers, etc. Finally, Montenegro *et al.* [21] presented an Open Distribution System Simulator (OpenDSS) for the smart grid. It is a simulation tool for the electrical power system principally for the electricity distribution grid. However, it failed in calculating mathematical models to develop a real-time simulation for available devices in the power grid and for showing their real behavior and communications.

Before the smart grid becomes fully operational in managing the electricity operations in a sustainable and reliable manner, it requires technological advancements in a number of interdisciplinary perspectives. National Institute of Standards and Technology (NIST), by identifying main smart grid domains, has developed a smart grid conceptual model describing their stakeholders and feasible communication paths. Domains are customers, markets, service providers, operations, bulk generation, transmission, and distribution [15]. 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. To easily identify such requirements, International Electrotechnical Commission (IEC) has proposed reliable and reproducible standards. In the same context, Universal Smart Energy Framework (USEF) has recently delivered a common standard clearly specifying interactions and roles of the aforementioned domains [11].

The model-based design of the smart grid as a robust formal framework is currently limited and not well supported. Andrén *et al.* [22] also recognized this issue and proposed a semantically-driven design method using Common Information Model (CIM) for transmission (IEC 61970-301) and distribution (IEC 61968-11). These standards have been highly promoted for modeling grid issues and the corresponding device/component communications. Additionally, since the smart grid requires a specific standard for communication networks and power utility automation systems, IEC 61850 has been launched in the course of an object-oriented information model. Nonetheless, an ICT-driven formal framework is needed to overcome the major shortcoming of the standards above, i.e., a limited number of covered domains beside discarding grid physics, communication and control issues.

Smart Grid Architecture Model (SGAM) intended to present the design of smart grid use cases in an architectural way [23]. To handle the model, it introduced five interoperability layers allowing the representation of entities and their relationships in the context of smart grid domains. This dissertation, to build a more generic profile, maps the *component*, *information*, and *communication* layers into *hardware*, *software*, and *network* aspects of the framework, respectively. Although SGAM provides the structural design of the smart grid applications in a high-level approach, however, it lacks the behavioral part describing feasible actions and behaviors of each actor.

Unified Modeling Language (UML) is widely applied to software modeling and the demonstration of its specifications based on hardware/software/network co-designs [24–26]. De Miguel *et al.* [24] introduced UML extensions for the representation of temporal requirements and resource usage of real-time systems. Their tools generated a model for the OPNET simulator. Hennig *et al.* [25] described a UML-based simulation framework for performance assessment of hardware/software systems described as sequence diagrams. The proposed simulator was based on the discrete event simulation package OMNet++.

2.2 Energy Management Systems and Scheduling Approaches

Recently, there has been an increasing amount of literature on incentivizing consumers to shift their electricity consumption by varying the electricity prices [27, 28]. Sou *et al.* [27] investigated the minimization of electricity bills combined with enforcing uninterruptible and sequential operation model constraints. Nonetheless, the utilized mixed integer linear programming approach to solve the scheduling problem is not scalable and the appliance classification is limited to the interruptibility feature. In addition, [28] has proposed an electricity load scheduling algorithm that controls the operation time and energy consumption of appliances based on adapting time-of-use pricing to minimize the total electricity bill. A serious weakness with this argument, however, is that the authors have used solely one smart house as the test-bed.

On the other hand, a solution to the problem of optimally scheduling a set of residential appliances under the day-ahead variable peak pricing scheme has been studied in [29]. Here, the objectives are minimizing the electricity bills and spreading the electricity usage out in each time interval simultaneously. On the contrary, they have considered a limited number of appliances. Finally, the focus in [30] is on applying the priority-based appliance methodology to quantify preferences of consumers for using appliances during peak times based on the Knapsack problem approach. Nonetheless, in the proposed mechanism, there is no consumption constraint to prevent consumers from exceeding it.

Electrical grids increasingly depend on RES production that is weather-dependent, often fluctuating and difficult (or impossible) to plan and control. To smoothen RES production, the use of physical storage (e.g., batteries and electrical vehicles) and virtual storage (e.g., DR systems) are currently being considered [31, 32]. As significant energy amounts are involved and substantial flexibility (elasticity) is available to operate the devices within the allowed bounds, and due to their broad availability, storage loads are of great value [33] for different smart-grid applications, e.g., demand and supply balancing [34], grid congestion problem solving [35], and electricity market trading [36]. However, large storage systems are still considered to be very costly and DR has been considered to a feasible way to provide a cost-effect virtual storage.

Today, most commercial DR services in Europe are leveraged by large consumers that e.g., find flexibility potentials in their production processes. More recently, attention has shifted towards smaller consumers such as residential households. Consumers may be engaged in event-driven (also known as incentive-based) DR programs [37]. In these programs, home appliances can be invoked in response to a variety of trigger conditions, including envi-

ronmental parameters (e.g., temperature); local or regional grid congestion; economics; or operational reliability requirements. For DR programs that foster an improved integration of renewable energy, aggregators can shift consumption to periods with lower CO_2 intensity or electricity prices.

Vardakas *et al.* [32] presented a survey on different DR schemes and programs according to their control mechanism, offered motivations, and decision variable. The authors reviewed various optimization algorithms for optimal operation of the smart grid. Di Giogio and Pimpinella [38] presented the design of a smart home controller strategy providing efficient management of electric energy in a domestic environment. Their work, to implement a pervasive control platform, provided an integration between ICT and automation, which allowed consumers to automatically fulfill the terms of previously subscribed contracts, while assuring cost-effective use of energy. The deployed controller is event-driven by reacting to events from the environment such as requests from the consumer for the execution of loads and signals from a demand-side control.

Early forms of event-driven DR include the old-fashioned ripple control acting as an emergency reserve [14]. Ripple control aims at protecting power systems in the emergency conditions caused by critical contingencies. Wang *et al.* [39] proposed an event-driven emergency DR scheme to improve the stability of the power system from experiencing voltage collapse. The proposed scheme was able to provide key setting parameters such as the amount of demand reductions at various locations to prepare the DR infrastructure and hereby act as a balancing asset for the grid in case of emergency.

Residential appliance load scheduling, to reduce the electricity bills following price fluctuations, has been explored by a number of researchers [40–42]. A central energy management system, with the aim of minimizing the grid purchase, is proposed in [40] and [41], where it controls households' appliances based on their reputations in storing their surplus PV generation in a shared BESS. A distributed version of such system is proposed in [42], where under a dynamic pricing system, a coordination strategy fairly controls the operation of appliances while respecting the transformer capacity limits. Even though these models incentivize the households to modify their consumption pattern to achieve lower electricity bills, they fail to study the impact of the high penetration of households, with different ownership levels of shiftable appliances and various flexibility types, on households' economic as well as on the distribution grid.

Coordinated charging scheduling of Electric Vehicles (EVs) is also a challenging problem, in which Mukherjee *et al.* [43] have reviewed the recent contributions. They argued that considering only grid-to-EV power flows could be a logical first step toward challenging more complex bidirectional models. Rassaei *et al.* [44] studied the impact of a game-based DR framework on shaping the aggregated charging profiles taking uncertain arrival times into account. Although the work succeeded to minimize the peak, however, the consumers' comfort level was not considered. Mohsenian-Rad *et al.* [45]

proved that the dependability of optimizing the charging scheduling of EVs on the uncertain departure times is undeniable. They developed a closed-form solution to this problem with respect to time-of-use electricity prices.

However, a majority of current unidirectional model-driven works made centralized decisions for coordinating a high penetration of EVs, which, due to the large number of decision-making processes and communications, were computationally intractable to handle. Xu *et al.* [46] framed a distributed concept into a hierarchical framework for coordinated charging of EVs. He *et al.* [47] proposed a centralized charging scheduling framework for charging and discharging of EVs, in which consumers could use it to minimize their energy cost. Deilami *et al.* [48] proposed a charging load scheduling algorithm for residential EVs using the amount of energy purchased in the day-ahead market based on forecasting methods.

Veit *et al.* [49] provided a Multi-Agent System (MAS)-based framework to the DR scheduling problem in response to the real-time supply. They formulated the constraints of individual scheduable electrical devices under the agent's control. Hu *et al.* [50] proposed an agent-based centralized concept for scheduling EVs including different layered agent types. Unda *et al.* [51] also presented an agent-based method for managing the battery charging problem of EVs in power distribution network according to electricity prices and grid stability constraints.

2.3 Negotiation Approaches

Presently, most trading of flexibility takes place bilaterally between companies that can interrupt their power consumption for some periods, and power grid system operators, possibly through an electricity trading company. Rather than investing in grid expansions, the system operators in some countries can pay, through market agreements, large electricity users to reduce the consumption in concerned hours so that the congestion in the grid is avoided. To realize the potential of providing flexibility, aggregators are requested to pool offers of reduced and shifted electricity demands into aggregated offers to the electricity market or to system operators [52]. Upon market acceptance, the aggregators actuate the flexible consumption according to a defined schedule by shifting electricity demand to meet the contractual obligations of the offer. Concurrent bilateral negotiations at each instant of time in a non-cooperative environment enable the power exchange between prosumers and the aggregator. The amount of power that each prosumer intends to trade changes during the negotiation. There is a risk that such fluctuations leads to an infeasible matching solution [53]. Furthermore, negotiating with increasing number of prosumers to reach an overall agreement in a reasonable time is computationally expensive.

A very comprehensive review of scheduling problems of distributed energy resources, such as PVs, from various aspects is done in [53], where the

authors propose considering microgrids and Virtual Power Plants (VPPs) as two suitable potential solutions. Limited research has been conducted on developing a scalable real-time framework for coordinating scheduling, sharing, and matching tasks engaged with a non-cooperative negotiation approach respecting negotiators' private information [54–66].

In [54], a day-ahead demand-side management mechanism for prosumers formulated as a non-cooperative game with a single objective of reducing monetary expenses is proposed. It preserves prosumers' privacy, limited communication with the central unit is needed, and the peak is reduced by 12.6%. However, prosumers' load demand scenarios should be known in advance and cannot change during the process. They should also commit to follow strictly the resulting consumption pattern. In [55] and [56], a lead-acid BESS coupled with PV is modeled through a home energy management system. To quantify the self-consumption and self-sufficiency of the model, load demands are satisfied first by the PV, then by the energy stored in the BESS, and finally by the grid. The main challenge with this single-objective system is that they simply consider the excess energy is injected to the grid with a fixed rate without any negotiation. The challenges of rapid residential PV installations in the recent years is discussed in [57], where the authors, to overcome the difficulty in balancing supply and demand, propose three independent centralized, decentralized, and distributed approaches using small-scale distributed BESSs based on model predictive control methodologies.

In [58], given a real-time pricing scheme, a simple model for buildings with the basic components of a generator, a BESS, and loads is proposed allowing two-way energy trading via a broker based on differential game theory. The convergence condition and time, however, are only characterized based on a limited number of buildings. A distributed power sharing framework formulated as a repeated game between households in a microgrid is proposed in [59], where each household decides on amount of power to trade with the grid. Households, by taking advantage of the variability in their load consumption patterns, achieve cost savings up to 20%. However, they require to have a list of preferences of households, with which they individually prefer to negotiate. A submission-based double auction mechanism with linear functions for a set of prosumers, possessing PVs and BESSs, is proposed in [60], where the mechanism is able to achieve an exact demand supply balance in a day-ahead power market subject to having a full information of consumption and generation profiles.

In [61], a set of computationally expensive off-line and on-line algorithms for the real-time cooperative energy management of only two microgrids are presented. These algorithms, however, assume that the renewable energy generation offset, by the aggregated load of individual microgrids, is known ahead of time. A similar double cooperative game to minimize the overall costs of both the utility companies and the residential prosumers is formulated in [62]. In [63], the authors develop a strategy including a heuristic algorithm for optimizing decentralized energy exchange depending on pro-

sumers' involvement and physical constraints of distribution networks, in which prosumers' cost is averagely reduced by 66% and the proportion of energy self-satisfaction reaches 98%.

In [64], a peer-to-peer energy sharing model with price-based demand response for prosumer including PVs are introduced. Although an energy sharing provider is defined to coordinate the power exchanges, however, no method for ensuring the match between demand and supply is proposed. Furthermore, peak demand reduction before and after using the proposed model is only 2% while the computation time of the simulation is very high, e.g., around 175 seconds for 25 prosumers. A similar work is also studied in [65]. Recently, the authors in [66] propose a multi-agent system to collectively optimize the energy flows of a neighborhood of prosumers with private objective functions, which bilaterally negotiate with an aggregator only to reduce their purchasing cost.

2.4 Contributions of the PhD Dissertation

This dissertation, to account for the gaps identified in [14,27–34,36–49,51–66], makes the following key contributions:

- Far too little attention has been paid by smart grid application designers to build a formal framework based on mentioned standards concerning the scalability, interoperability, and updatability of domains. This dissertation, to model smart grid applications, at the first stage, puts some efforts into proposing a practical and robust formal framework. The framework formally describes each domain and its actors on the basis of "separation of concerns" design principle. Such descriptions can be trajected into formal models using UML techniques. These models can finally be converted into executable models, i.e., by following the Model-to-Text transformation approach, such as Python or Matlab code.

- As the second stage, this dissertation proposes the SALSA system derived from the smart grid formal framework proposed in the first stage. Fig. 2.1 shows a high-level view of its architecture. For the sake of simplicity, the SALSA only relies on the interconnectivity of customers and operations domains in the smart grid. It should be noted that the SALSA system is open to be expanded with other domains. 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. SALSA defines a smart grid, which is composed of a neighborhood of distributed prosumers communicating with an aggregator [53]. Each prosumer includes a set of smart appliances, a PV, a BESS, and has private consumption, generation, and storage flexibilities. To manage the resources within each

prosumers following their subjective views on their available flexibilities and the electricity prices adjusted by the market, SALSA develops a Multi-Objective Mixed Integer Nonlinear Programming (MO-MINLP) model [5] and solves it with the Non-dominated Sorting Genetic Algorithm-III (NSGA-III) [67]. The model confronts the prosumers' conflicting objectives of "maximizing the comfort level and profit" at the same time, where they make decisions on *offer packages* declaring the amount of "power" to sell/buy and "price" to trade it. To prevent any imbalance between generation and demand in the grid, SALSA considers the aggregator, which, by using a similar MO-MINLP model approached by the NSGA-III, matches surplus power with demand. It receives "demand with buying price" and "surplus with selling price" offer packages from prosumers and intends to "maximize its profit" and "minimize the grid purchase" simultaneously.

- This dissertation, at the third stage, incorporates a *bilateral* multi-issue negotiation approach in the SALSA system and defines a VPP, which on behalf of prosumers, negotiates with the aggregator on *aggregated* power and price offer packages with no private information shared between them [68]. The negotiation approach assumes that the VPP and the aggregator (negotiators) have *private* nonlinear utility functions and start the negotiation with an offer package providing the highest possible utility value. Each offer package is quantified based on *satisfaction index*, a novel evaluation metric to determine to which extend the offer package is able to optimize the relevant objectives. To guarantee the convergence within a reasonable time, the approach follows an alternating-offer production protocol and a utility value concession strategy. Negotiators continuously concede to their pre-defined reservation utility values (linked to their worst but feasible offer packages). That is, they neither propose nor accept any offer package with utility value lower than their reservation utility values.

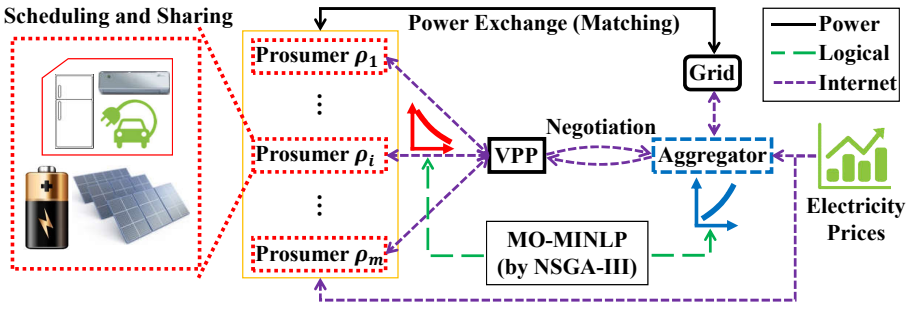


Fig. 2.1: System model of the SALSA [9].

Chapter 3

Formal Smart Grid Framework

Building multidisciplinary and interoperable smart grid applications requires an efficient way to model, integrate, and validate different grid aspects and components. To effectively design a formal and practical *framework* for the smart grid: i) its infrastructure must be scalable and specified together with its constituents; ii) its main model must be interoperable and supportive to the grid's future expansions; iii) its design must consider the grid's topology and its hardware/software/network aspects; and iv) its updating process must perform independently of the grid's aspects. This chapter proposes a formal framework to model smart grid applications, as Fig. 3.1 presents its top-down design methodology. This strategy helps the framework build reliable and self-healing smart grid applications. Moreover, it is entirely adaptable to various domains, their actors, and interior entities. The content of this chapter originates and adapts from the following publication:

[7] Armin Ghasem Azar, Emad Samuel Malki Ebeid, and Rune Hylsberg Jacobsen, "A Formal Framework for Modeling Smart Grid Applications: Demand Response Case Study," In *IEEE Euromicro Conference on Digital System Design (DSD)*, 2016, pages 46-54, doi: 10.1109/DSD.2016.61

3.1 Aspects and Modeling

The framework is established according to the "separation of concerns" design principle [69]. This principle allows the re-usability, development, and upgrading of the framework's components independently. Therefore, the framework orchestrates the smart grid system with respect to three most important aspects; i.e., *hardware*, *software*, and *network*.

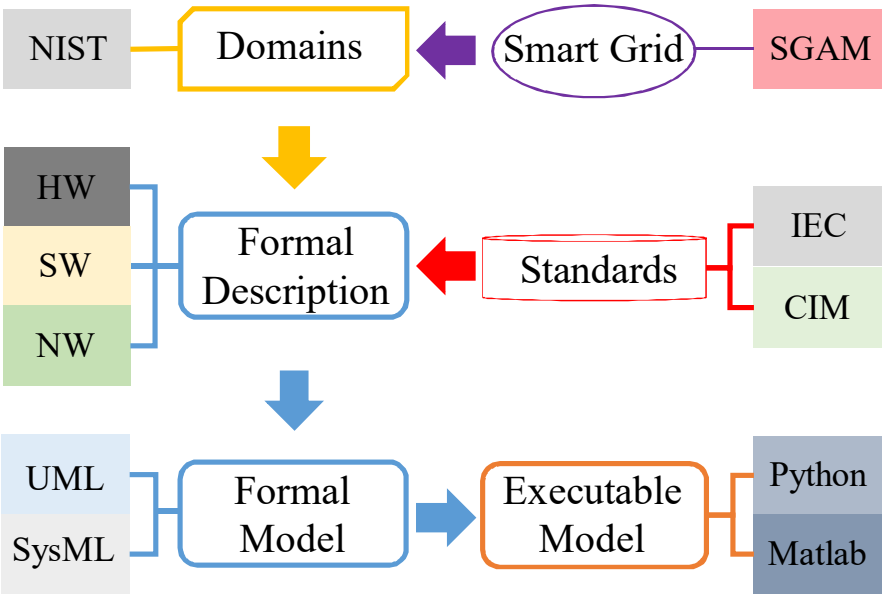


Fig. 3.1: Top-down design methodology of the proposed framework [7].

3.1.1 Aspects

Fig. 3.2 demonstrates an overview of the framework presenting these aspects. For instance, a gateway inside the territory of a customer is a hardware including a software, which communicates with other devices, such as an EV, and other domains, such as a DSO, through the network aspect. The framework is formalized using grid component definitions followed in architectural guidelines of the future smart grid such as IEC standards [22, 70] and SGAM [23]. They organize how application characteristics exchange data model specifications among described aspects. The following parts explain how these aspects are formalized.

3.1.1.1 Hardware Aspect

The smart grid comprises different *hardware devices* employed in various domains. Devices can be digital, analog, or heterogeneous with discrete or continuous behavior. They are characterized based on different structures and responsibilities. The hardware aspect defines the physical model of these devices. To understand the general constituents of a device helps the formal framework allocate appropriate responsibilities to each different device from both computational and communicational points of view. For instance, a HEMG can perform optimization techniques and message broadcasting op-

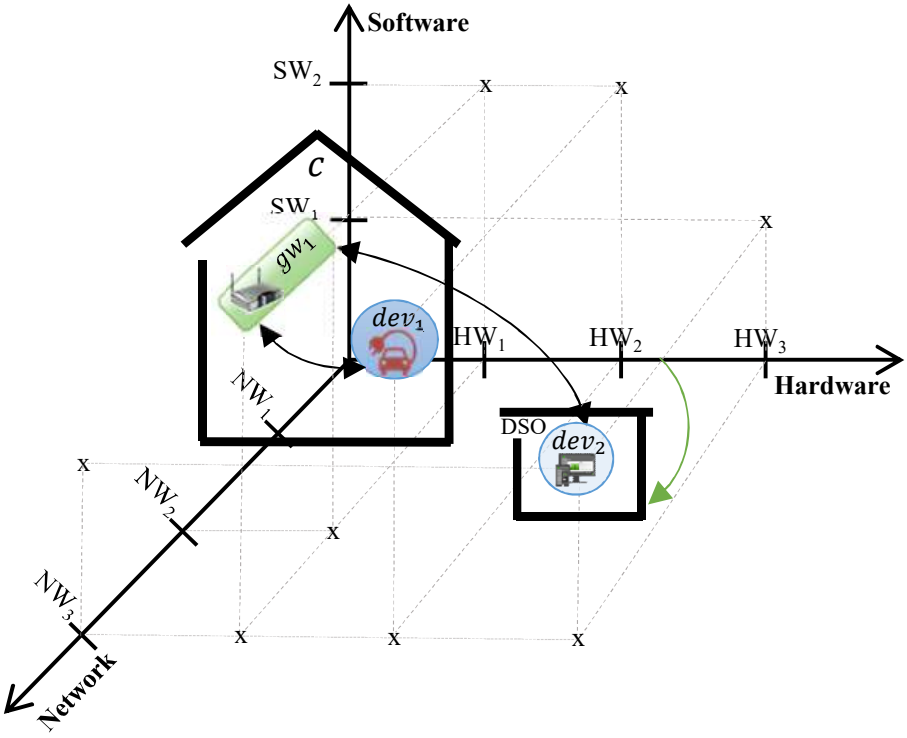


Fig. 3.2: Overview of the hardware, software, and network aspects [7].

erations with certain specifications. Let

$$dev = [dig, ana] \in \mathbb{D}, \tag{3.1}$$
$$dig = [comp, comm], \tag{3.2}$$
$$ana = [phy, mech], \tag{3.3}$$

where each device *dev* is a member of a multiset of devices $\mathbb{D}$ including digital *dig* and analog *ana* components. Investigating the latter component is outside of the scope of this dissertation. A digital component consists of computational *comp* and communication *comm* components. The computational component, e.g., a CPU, performs computing operations. The communication component, e.g., a network interface, distributes the information to other devices (using the network aspect). Let

$$comp = [r, q], \tag{3.4}$$
$$r = [f_1, f_2, \ldots, f_n] \in app, \tag{3.5}$$
$$q = [o_1, o_2, \ldots, o_m] \in \mathbb{R}_{\geq 0}, \tag{3.6}$$

where the computational component *comp* includes computation r and overhead q vectors. The former includes $n \in \mathbb{N}$ function elements. Each element performs a specific procedure, for instance, actuating an appliance. This is done as a software application *app* running in a hardware entity, e.g., CPU. The latter contains $m \in \mathbb{N}$ overhead elements. Each represents the processing time of running a subset of function elements. The information, processed in the computational component, is distributed to other devices through the communication component using the network aspect *nw*. Let

$$comm = [b, z], \tag{3.7}$$
$$b = [elec, info] \in nw, \tag{3.8}$$
$$z = [e_1, e_2, \ldots, e_v] \in \mathbb{R}_{\geq 0}, \tag{3.9}$$

where the communicational component *comm* includes vectors of communication interfaces b and communication overheads z. The vector of communication interfaces b, as a member of a network component *nw*, includes electricity *elec* and information *info* elements. The former is responsible for satisfying the electricity demand of the device through physical power lines. The latter, to exchange the information with other devices, is performed on top of a communication protocol. The vector of communication overheads z comprises $v \in \mathbb{N}$ communication overhead elements. Each element is caused by characteristics of the communication media of each corresponding device. Packet loss rate, latency, and throughput are some examples of communication overheads.

3.1.1.2 Software Aspect

The *software* aspect aims at developing platform-independent models of software *applications* that are executable in *hardware* devices. Software applications can describe miscellaneous functionalities, ranging from the basic operation of an appliance (e.g., ON/OFF) to heterogeneous communication protocols (e.g., the Smart Energy Profile 2.0 (SEP2) [71]). The framework considers the software development from the object-oriented programming point of view. Let

$$app = [sv, bv] \in Apps, \tag{3.10}$$
$$sv = [en, re] \in SV, \tag{3.11}$$
$$bv = [s, \iota] \in BV, \tag{3.12}$$

where application *app* is a member of a multiset of applications *Apps* consisting of two correlated structural *sv* and behavioral *bv* views. The former describes the structure of the application while the latter presents its dynamics. The structural view *sv*, as a member of a multiset of structural views *SV*, is a combination of entities *en* and relationships *re*. An entity, which can be periodic or aperiodic, represents the functional part of an application. A

relationship defines a logical connection between entities. The behavioral view *bv*, as a member of a multiset of behavioral views *BV*, represents the dynamical view of an application. An Extended Finite State Machine (EFSM) can capture such behavior, where a set of states *s* describes the application's actions while a set of transitions ι provides conditional paths between them. The detailed description of EFSM can be found in [72].

3.1.1.3 Network Aspect

The *network* aspect enables the communication between two or more devices according to CIM in IEC 61970 standards series [70]. It defines the communication media that delivers the information broadcast between *software* and *hardware* aspects. Its characteristics shape its performance, for instance, the latency, throughput, and packet loss. Let

$$nw = [QoS, dist, mob] \in NW, \tag{3.13}$$

$$QoS = [x_1, x_2, \ldots, x_y] \in \mathbb{R}_{\geq 0}, \tag{3.14}$$

$$dist \in \mathbb{R}_{\geq 0}, mob \in \mathbb{B}, \tag{3.15}$$

where each network component *nw* is a member of a multiset of network components *NW*. The *QoS* vector includes $y \in \mathbb{N}$ evaluation parameters x representing the overall performance of the communication network. The distance *dist* indicates the topological distance value between two connected device entities *dev*. Finally, mobility *mob* represents the device's mobility type, i.e., wired or wireless.

3.1.2 Modeling

This section shows the trajection of the previously described mathematical equations of the framework's aspects into a formal model. Fig. 3.3 illustrates a novel UML profile, in which aspects are labeled with red-dashed rounded rectangles. The used UML meta-classes are:

1. **Class**: It is an extensible template for creating objects, providing initial values for attributes and implementations of the behaviors. It is used to model the `Digital` and `Analog` hardware aspects of the smart grid applications.

2. **Node**: It is a physical object that represents a computational resource of the system, such as servers. It is used to model the `Digital` hardware aspect of the smart grid applications.

3. **Device**: It is a type of node that represents a physical computational resource, such as a gateway. The device is used to model the `Digital` and `Analog` hardware aspects of the smart grid applications.

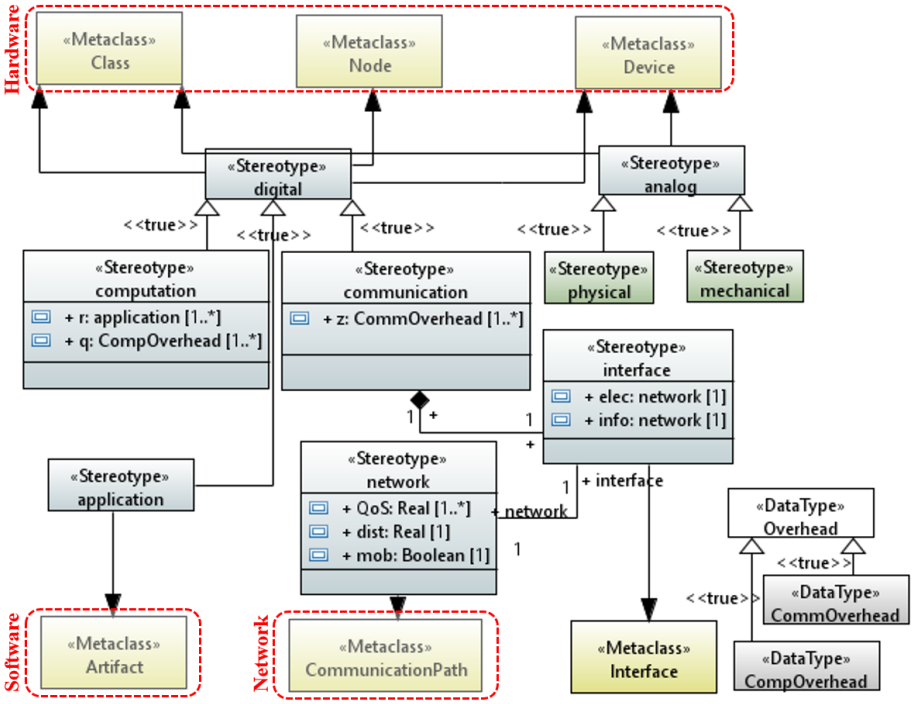


Fig. 3.3: UML profile diagram of aspects of the framework [7].

4. **Artifact**: It is a software component, such as an executable software component, files or libraries, deployed inside the **Node**. It is used to model the software `Application` aspect.

5. **CommunicationPath**: It defines the path between two nodes that are able to exchange signals and messages such as a wired/wireless communication channel. It is used to model the `Network` aspect of the smart grid applications.

6. **Interface**: It is a collection of operation signature and attribute definitions that ideally defines a cohesive set of behaviors. It is used to model the `Network` aspect of the smart grid applications in the sense of connections between devices and the network.

The following section emphasizes how the UML profile is used to enrich the semantics of the UML diagrams in representing smart grid applications.

3.2 Formalization

This section first, describes the formal framework based on the aspects defined in Section 3.1.1. Then, it models the framework mathematically using a UML smart grid class diagram. The smart grid formalization requires a high-level conceptual framework taking its domains into account. This work intends to both identify the actors inside each smart grid domain and establish their possible communication routes appropriately. Let

$$sg = [w] \in \Psi, \tag{3.16}$$

$$w \subseteq \omega, \tag{3.17}$$

where a smart grid entity sg 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 significant challenge about these domains is how to organize them to work consistently focusing on delivering appropriate services to their relevant interior actors. As the concept of the separation of concerns is employed, each domain corresponds to an *add-in* feature to the framework. Hence, adding or removing a domain will not affect the framework's functionality, which strengthens its robustness and flexibility.

This dissertation considers $w = [\mathbb{C}, \mathbb{O}]$ as the first step in formalizing a smart grid, where $\mathbb{C}$ and $\mathbb{O}$ correspond to *Customers* and *Operations* domains, respectively [15].

3.2.1 Customers Domain

The Customers domain typically provides customers (or prosumers, as used from the next chapter) with applications to manage their electricity consumption behaviors. Let

$$\mathbb{C} = [c_1, c_2, \ldots, c_h], \tag{3.18}$$

$$c = [SA, PV, B, ems, sm, gw], \tag{3.19}$$

$$SA \subseteq \mathbb{A}, ems \in app, \{PV, B, sm, gw\} \in dev, \tag{3.20}$$

where customers domain $\mathbb{C}$ includes $h \in \mathbb{N}$ customers. Each customer c has an individual set of smart appliances SA, as a subset of a multiset of smart appliances $\mathbb{A}$, a PV system PV , a BESS B, a smart meter sm, and a gateway gw. Customers are interested in enhancing the efficiency and profitability of their consumption, generation, and storage. This is done using an EMS, as a software application, running on a hardware device. The smart meter measures the electricity consumption of smart appliances and periodically sends them to the electric utility via the power line communication. The gateway gw is responsible for routing different device-to-device communications. Since PVs, BESSs, smart meters, and gateways are predefined entities, their descriptions

are discarded. The following defines *SA* and *ems* precisely.

3.2.1.1 Smart Appliances

They are the main drivers of electricity demands. Let

$$SA = [sa_1, sa_2, \ldots, sa_{\mathfrak{M}}], \tag{3.21}$$
$$sa = [sf, ct_{sa}, lp], \tag{3.22}$$
$$sf = [shift, intr] \in \mathbb{B}, \tag{3.23}$$
$$ct_{sa} = [\zeta_1, \zeta_2, \ldots, \zeta_a] \in \mathbb{R}_{\geq 0}, \tag{3.24}$$
$$lp = [ec, \Delta\tau], \tag{3.25}$$
$$ec \in \mathbb{R}_{\geq 0}, \Delta\tau \in \mathbb{R}_{>0}, \tag{3.26}$$

where each customer c possesses $\mathfrak{M} \in \mathbb{N}$ smart appliances. Each smart appliance $sa \in dev$ has a smart feature pair sf including two dependent Boolean functions of shiftability *shift* and interruptibility *intr* [5]. Shiftability allows smart appliances to shift their operating start times to the future. For instance, refrigerator and lighting systems are non-shiftable appliances whereas washing machine and the electric vehicle are shiftable appliances. Interruptibility allows smart appliances to interrupt their operating cycles in the middle. For instance, the operating cycle of an electric vehicle is both shiftable and interruptible. The dependency between these features indicates that a shiftable appliance can either be interruptible, i.e., $sf = [\text{TRUE}, \text{TRUE}]$ or uninterruptible, i.e., $sf = [\text{TRUE}, \text{FALSE}]$. Nevertheless, if it is non-shiftable, then, it is also uninterruptible, i.e., $sf = [\text{FALSE}, \text{FALSE}]$. In addition, each smart appliance has a set of constraints ct_{sa} including $a \in \mathbb{N}$ constraint elements ζ. For instance, each appliance should finish its operation cycle in the defined period. The smart appliance *sa* follows a specific load profile *lp* in each operating cycle. It is determined with respect to its program predefined by the corresponding customer. Each *lp* is presented as a vector of time-series electricity consumptions *ec* (kW) with a specific time resolution $\Delta\tau$ (min. or hrs.).

3.2.1.2 Energy Management System

It is a software application *app* running on a device *dev*. It can be a web service in the cloud or an application installed on a server. Providing combination of energy optimization and information processing functions, it integrates the efficiency into advanced control and optimization strategies. Let

$$ems = \left[obj_{\mathbb{C}}, pref^{\mathfrak{M}}, ev^{\mathfrak{M}}, rsp^{\mathfrak{M}}\right], \tag{3.27}$$
$$obj_{\mathbb{C}} = [\sigma_1, \sigma_2, \ldots, \sigma_J] \in \mathbb{R}_{\geq 0}, \tag{3.28}$$

$$pref = [ost, opr, ofl] \in \mathbb{R}_{\geq 0}, \tag{3.29}$$

$$ev = \left[est, ept, obj_{\mathbb{C}}, pref, sf, ct_{sa}, lp\right], \tag{3.30}$$

$$\{est, ept\} \in \mathbb{R}_{\geq 0}, \tag{3.31}$$

$$rsp = [dec, rst], \tag{3.32}$$

$$dec \in \mathbb{B}, rst \in \mathbb{R}_{\geq 0}, \tag{3.33}$$

where each customer, using an *ems*, adjusts its objective set $obj_{\mathbb{C}}$ including $\jmath \in \mathbb{N}$ distinct objectives. These objectives can be in conflict or in line with each other, for instance, minimizing the electricity cost and CO_2 emission, maximizing comfort level, minimizing appliance service delay, etc. In addition, for each smart appliance *sa* in *SA*, the customer provides a vector of operating preferences *pref*. It is a triple including customer's interested operating start time *ost*, operating program *opr*, and operating flexibility *ofl*. As the first element, customers adjust the time, at which they want to operate their smart appliance. As the second element, customers set a specific program to operate each smart appliance with (e.g., washing the clothes at 60 °C). This program influences the load profile *lp* mentioned before. As the last element, customers offer a voluntarily flexibility *ofl*. Two flexibility types, named *deadline* and *temperature*, are defined. Deadline flexibility is an additional time to the required period of the main operating cycle of physically-controllable smart appliances. Providing this flexibility, the corresponding smart appliances can be shifted and interrupted until reaching the adjusted deadline flexibility. Temperature flexibility is a feature of thermostatically-controllable smart appliances, e.g., Heating, Ventilation, and Air Conditioning (HVAC). It is notable that *ost* for these smart appliances equals to the operating set temperature. Similarly, offering the temperature flexibility, the household's temperature can fluctuate over the operating set temperature.

Once customers set the objectives and preferences, the EMS sends events *ev* to the OMS of the Operations domain (described later). Then, the EMS waits to receive responses *rsp* of the events sent. Notation *est* refers to the time, at which the event has been sent. The event pooling time *ept* defines a length of time, at which each smart appliance waits to receive a response from the EMS after sending the event. If no response arrives, another event is forwarded after *ept* (sec., min., or hrs.). 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 actuating smart appliances in accordance with the received responses.

3.2.2 Operations Domain

This domain handles the transport of electricity. This is facilitated by contracts traded in the energy market and fulfilled by the Operations Domain.

DSOs are the main actors of this domain. Current shortcomings of the electrical grid motivate them to employ the ICT to react upon the grid information [3]. They facilitate the continuous grid management functions. Their responsibilities include maintaining and operating the electricity distribution infrastructure efficiently while delivering the electricity to customers securely. Let

$$\mathbb{O} = [oms, nms], \tag{3.34}$$
$$\{oms, nms\} \in app, \tag{3.35}$$

where operations domain $\mathbb{O}$ includes two representative software applications Operating Management System (OMS) and Network Management System (NMS) derived from IEC 61970 [70]. The former, as an operational planning software, is responsible for performance monitoring and optimization of the electrical grid, e.g., load balancing and matching. The latter, as a network maintenance software, monitors the communication network of smart grid domains, for instance, fault management, overhead and delay calculation, etc. (beyond the scope of this dissertation).

3.2.2.1 Operating Management System

An aggregator or a DSO has access to an OMS inside the operations domain. Let

$$oms = \left[obj_{\mathbb{O}}, ct_{dso}, ep, drs\right], \tag{3.36}$$
$$obj_{\mathbb{O}} = [\varsigma_1, \varsigma_2, \ldots, \varsigma_g] \in \mathbb{R}_{\geq 0}, \tag{3.37}$$
$$ct_{dso} = [\chi_1, \chi_2, \ldots, \chi_l] \in \mathbb{R}_{\geq 0}, \tag{3.38}$$
$$ep = [P_1, P_2, \ldots, P_d] \in \mathbb{R}_{\geq 0}, \tag{3.39}$$
$$drs = [buf, sch, rsp^{\mathfrak{M}}], \tag{3.40}$$
$$buf = [iw, is, do, dw] \in \mathbb{R}_{\geq 0}, \tag{3.41}$$
$$sch = [obj_{\mathbb{O}}, ct_{dso}, ep, ev^{\mathfrak{M}}, \lambda], \tag{3.42}$$
$$\lambda \in \mathbb{R}_{>0}, \tag{3.43}$$

where oms has a set of objectives $obj_{\mathbb{O}}$ including $g \in \mathbb{N}$ distinguishable objectives ς. Flattening the aggregated electricity consumption, reducing the outages, reducing the CO_2 emission, etc. are some examples of objectives. In addition, ct_{dso} corresponds to a set of $l \in \mathbb{N}$ grid stability constraints χ. For instance, hard and soft Electricity Consumption Thresholds (ECTs) in the feeder/substation and household levels, active and reactive power flow capacities, etc. are some of such examples. The DSO adjusts the set of electricity prices ep over $d \in \mathbb{N}$ time periods in different schemes, i.e., real-time and day-ahead. Finally, the DSO employs a Demand Response System (DRS)

to respond to the events received from the EMSs of customers. 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 relocate them to different buffers. The scheduler *sch*, to make these decisions, follows a Load Demand Scheduling (LDS) mechanism λ. Decisions are made based on the information stored in the events. The scheduling approach can be either stochastic or deterministic applying single-objective or multi-objective optimization techniques.

3.3 UML Class Diagram

Fig. 3.4 shows a complete description of the proposed framework as a UML class diagram combined with the profile diagram (see Fig. 3.3). The `smartgrid` class is extended by the `«digital»` stereotype that contains the predefined framework's aspects (analog part is not considered). Objects can be instantiated and linked together to compose a variety of smart grid applications. Next part exemplifies such application as a case study.

3.4 Case Study

Figs. 3.5 and 3.6 show a "conceptual view" and "UML deployment diagram" of objects of one instantiated customer communicating with a DSO through a gateway, respectively. The customer has an EV and communicates with the DSO through a gateway that handles its preferences via the EMS system. On the other hand, the DSO receives events from the customer side and forwards the responses back including the decision about the load requests of the EV over time. Decisions take place through various elements, i.e., objectives, thresholds, and electricity prices. The DSO contains a server that has an OMS, which runs a DR scheduling algorithm. The interactions between the elements of the case study are presented in Fig. 3.7. Fig. 3.7(a) shows the UML sequence and activity diagram. Fig. 3.7(b) demonstrates the behavioral part of the DR scheduler elaborating how the scheduler decides about the incoming events. UML diagrams, to validate these high level models, have been manually mapped into a Matlab code. Table 3.1 shows the main mapping patterns of the UML diagrams and the generated executable Matlab code. In order to build an automatic synthesis tool, the developer needs to use a model of computation to capture the UML models and generate the corresponding code [72].

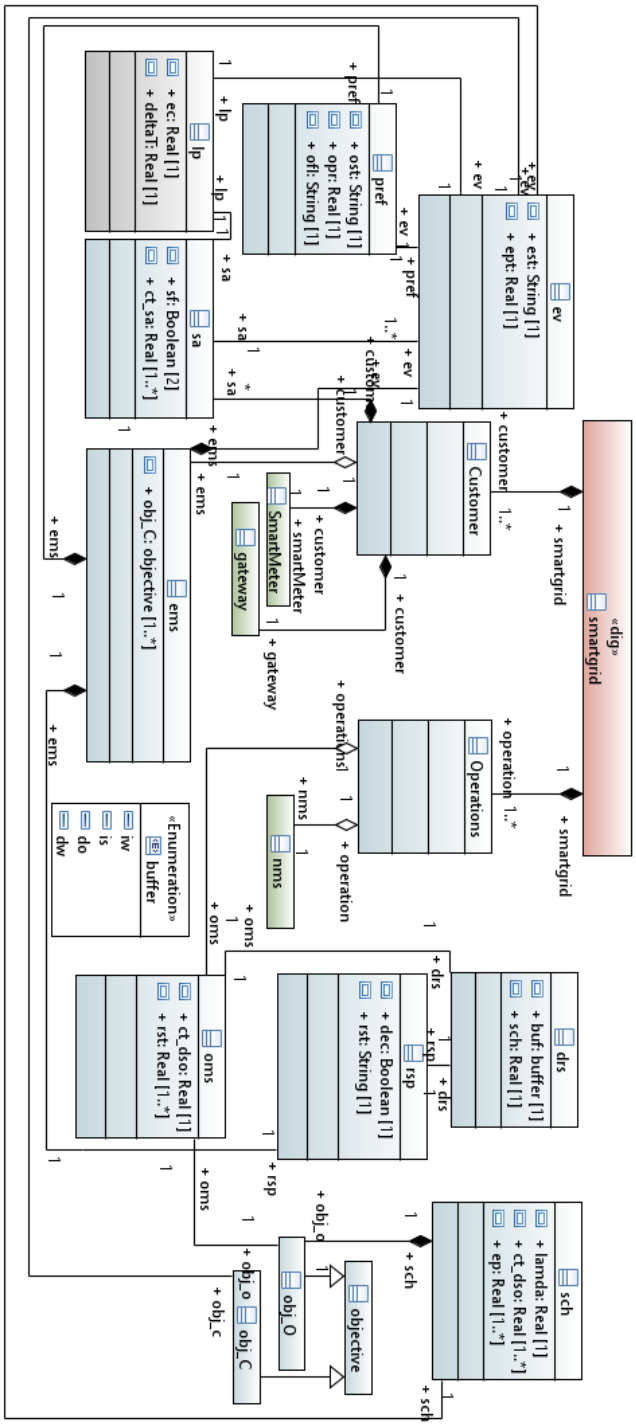
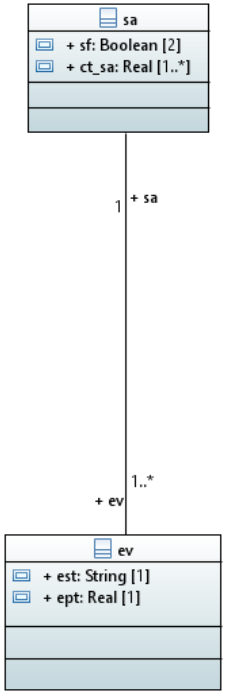
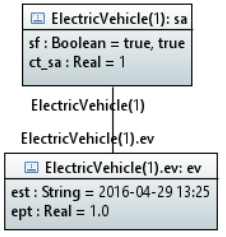


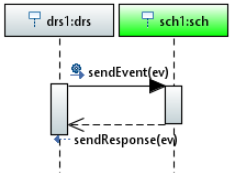
Fig. 3.4: UML class diagram of the elements of the framework [7].

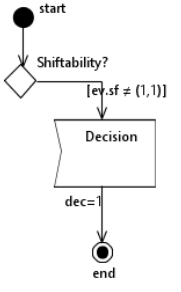
Table 3.1: Mapping patterns from UML diagrams to Matlab code [7].

UML entity	Matlab code

Classes: 	`classdef sa`
	`  properties`
	`  sf`
	`  ct_sa`
	`  ev`
	`  end`
	`  methods`
	`  function ...`
	`  obj=sa(sf,ct_sa)`
	`  obj.sf = sf;`
	`  obj.ct_sa = ...`
	`  ct_sa;`
	`  end`
	`  function obj = ...`
	`  CreateEvent(obj)`
	`  est = datetime;`
	`  ept = 2;`
	`  obj.ev = ...`
	`  Event(est,ept);`
	`  end`
	`  end`
	`end`
	`classdef Event`
	`  properties`
	`  est`
	`  ept`
	`  end`
	`  methods`
	`  function ...`
	`  obj=Event(est,ept)`
	`  obj.ept = ept;`
	`  obj.est = est;`
	`  end`
	`  end`
	`end`

Objects: 	`sf = [1,1];`
	`ct_sa = 1;`
	`ElectricVehicle(1) = sa(sf,ct_sa);`
	`ElectricVehicle(1) = ...`
	`  CreateEvent(ElectricVehicle(1));`

Sequence: 	`sendEvent(ev);`
	`iw = {iw,ev};`
	`Scheduler(iw);`
	`sendResponse(ev);`

Activity: 	`if ev.sf ~= [1,1];`
	`  dec = 1;`
	`end`

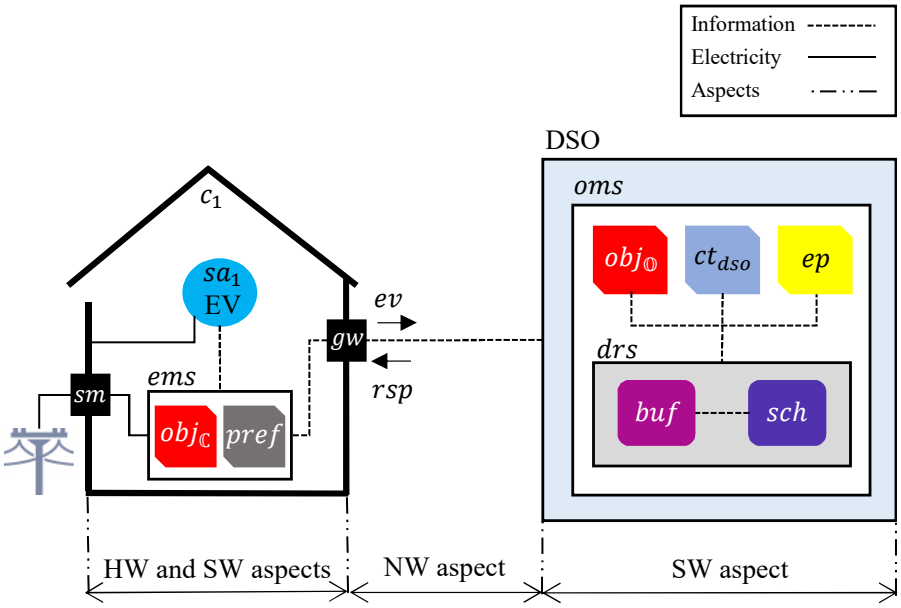


Fig. 3.5: Conceptual view of the case study [7].

3.5 Conclusions

This chapter proposed a formal framework for modeling smart grid applications. It defined the main grid elements using three aspects: hardware, software, and network. A UML profile was developed to integrate these aspects into a generic profile. Employing this profile, a formal framework for modeling the main semantics of smart grid systems was defined and mathematically formalized with an emphasis on the customers and operations domains. A novel UML profile and a class diagram were also developed to support the implementation of the framework, reflect the mathematical formulas, and create formal grid models. To prove the validity of the formal framework, a case study was developed demonstrating how to synthesize the formal framework into an executable code. Hence, this chapter has led to the following System Level Contributions:

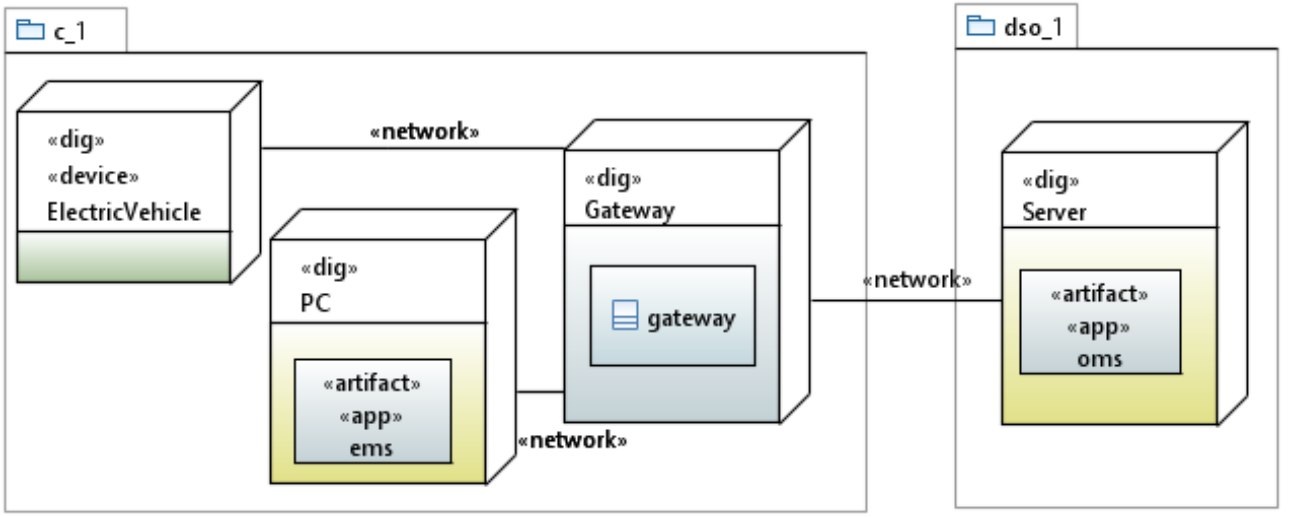
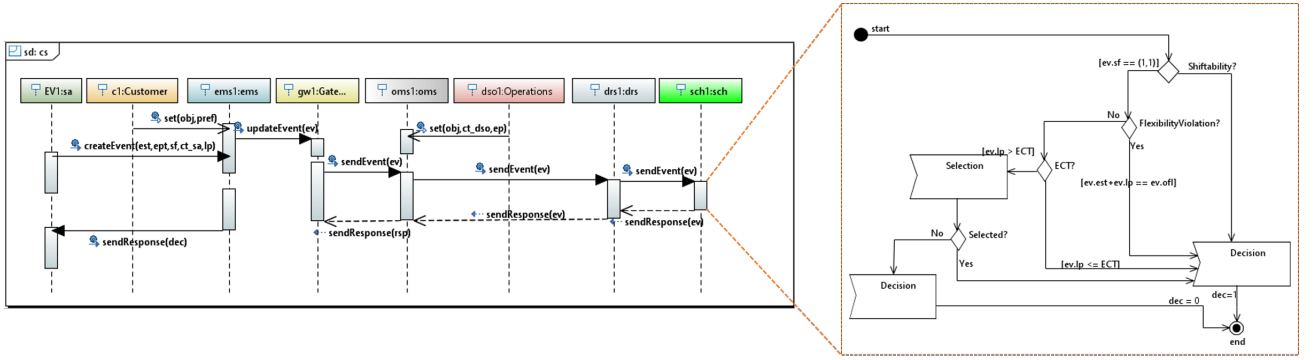


Fig. 3.6: UML deployment diagram of the instantiated objects [7].



(a) (b)

Fig. 3.7: (a) UML sequence diagram of the interactions of the instantiated objects, (b) UML activity diagram of the scheduling algorithm [7].

SLC-1: Proposing a generic formal framework by providing a smooth way to describe smart grid elements, domains, and their interactions;

SLC-2: Modeling and formulating three essential smart grid aspects, i.e., hardware, software, and network, to demonstrate the formal framework's scalability, reusability, interoperability, and updatability;

SLC-3: Modeling, designing, and developing two novel UML smart grid profiles to map the formal framework into various smart grid applications.

Next chapter, by using the proposed formal framework, develops a novel hierarchical optimization system for smart grid targeting interactions between customers and operations domains.

Chapter 4

Optimization Models and Algorithms for the SALSA

This chapter aims at developing the Scalable Aggregation of Load Schedulable Appliances (SALSA) system, which is formulated based on the formal framework presented in Chapter 3. SALSA is modeled and implemented based on an agent-based modeling approach. Agent-based modeling of real-world problems leads to more flexibility since: i) different behavioral criteria can be analyzed, and ii) it is possible to add more *properties to agents* or add more *distributed agents* to the model [49, 51]. Fig. 4.1 displays the mind map of the SALSA system. The content of this chapter originates and adapts from the following publications:

[1] Armin Ghasem Azar, Rune Hylsberg Jacobsen, and Qi Zhang, "Aggregated Load Scheduling for Residential Multi-Class Appliances: Peak Demand Reduction," In *IEEE International Conference on the European Energy Market (EEM)*, 2015, pages 1-6, doi: 10.1109/EEM.2015.7216702

[2] Rune Hylsberg Jacobsen, Armin Ghasem Azar, Qi Zhang, and Emad Samuel Malki Ebeid, "Home Appliance Load Scheduling With SEMIAH," In *Fourth International Conference on Smart Systems, Devices, and Technologies (SMART)*, 2015, pages 1-2, Link to paper

[4] Armin Ghasem Azar, "Demand Response Driven Load Scheduling in Formal Smart Grid Framework," *Technical Report Electronics and Computer Engineering, Aarhus University*, vol. 4, no 24, 2016, pages 1-35, ISSN: 2245-2087, Link to report

[5] Armin Ghasem Azar and Rune Hylsberg Jacobsen, "Appliance Scheduling Optimization for Demand Response," *International Journal on Advances in Intelligent Systems*, vol. 2, no 1&2, 2016, pages 50-64, Link to paper

Fig. 4.1: Mind map of the SALSA system.

[8] Armin Ghasem Azar and Rune Hylsberg Jacobsen, "Agent-Based Charging Scheduling of Electric Vehicles," In *IEEE Online Conference on Green Communications (OnlineGreenComm)*, 2016, pages 64-69, doi: 10.1109/OnlineGreenCom.2016.7805408

[9] Armin Ghasem Azar, Hamidreza Nazaripouya, Behnam Khaki, Chi-Cheng Chu, Rajit Gadh, and Rune Hylsberg Jacobsen, "A Non-Cooperative Framework for Coordinating a Neighborhood of Distributed Prosumers," Submitted to *IEEE Transactions on Smart Grid*, 2017

4.1 Prosumers

SALSA is represented by a set of prosumers (located in the customers domain), which are connected to the power grid distribution system. They communicate with an aggregator $\mathcal{A}$ (located in the operations domain), which trades their flexibilities (in terms of power and price) in market (will be described in Section 4.2). Prosumer $\rho_i \in \mathcal{P} = \{\rho_1, \ldots, \rho_h\}$, where $1 \leq i \leq h$, consists of a set of smart appliances, a PV system, a BESS, and an EMS. It has two main responsibilities: i) scheduling appliances, and ii) selling/buying power to/from the grid. Fig. 4.2 shows the prosumer's *power actions* at each time interval $t \in \mathbb{R}_{>0}$.

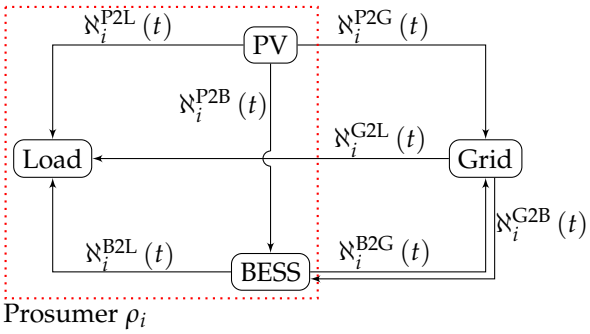


Fig. 4.2: Model diagram of power actions of prosumer ρ_i. The dotted red box conceptualizes the prosumer's physical equipment. "Load" points to the set of appliances. Exchanging power between prosumers and the "Grid" is controlled by the aggregator. Notations are described in Sections 4.1.1 to 4.1.3 [9].

4.1.1 Appliances

Appliances and main drivers of prosumers' electricity consumptions. Let

$$SA_i = \left\{ \overbrace{sa_{1,i}, \ldots, sa_{j,i}}^{\mathcal{NS}_i}, \overbrace{sa_{j+1,i}, \ldots, sa_{\mathfrak{M}_i,i}}^{\mathcal{S}_i} \right\}, \tag{4.1}$$

where SA_i is set of appliances of prosumer ρ_i, where each smart appliance $sa_{j,i} \in SA_i, 1 \leq j \leq \mathfrak{M}_i \in \mathbb{N}$ is a member of subsets of either non-shiftable $\mathcal{NS}_i \subseteq SA_i$ or shiftable $\mathcal{S}_i \subseteq SA_i$ appliances. *Shiftability* feature provides the prosumer with a *flexibility* degree to interrupt the operating cycle of appliances (see Section 3.2.1.1 and [5,73] for more information). Let

$$lp_{j,i} = \left\{ ec_{j,i}\left(ost_{j,i}\right), ec_{j,i}\left(ost_{j,i} + \Delta\tau\right), \ldots, ec_{j,i}\left(\beta_{j,i}\right) \right\}, \tag{4.2}$$

$$\sum_{t=ost_{j,i}}^{\beta_{j,i}} ec_{j,i}(t) \times dec_{j,i} = \left| lp_{j,i} \right|, \tag{4.3}$$

$$\sum_{j=1}^{\mathfrak{M}_i} ec_{j,i}(t) \times dec_{j,i}(t) = \aleph_i^{\text{P2L}}(t) + \aleph_i^{\text{B2L}}(t) + \aleph_i^{\text{G2L}}(t) \leq ect_i^{\mathscr{S}}(t), \tag{4.4}$$

where appliance $sa_{j,i}$ follows a specific load profile $lp_{j,i}$ during its operating cycle. It is determined with respect to the program preset by the prosumer. $ec_{j,i}(t) \in \mathbb{R}_{\geq 0}$ (kW) is the load demand of appliance $sa_{j,i}$ at time interval t specifying the amount of power it needs to operate between t and $t + \Delta\tau$, where $\Delta\tau$ is the time interval resolution. $dec_{j,i}(t) \in \mathbb{B}$ is the binary decision variable of load demand $ec_{j,i}(t)$ at time interval t. $\aleph_i^{\text{P2L}}(t)$,

$\aleph_i^{\text{B2L}}(t), \aleph_i^{\text{G2L}}(t) \in \mathbb{R}_{\geq 0}$ (kW) denote the power transferred from the PV, the BESS, and the grid to appliances, respectively. The prosumer aims at keeping the aggregated load demands below a *soft* ECT $ect_i^{\mathscr{S}}(t) \in \mathbb{R}_{\geq 0}$ (kW) at time interval t. This threshold sometimes due to the provided deadline flexibilities and non-shiftability feature of some appliances cannot be satisfied. Therefore, it is only applied on *shiftable* appliances, which: i) have not started yet, and ii) have started but still have enough flexibility. ECTs can follow various strategies, such as: i) a fixed percentage of the peak demand in the previous day, ii) a fixed percentage of the aggregated load demands at each time interval, and iii) real-time electricity prices. Section 4.1.4.2 describes these policies in more detail. Let

$$\begin{bmatrix} dec_{j,i}(t) = 1, \\ flex_{j,i}(t) = 0, \end{bmatrix} \quad \forall sa_{j,i} \in \mathcal{NS}_i, \tag{4.5}$$

$$\begin{cases} dec_{j,i}(t) \in \{0,1\} & flex_{j,i}(t) = 1, \forall sa_{j,i} \in \mathcal{S}_i, \\ dec_{j,i}(t) = 1 & \text{otherwise}, \end{cases} \tag{4.6}$$

$$\text{(Except HVAC)} \ flex_{j,i}(t) = \begin{cases} 0 & (\beta_{j,i} - ost_{j,i}) \leq \left(ofl_{j,i} - t\right), \\ 1 & \text{otherwise}, \end{cases} \tag{4.7}$$

$$\text{(Only HVAC)} \begin{cases} flex_{j,i}(t) = \begin{cases} 0 & \hbar_i(t-1) \geq \left(\widetilde{\hbar}_i + tsp_i\right), \\ 0 & \hbar_i(t-1) \leq \left(\widetilde{\hbar}_i - tsp_i\right), \\ 1 & \text{otherwise}, \end{cases} \\ \hbar_i(t) = \begin{cases} \hbar_i(t-1) + \Delta\hbar_i & dec_{j,i}(t) = 1, \\ \hbar_i(t-1) - \Delta\hbar_i & \text{otherwise}, \end{cases} \\ \widetilde{\hbar}_i - tsp_i \leq \hbar_i(t) \leq \widetilde{\hbar}_i + tsp_i, \end{cases} \tag{4.8}$$

where $flex_{j,i}(t) \in \mathbb{B}$ is the binary flexibility status of appliance $sa_{j,i}$. Decision variable for non-shiftable appliances always equals to one. These appliances operate uninterruptedly until their completion, since they are provided with no flexibility. For shiftable appliances, the prosumer can decide to satisfy the load demand $\left(dec_{j,i}(t) = 1\right)$ or postpone it to the next interval $\left(dec_{j,i}(t) = 0\right)$ [8,74]. The prosumer adjusts an operating deadline $ofl_{j,i} \in \mathbb{R}_{\geq 0}$ defining for how long the prosumer can wait to have the appliance's operation completed after its normal end time $\beta_{j,i} \in \mathbb{R}_{\geq 0}$. This flexibility depends on the *desired* start time $ost_{j,i} \in \mathbb{R}_{\geq 0}$ and the appliance's load profile $lp_{j,i}$ (see Section 3.2.1.1), which together adjust the appliance's normal end time. Fig. 4.3 illustrates how the concept of flexibility reshapes the load profile of an appliance. Brown ovals, which their length depends on $\Delta\tau$, symbolize the load demands. Thus, any appliance schedule that lies outside the desired start time and the assigned flexibility deadline is considered invalid.

The concept of flexibility varies according to the appliance type in charge. For appliances, which their physical operations can be interrupted over time (e.g., washing machine and EV), the flexibility is considered as a *deadline* (see Equation (4.7)). This concept from the perspective of the HVAC is the *comfortable temperature range* [75, 76]. $\hbar_i(t), \widetilde{\hbar}_i, tsp_i \in \mathbb{R}$ (°C) are the household temperature at time interval t, the prosumer's desired flexible temperature range, and the temperature set point, respectively. Fig. 4.4 shows the temperature fluctuation of a sample household with and without providing any flexibility. Note that this dissertation for the sake of simplicity only considers the heating part of the HVAC. When the temperature at time interval $t-1$ equals to the maximum allowable temperature, i.e., $\widetilde{\hbar}_i + tsp_i$, the HVAC is turned OFF and the temperature will gradually decrease by the fluctuation degree of $\Delta\hbar_i \in \mathbb{R}_{>0}$ (°C). It is turned ON whenever $\hbar_i(t-1) = \widetilde{\hbar}_i - tsp_i$. Otherwise, the temperature is within the preset comfort range and two possibilities of either turning it ON or OFF exist (see Equation (4.8)). The temperature fluctuation of household of each prosumer at each time interval depends on the capacity of the HVAC unit, and heat gains or losses of the household [77].

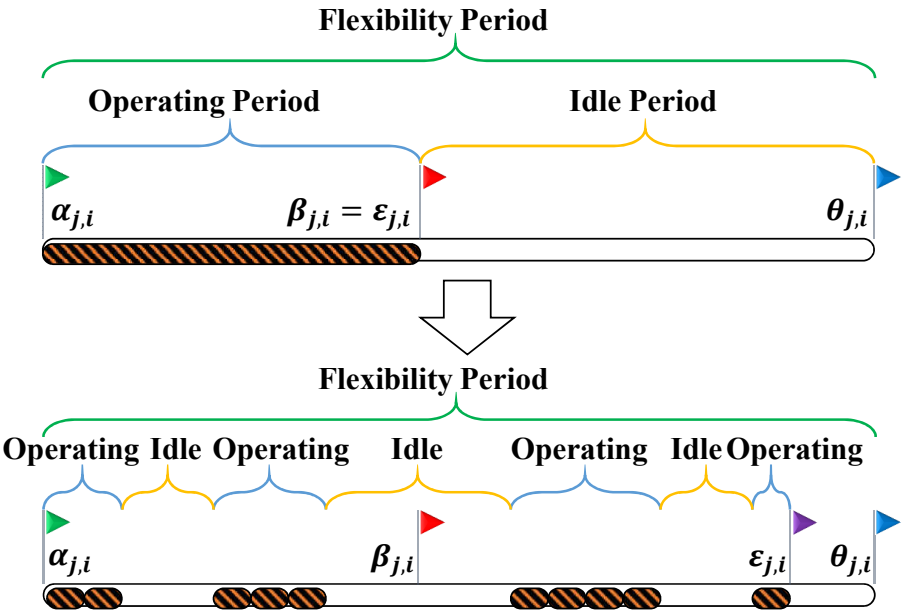


Fig. 4.3: Reshaping the operating cycle of a shiftable appliance $sa_{j,i}$ through the concept of flexibility. $\varepsilon_{j,i} \in \mathbb{R}_{\geq 0}$ is its exact end time after scheduling [8, 9].

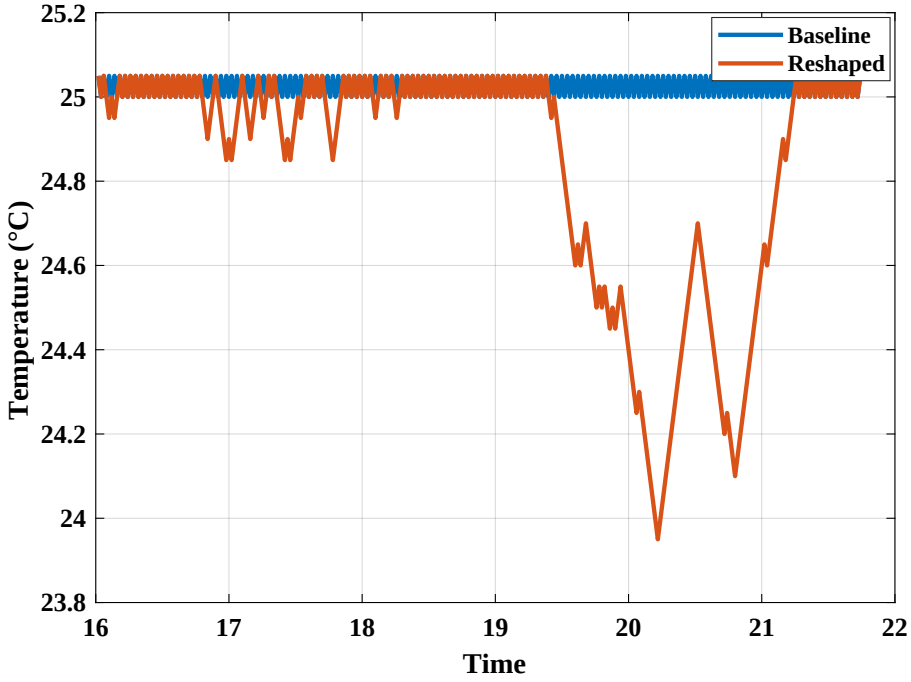


Fig. 4.4: Temperature fluctuation of a sample household with (reshaped) and without (baseline) using the proposed framework. The prosumer sets temperature set point $tsp = 25$ °C and the flexibility $\widetilde{\hbar} = \pm 3$ °C. The temperature fluctuation and the time resolution are $\Delta\hbar = 0.05$ °C and $\Delta\tau = 1$ min, respectively.

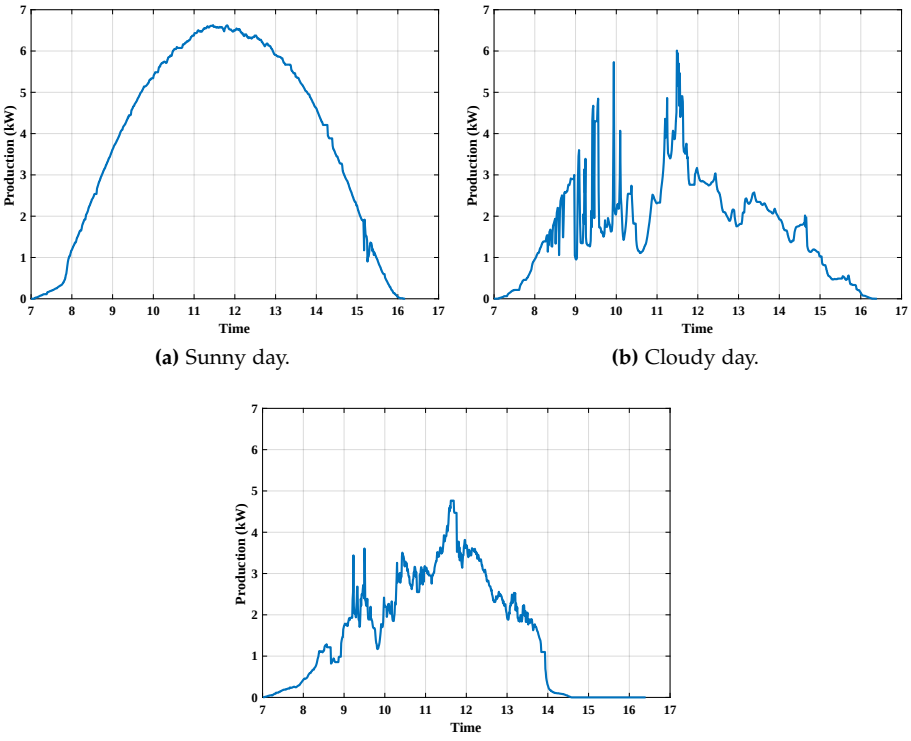
4.1.2 Photovoltaic

Each prosumer is equipped with a locally installed PV system (behind the meter). Let

$$PV_i^G(t) = \aleph_i^{\text{P2L}}(t) + \aleph_i^{\text{P2B}}(t) + \aleph_i^{\text{P2G}}(t) \leq PV_i^{cap}, \tag{4.9}$$

where $PV_i^G(t), PV_i^{cap} \in \mathbb{R}_{\geq 0}$ (kW) are the amount of power that the PV generates at time interval t and its maximum generating capacity, respectively. $\aleph_i^{\text{P2B}}(t), \aleph_i^{\text{P2G}}(t) \in \mathbb{R}_{\geq 0}$ (kW) are the amounts of power transferred from the PV into the BESS and the grid, respectively [9,55]. PVs are RESs with production relying on external factors, such as weather condition [52]. Fig. 4.5 shows how different weather conditions influence the PV production. Obviously, the time when renewable energy is harvested and the time of prosumers' power consumption do not necessarily overlap. As an effect, a mismatch occurs between the local power generation and consumption, which reduces the local RES utilization [9]. Next part describes how BESSs can alleviate such challenge by storing the energy during off-peak and utilize it during

peak periods.



(a) Sunny day.

(b) Cloudy day.

(c) Rainy day.

Fig. 4.5: PV production in different weather conditions. The data belong to the PVs installed at UCLA Ackerman Union [10].

4.1.3 Battery Energy Storage System

Each prosumer owns a BESS. It accumulates excess energy created by the local PV and stores it to be used when there is insufficient amount of energy to supply the demands. Let

$$B_i^{\circledcirc}(t+1) = B_i^{\circledcirc}(t) + \left(B_i^{chg}(t) \times v_i^{chg}(t) - B_i^{d\text{-}chg}(t) \times v_i^{d\text{-}chg}(t)\right) \times \Delta\tau, \quad (4.10)$$

$$B_i^{chg}(t) = \aleph_i^{\mathrm{P2B}}(t) + \aleph_i^{\mathrm{G2B}}(t) \leq \overline{B_i^{chg}}, \quad (4.11)$$

$$B_i^{d\text{-}chg}(t) = \aleph_i^{\mathrm{B2G}}(t) + \aleph_i^{\mathrm{B2L}}(t) \leq \overline{B_i^{d\text{-}chg}}, \quad (4.12)$$

$$v_i^{chg}(t) + v_i^{d\text{-}chg}(t) \leq 1, \quad (4.13)$$

$$B_i^{\circledast}(0) = B_i^{cap} \times \frac{B_i^{\underline{\text{SOC}}} + B_i^{\overline{\text{SOC}}}}{2}, \tag{4.14}$$

$$B_i^{\underline{\text{SOC}}} \leq B_i^{\text{SOC}}(t) \leq B_i^{\overline{\text{SOC}}}, \tag{4.15}$$

$$B_i^{\text{SOC}}(t) = \frac{B_i^{\circledast}(t)}{B_i^{cap}}, \tag{4.16}$$

where $B_i^{\circledast}(t), B_i^{cap} \in \mathbb{R}_{>0}$ (kWh) are the amount of energy stored in the BESS until time interval t and the BESS capacity, respectively. Notations $B_i^{chg}(t), B_i^{d\text{-}chg}(t) \in \mathbb{R}_{\geq 0}$ (kW) denote the amounts of power the battery is "charged" and "discharged" with, respectively, subject to $B_i^{\overline{chg}}, B_i^{\overline{d\text{-}chg}} \in \mathbb{R}_{>0}$ (kW) as maximum charging and discharging power, respectively. Notations $\aleph_i^{\text{B2G}}(t), \aleph_i^{\text{G2B}}(t) \in \mathbb{R}_{\geq 0}$ (kW) denote the amounts of power transferred from the BESS to the grid and vice versa, respectively. $v_i^{chg}(t), v_i^{d\text{-}chg}(t) \in \mathbb{B}$ are binary charging and discharging variables, respectively. Concurrent charging and discharging are not allowed. The BESS at each time interval can charge, discharge, or remain silent (see Equation (4.13)). $B_i^{\circledast}(0)$ is the initial available amount of energy. $B_i^{\underline{\text{SOC}}}, B_i^{\overline{\text{SOC}}}, B_i^{\text{SOC}}(t) \in [0,1]$ are the lowest and highest possible State of Charges (SOCs) of the BESS, and its value at time interval t, respectively. Charging and discharging efficiencies, for clarity of presentation, are assumed to be one [78].

4.1.4 Energy Management System

Fig. 4.6 shows the system model and connections of the EMS of a prosumer. The prosumer interacts with its EMS through a Graphical User Interface (GUI). This GUI provides the prosumer with the possibility of adjusting the operation of smart appliances, controling the flexibilities, and setting the scheduling strategies. According to Section 3.2.1.2, the EMS proposed in this dissertation is a software application running on a gateway (connected to the Internet). EMSs through the Internet are provided with electricity prices and negotiate with the grid about their demands and surplus energy (see Chapter 5). Physical equipment are connected to the gateway through either wired or wireless connections. The prosumer is able to adjust the ECT policy according to the electricity prices captured from the market or the flexibility zone materialized through heterogeneous flexibilities provided. Moreover, the prosumer based on its own preferences and requirements choose a LDS mechanism to apply on load demands of appliances over time. These concepts form a MO-MINLP optimization model, which is solved using the NSGA-III algorithm. Actuating smart appliances or forcing them to wait is done through the load demand buffering sub-system (will be described in Section 4.1.4.3). This sub-system handles events and responses of smart appliances sent and received during their operation over time. Next parts

elaborate the content of the EMS more precisely.

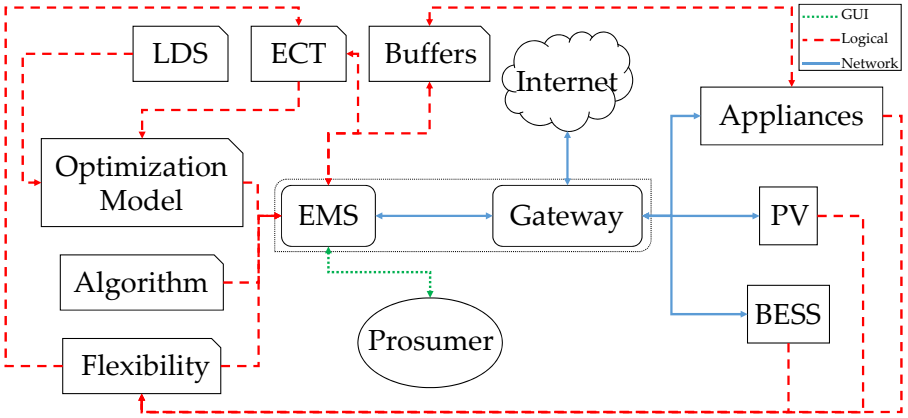


Fig. 4.6: System model and connections of the EMS of a prosumer. Power connections between appliances, the PV, and the BESS are shown in Fig. 4.2.

4.1.4.1 Prosumer Flexibility Zone

Fig. 4.7 illustrates the feasible *flexibility zone* and *flexibility point* for a prosumer. As described in Section 4.1, the following flexibility types are defined:

- **Consumption**: This type belongs to shiftable appliances (except HVAC), in which their operating cycle, due to the provided operating flexibility time, can continuously/discretely be interrupted.
- **Temperature**: This type corresponds to the HVAC, where it provides the EMS with a temperature flexibility range $\left[\widetilde{\hbar}_i - tsp_i, \widetilde{\hbar}_i + tsp_i\right]$.
- **Generation**: This type is related to the PV generation capacity.
- **Storing**: This type is associated with the desired lowest and highest SOCs of the BESS.

Note that consumption and generation flexibilities have similar characteristics (in terms of the power unit). Let

$$\underline{\aleph}_i(t) = PV_i^G(t) - \sum_{j=1}^{\mathfrak{M}_i} ec_{j,i}(t) - \overline{B_i^{chg}}, \tag{4.17}$$

$$\overline{\aleph}_i(t) = PV_i^G(t) + \overline{B_i^{d\text{-}chg}} - \sum_{\substack{flex_{j,i}(t)=0,\\ \forall sa_{j,i} \in SA_i}} ec_{j,i}(t), \tag{4.18}$$

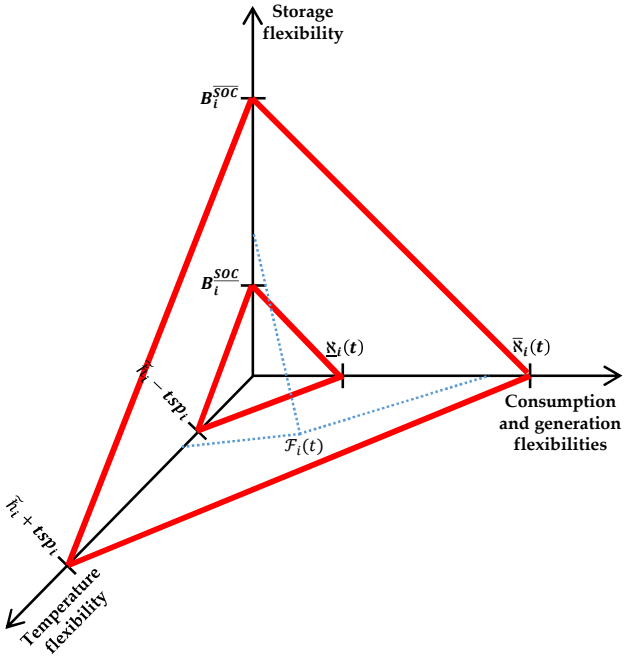


Fig. 4.7: The feasible flexibility zone of a prosumer. Point $\mathcal{F}_i(t)$ reflects a feasible flexibility triple.

where $\underline{\aleph}_i(t), \overline{\aleph}_i(t)$ are the optimum values of maximizing the comfort level and profit, respectively (will be described in Section 4.1.4.4). The electric power generated by the PV and the energy stored in the BESS: i) in the former, satisfy all demanding appliances, and ii) in the latter, supply only appliances with no flexibility. The remaining is always sold to the grid. As a result, let

$$\Im_i(t) \triangleq \left\{ \left\{ \underline{\aleph}_i(t), B_i^{\text{SOC}}, \widetilde{\hbar}_i - tsp_i \right\}, \ldots, \left\{ \overline{\aleph}_i(t), B_i^{\overline{\text{SOC}}}, \widetilde{\hbar}_i + tsp_i \right\} \right\}, \tag{4.19}$$

$$\mathcal{F}_i(t) \triangleq \left\{ \aleph_i(t), B_i^{\text{SOC}}(t), \hbar_i(t) \right\} \subseteq \Im_i(t), \tag{4.20}$$

$$\underline{\aleph}_i(t) \leq \aleph_i(t) = \aleph_i^{\text{B2G}}(t) + \aleph_i^{\text{P2G}}(t) - \aleph_i^{\text{G2B}}(t) - \aleph_i^{\text{G2L}}(t) \leq \overline{\aleph}_i(t), \tag{4.21}$$

where $\Im_i(t), \mathcal{F}_i(t)$ is the *feasible* flexibility zone and an arbitrary flexibility point (triple) of prosumer ρ_i at time interval t. $\aleph_i(t) \in \mathbb{R}$ (kW) is the *desired* amount of power the prosumer strives to exchange with the grid.

4.1.4.2 Electricity Consumption Threshold Policies

Many electricity producers are experiencing a deficit of electricity generation capacity in consequence of load demands by prosumers. More accurately, the

aggregated supply (either generated by electric utilities or surplus energy injected by prosumers) cannot meet the aggregated demands of prosumers at a specific time interval. SALSA defines the concept of ECT as a two-level boundary. The first level, as described in Section 4.1.1, includes soft ECTs belonging to prosumers. The second level includes *hard* ECTs, which are adjusted by the aggregator (will be described in Section 4.2.1.1). This two-level concept complies with the smart grid operating regimes as defined by USEF and is shown in Fig. 4.8 [11]. As the electricity demand increases the grid operation moves gradually from the "Normal Operations" (first level) scheme to the "Capacity Management" (second level) regime. At the same time, the regulating electricity markets start planning a significant role by providing peak load reduction and power balancing between electricity supply and demand through market mechanisms. Should the aggregator not be able to successfully manage capacity, there is a risk of hitting the second demarcation point, in which the regime of "Graceful Degradation" starts. These demarcation points are marked by the ECT in Fig. 4.8. When crossing this boundary the aggregator may take action to curtail load regarding any Service Level Agreement (SLA) promises just to secure the grid supply and to regain control of the network (curtailment is outside of the scope of this dissertation).

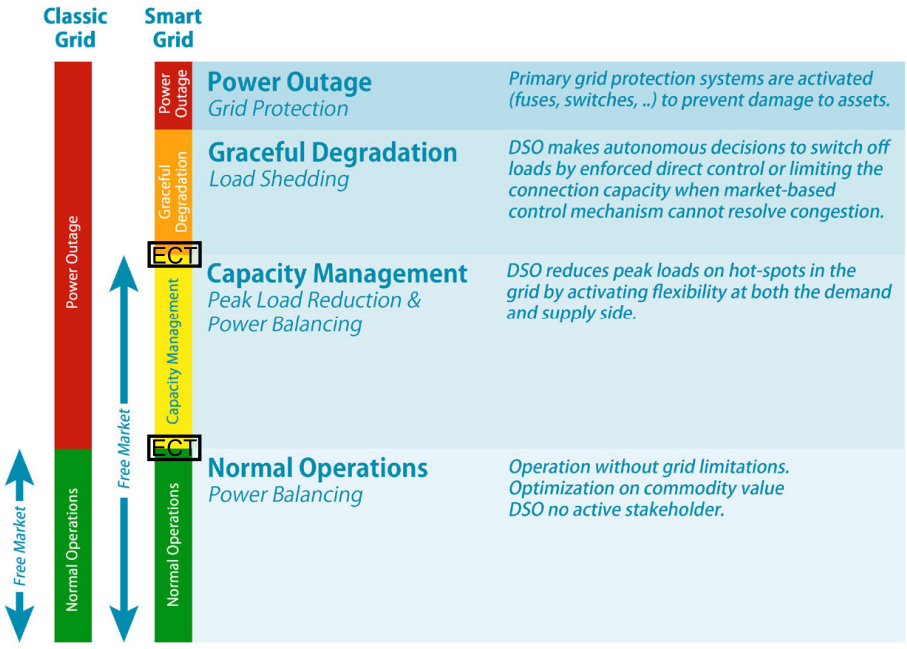


Fig. 4.8: Adoption of the ECT concept to the USEF operating regimes [11].

The following defines three generic ECT policies for prosumers and the aggregator. Table 4.1 compares these policies with respect to relevant pa-

rameters in the framework. Each policy is followed by a figure displaying a conceptual example of applying the corresponding policy on load demands of a prosumer. Similar examples for the aggregator will be provided in Chapter 6.

Table 4.1: Comparison of different ECT policies with respect to relevant parameters in the framework. Last two parameters are adopted from the smart grid operating regimes defined by the USEF.

	ECT-P1	ECT-P2	ECT-P3
Day-ahead	✓	–	✓
Real-time	–	✓	–
Price-driven	–	–	✓
Normal operations	✓	✓	✓
Capacity management	✓	✓	✓

4.1.4.2.1 ECT-P1

This policy aims at keeping the aggregated load demands below a constant value at all time intervals. Let

$$ect_i^{\mathscr{S}}(t) = \overbrace{\left(\max_{t \leq T} \sum_{j=1}^{\mathfrak{M}_i} ec_{j,i}(t) \times dec_{j,i}(t)\right)}^{\text{Previous day}} \times \mathcal{X}_i(t), \tag{4.22}$$

$$\mathcal{X}_i(1) = \mathcal{X}_i(2) = \ldots, \mathcal{X}_i(t) = \ldots, \tag{4.23}$$

where $\mathcal{X}_i(t) \in (0, 1]$ is a fraction value for the ECT of prosumer ρ_i at each time interval t. This policy follows a day-ahead approach, since ECTs are a percentage of the previous day's peak demand [1]. Fig. 4.9 displays a conceptual example of applying this policy on load demands of a prosumer.

4.1.4.2.2 ECT-P2

This policy, in contrast to ECT-P1, adjusts the ECT at each time interval with the multiplication of the aggregation of load demands at that time and a fixed constant percentage. Let

$$ect_i^{\mathscr{S}}(t) = \left(\sum_{j=1}^{\mathfrak{M}_i} ec_{j,i}(t)\right) \times \mathcal{X}_i(t). \tag{4.24}$$

Fig. 4.10 illustrates an example of applying this policy on load demands of the random prosumer (assumed the same prosumer selected for Fig. 4.9).

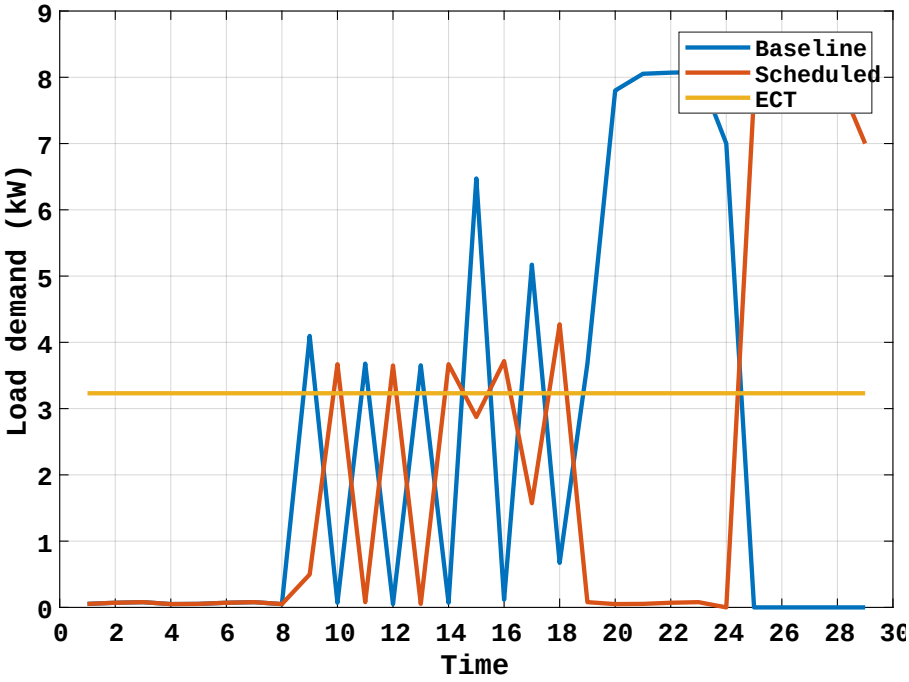


Fig. 4.9: A schematic view of applying ECT-P1 on load demands of a random prosumer.

4.1.4.2.3 ECT-P3

This policy follows the day-ahead normalized electricity prices over time. Let

$$ect_i^{\mathscr{S}}(t) = \left(\sum_{j=1}^{\mathfrak{M}_i} ec_{j,i}(t)\right) \times \left(1 - \frac{P_{\mathcal{G}}(t) - \min\limits_{t \leq T} P_{\mathcal{G}}(t)}{\max\limits_{t \leq T} P_{\mathcal{G}}(t) - \min\limits_{t \leq T} P_{\mathcal{G}}(t)}\right), \tag{4.25}$$

where $P_{\mathcal{G}}(t) \in \mathbb{R}_{>0}$ (\$/kWh) is the day-ahead price that the grid announced at time time interval t (assuming that no trading was happening between prosumers and the grid). This policy follows the multiplication of the aggregation of load demands and the normalized electricity price [8]. Fig. 4.11 shows how this policy influences the scheduling of a random prosumer's consumption over time (assumed the same prosumer selected for Fig. 4.9).

4.1.4.3 Load Demand Buffering Sub-System

This subsystem is composed of four various buffers named Immediately Wait (IW), Immediately Start (IS), Decided to Operate (DO), and Decided to Wait (DW). First, the sub-system stores all incoming load demands into the IW buffer. The IS buffer is specialized for load demands of non-shiftable

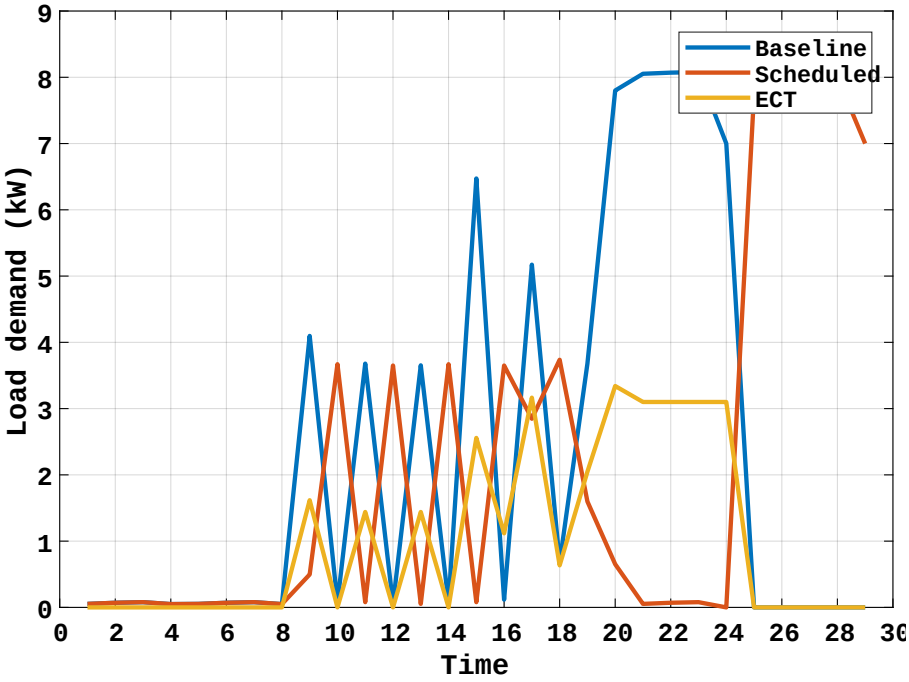


Fig. 4.10: A schematic view of applying ECT-P2 on load demands of a random prosumer.

appliances while the last two buffers are designed for shiftable appliances. Then, a load demand is moved to the DO buffer if the EMS decides to operate the corresponding appliance at that time interval. Otherwise, the load demand will be moved to the DW buffer. At the next time interval, the buffering sub-system removes the load requests from the DW buffer and appends them to the IW buffer. Indeed, the EMS will decide about appended and newly arrived load demands simultaneously. At each time interval, the EMS sends schedules to the gateway based on load demands located in the DO and the DW buffers. Fig. 4.12 shows the flow process of handling a load demand by the buffering sub-system.

4.1.4.4 Optimization Model

Multi-Objective Optimization (MOO) is an area of multiple criteria decision-making, where mathematical optimization problems involving more than one objective function are optimized simultaneously [79]. Optimal decisions need to be taken in the presence of trade-offs between such conflicting objectives. When decision making is emphasized, the purpose of solving a MOO problem is referred to support a *decision maker* in finding the most preferred *non-dominated* solutions. The decision maker, in this work, is the EMS of each

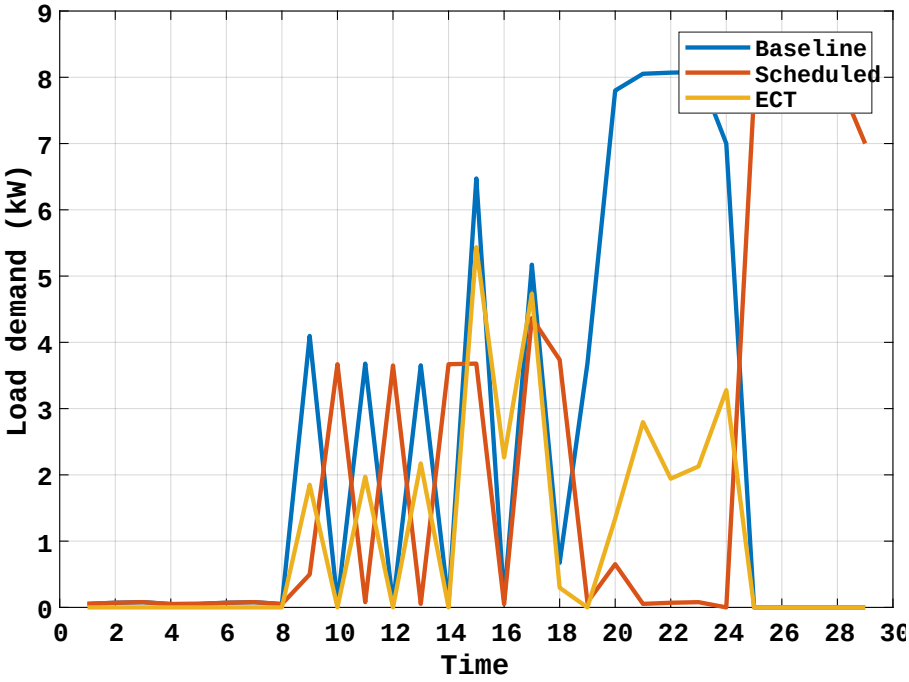


Fig. 4.11: A schematic view of applying ECT-P3 on load demands of a random prosumer.

prosumer and the OMS of the aggregator (will be described in Section 4.2). The objective functions are said to be conflicting, whenever there exists an infinite number of non-dominated solutions. A solution does not improve for one objective unless it satisfies others. The main goal in MOO problems is to find a finite number of diverse solutions in the objective space.

The following defines a MOO model for each prosumer, to schedule its appliances and trade its surplus energy. Let

$$\underset{\aleph_i(t),P_i(t)}{\text{maximize}}\ \aleph_i\left(t\right)\times P_i\left(t\right)\times\Delta\tau, \tag{4.26}$$

$$\underset{\left\{dec_{j,i}(t)\right\}_{j=1}^{\mathfrak{M}_i},B_i^{chg}(t)}{\text{maximize}}\ \sum_{j=1}^{\mathfrak{M}_i} ec_{j,i}\left(t\right)\times dec_{j,i}\left(t\right)+B_i^{chg}\left(t\right), \tag{4.27}$$

subject to

$$\begin{aligned}&(4.1)-(4.25),\\ &P^{\mathbb{l}}\left(t\right)\leq P_i\left(t\right)\leq P^{\mathfrak{u}}\left(t\right),\end{aligned} \tag{4.28}$$

where Equation (4.26) intends to maximize the profit by increasing the selling amount while Equation (4.27), by satisfying as many load demands as

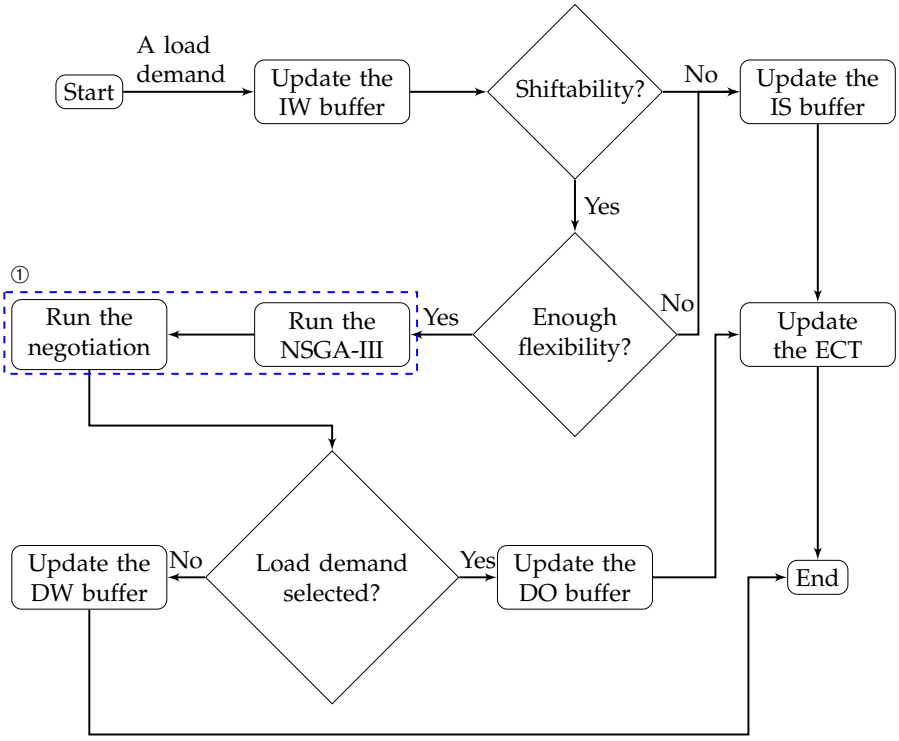


Fig. 4.12: Flow process of handling a load demand by the buffering sub-system. ① The load demand is appended to a *temp buffer*, when it has enough flexibility. Functions inside the dashed blue box are applied on all load demands stored in the temp buffer. These functions are described in Section 4.1.4.6 and Chapter 5.

possible and charging the BESS as much as possible, aims at maximizing the comfort level. These two objectives are in conflict with each other, since trying to inject more power to the grid results in jeopardizing the comfort level. $P_i(t) \in \mathbb{R}_{>0}$ (\$/kWh) is a price offer selected between $\left[P^{\mathbb{l}}(t), P^{\mathbb{u}}(t)\right] \in \mathbb{R}_{>0}$ (\$/kWh) as the minimum and maximum offerable price for trading energy, respectively. For each prosumer at each time interval, one of the following cases can happen:

- The prosumer is a buyer, i.e., $\underline{\aleph}_i(t) \leq \overline{\aleph}_i(t) < 0$: The prosumer, through the electric power generated by the PV, the energy stored in the BESS, and the amount intended to purchase externally, fully satisfies appliances with zero flexibility. The profit, however, is always below zero. To lower the cost, the prosumer strives to pay less by offering a buying price close to $P^{\mathbb{l}}(t)$.

- The prosumer is a seller, i.e., $0 < \underline{\aleph}_i(t) \leq \overline{\aleph}_i(t)$: Appliances with zero flexibility, by the PV generation and energy stored in the BESS, are fully

satisfied and the surplus power is injected to the grid. The prosumer, to make the most beneficial contract, attempts to offer a selling price close to $P^{\mathfrak{u}}(t)$.

- The prosumer is flexible, i.e., $\underline{\aleph}_i(t) < 0$ and $\overline{\aleph}_i(t) > 0$: Both of the previous cases can happen, which makes a trade-off between the comfort level and profit. In this case, the prosumer can also be silent, i.e., $\aleph_i(t) = 0$, where $\aleph_i^{\text{G2B}}(t) = \aleph_i^{\text{G2L}}(t) = \aleph_i^{\text{P2G}}(t) = \aleph_i^{\text{B2G}}(t) = 0$. Power actions $\aleph_i^{\text{B2L}}(t), \aleph_i^{\text{P2L}}(t), \aleph_i^{\text{P2B}}(t)$ are tuned so as to maximize the comfort level, where the profit is zero.

Prosumers are not allowed to buy and sell at the same time. If the prosumer intends to buy electric power from the grid, i.e., $\aleph_i^{\text{G2B}}(t) > 0$ or $\aleph_i^{\text{G2L}}(t) > 0$, then, $\aleph_i^{\text{B2G}}(t) = 0$ and $\aleph_i^{\text{P2G}}(t) = 0$. If the prosumer is a seller, i.e., $\aleph_i^{\text{B2G}}(t) > 0$ or $\aleph_i^{\text{P2G}}(t) > 0$, then, $\aleph_i^{\text{G2B}}(t) = 0$ and $\aleph_i^{\text{G2L}}(t) = 0$. Prosumers, besides fully satisfying appliances with zero flexibility, have the option to operate the remaining appliances. Synthesizing various LDS mechanisms with the optimization model affects the prosumers' behaviors in terms of power actions over time [5, 9]. Such mechanisms are described in the following part.

4.1.4.5 Load Demand Scheduling Mechanisms

To produce a specific schedule for appliances of each prosumer at each time interval based on the aforementioned objective and constraints, EMSs continuously apply a LDS mechanism on load demands stored in their IW buffer [6]. Choosing a LDS mechanism depends on the prosumer's preferences, which will obviously, yield to different results. Apart from the ECT imposed, EMSs permit: i) non-shiftable appliances, and ii) shiftable appliance without enough flexibility [5]. Then, the LDS mechanism will schedule the remaining load demands and forward the decisions back to EMSs. These decisions will be used to update the load demand buffering sub-system. The following proposes various mechanisms used in the framework.

4.1.4.5.1 0-1 Knapsack Problem

The 0-1 Knapsack problem is a traditional combinatorial optimization problem in computer science, where given $M \in \mathbb{N}$ items (each with a weight and value), it tries to obtain the maximum total value out of the items being packed subject to knapsack's limited capacity $c \in \mathbb{R}_{>0}$. This problem is NP-Complete since the time complexity of solving it through the brute-force method is $O(2^M)$ [80]. This method, in order to find the optimal solution calculates, all feasible subsets. The load demand scheduling problem is NP-complete by a reduction from the 0-1 Knapsack problem, since items are equivalent to load demands, where their demanding power and cost reflect

the weight and value, respectively [5, 29, 30]. The Knapsack procedure receives current load demands and calculates the fitness of produced feasible subsets, where each subset comprises some load demands. The outcome of this approach is a subset of load demands, which their corresponding appliances are either allowed to continue operating or forced to wait. In order to decrease the computation time, a Dynamic Programming approach with the time complexity of $O(M \times c)$ is applied. For more information about this approach, the reader is referred to [1, 4].

4.1.4.5.2 Earliest Deadline First

The Earliest Deadline First (EDF) mechanism is an optimal scheduling algorithm and derived from real-time systems theory, which is particularly heuristic but presenting satisfactory scalability properties [8, 81]. It is used in real-time operating systems to place preemptive uniprocessors in a priority queue. Whenever a scheduling event occurs (here load demands) the queue will be searched for the event closest to its deadline. Hence, this mechanism starts from a load demand with lowest flexibility $ofl_{j,i}, \forall sa_{j,i} \in \mathcal{S}_i$ calculated at each time interval and attempts to satisfy as many load demands as possible subject to the current ECT.

4.1.4.5.3 Least Slack Time

The Least Slack Time (LST) scheduling mechanism is particularly practicable in applications comprising aperiodic tasks (such as load demands), since no prior assumptions on their rate of occurrence are made [82]. This mechanism, by calculating $ofl_{j,i} - t - lp_{j,i}, \forall sa_{j,i} \in \mathcal{S}_i$ at each time interval t, ascendingly sorts potential load demands. Note that $lp_{j,i}$ in this mechanism denote the remaining load profile from time interval t.

4.1.4.5.4 Latest Release Time

In contrast to the EDF mechanism, the Latest Release Time (LRT) mechanism is a time-reversed scheduling algorithm. It assigns the highest priority to a load demand with the latest flexibility time. The same optimality claim holds for the LRT mechanism. For more information about the proof, the reader is referred to [83].

4.1.4.5.5 Rate-Monotonic Scheduling

The Rate-Monotonic Scheduling (RMS) mechanism at each interval assigns priorities to the rate monotonic conventions. In other words, load demands with shorter remaining load profiles are given higher priorities [81, 84].

4.1.4.5.6 First In First Out

The First In First Out (FIFO) is one of the conventional scheduling mechanisms used in the theory of operating systems. This mechanism for each prosumer sorts its load demands at each time interval according to their operating start times $ost_{j,i}, \forall sa_{j,i} \in \mathcal{S}_i$ [85].

Comparison: Fig. 4.13 pictures a schematic view of applying various LDS mechanisms on a set of sample load demands during three consecutive time intervals. Table 4.2 compares these LDS mechanism with respect to relevant parameters in the framework. Such parameters make a trade-off among LDS mechanism. Choosing a proper mechanism depends on each prosumer's preferences toward operating its appliances and the aggregator's requirements at each time interval.

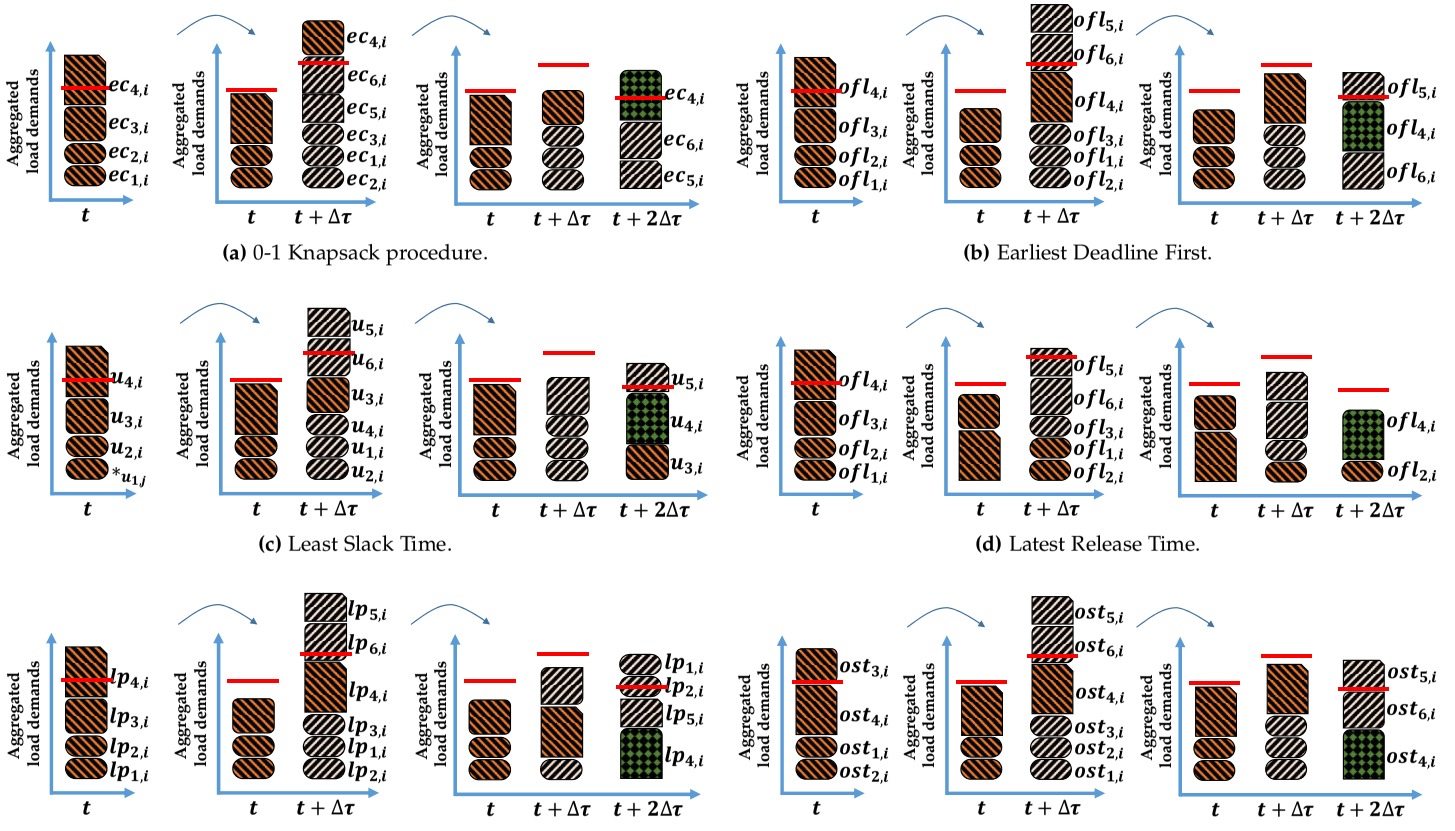
Table 4.2: Comparison of different LDS mechanisms with respect to relevant parameters in the framework.

	Knapsack	EDF	LST	LRT	RMS	FIFO
Current time interval	✓	✓	✓	✓	–	✓
Load demand	✓	–	–	–	–	✓
Appliance load profile	–	–	✓	–	✓	–
Operating start time	–	–	–	–	–	✓
Operating flexibility	–	✓	✓	✓	–	–
Scalable	*–	✓	✓	✓	✓	✓
Priority-based	–	✓	✓	✓	✓	✓
ECT-driven	✓	✓	✓	✓	✓	✓
Price-driven	✓	–	–	–	–	–

* Using evolutionary algorithms can be an option to schedule a scalable number of load demands with the 0-1 Knapsack mechanism. However, this can be a computationally expensive way. More discussion will follow in Chapter 6.

4.1.4.6 Optimization Algorithm

Algorithm 4.1 describes how the optimization model, presented in Section 4.1.4.4, is approached. This algorithm consists of three parts: i) load demand buffering sub-system for handling incoming load demands and actuating the appliances according to decisions made, ii) the NSGA-III algorithm to adjust power actions and schedule the appliances, and iii) communication with the aggregator, through a VPP, to satisfy remaining demands/sell surplus energy through the grid. The VPP, its definition, and responsibilities, along with relevant notations are described in Chapter 5. In Algorithm 4.1, first, all load demands stored in the IW buffer are controlled with respect to their shiftability feature. Non-shiftable appliances and those, which hold insufficient flexibility, are allowed to operate. Then, the ECT is updated accordingly.



(a) 0-1 Knapsack procedure.

(b) Earliest Deadline First.

(c) Least Slack Time.

(d) Latest Release Time.

(e) Rate-Monotonic Scheduling.

(f) First In First Out.

Fig. 4.13: A schematic view of applying various LDS mechanisms on a set of sample load demands during three consecutive time intervals. In all figures, relevant variables at each time interval are sorted ascendingly (bottom to top). See Table 4.2 for their comparison.

Next, the NSGA-III generates a set of feasible solutions and forwards them to the VPP. The VPP, after negotiating with the aggregator, returns the indexes of load demands, which it has agreed with the aggregator on, to be allowed to operate. The DO, IW, and ECT are updated accordingly. Next part describes the NSGA-III algorithm and its mechanism in producing the solutions.

Algorithm 4.1: The main optimization algorithm run by the EMS of each prosumer $\rho_i \in \mathcal{P}$ at each time interval t [4,5,8].

Register each incoming load demand in the IW buffer (see Fig. 4.12);
while $IW \neq \{\}$ **do**
  foreach $ev_{j,i} \in IW$ **do**
    if $ev_{j,i}.shift = 0 \vee flex_{j,i} = 0$ **then**
      $\text{IS} = \text{IS} \cup ev_{j,i}$;
      $\text{IW} = \text{IW} \setminus \{ev_{j,i}\}$;
      $ect_i^{\mathscr{S}}(t) = ect_i^{\mathscr{S}}(t) - ev_{j,i}.ec$;
    end
  end
  Run the NSGA-III to produce $\ell_i(t)$;
  Determine the reservation behavior pair $\mathcal{BP}_i^{res}(t)$;
  Send $\ell_i(t)$ and $\mathcal{BP}_i^{res}(t)$ to the VPP (see Algorithm 5.2);
  $ev_{\{1,2,\ldots\},i} \leftarrow$Load demands agreed to be allowed at this time interval;
  $\text{DO} = \text{DO} \cup ev_{\{1,2,\ldots\},i}$;
  $\text{IW} = \text{IW} \setminus \left\{ev_{\{1,2,\ldots\},i}\right\}$;
  $ect_i^{\mathscr{S}}(t) = ect_i^{\mathscr{S}}(t) - \sum ev_{\{1,2,\ldots\},i}.ec$;
end

4.1.4.6.1 The Overall Process of the NSGA-III

Multi-Objective Evolutionary Algorithms (MOEAs), due to their independent search space structure, are among the most well-known meta-heuristic search mechanisms utilized for the MOO problems [86]. MOEAs form a subset of evolutionary computations, in which they generally involve techniques and implementing mechanisms inspired by biological evolutions, such as reproduction, mutation, recombination, natural selection, and survival of the fittest. The main advantage of these algorithms, when applied to solve MOO problems, is the fact that they typically generate sets of solutions, allowing computation of the entire Pareto-front. This work, by employing the evolutionary NSGA-III, generates a finite number of non-dominated solutions to optimization problems. This algorithm, to guarantee the diver-

sity among such solutions, uses a *reference-point-based non-dominated sorting approach*. These points are all permutations of extreme values of power (i.e., $\min_{\forall \rho_i \in \mathcal{P}} \underline{\aleph}_i(t), \max_{\forall \rho_i \in \mathcal{P}} \overline{\aleph}_i(t), \forall t \leq T$) and price (i.e., $P^{\mathtt{l}}(t), P^{\mathtt{u}}(t), \forall t \leq T$) located on a normalized hyper-plane. Note that each of prosumers and the aggregator (will be described in Section 4.2) at each time interval run the NSGA-III independently. The framework for generating the first Pareto-front is open to use other algorithms.

Algorithm 4.2 explains the procedures of the NSGA-III. The algorithm starts by generating an initial parent population including $\mathcal{Q} \in \mathbb{N}$ feasible solutions. Section 4.1.4.6.2 for prosumers and Section 4.2.1.2.1 for the aggregator propose a generic formulation describing the mechanism of producing these solutions. The fitness value of each solution is a pair by the evaluation through the relevant objective functions. It continues until the maximum number of generations $\mathcal{W} \in \mathbb{N}$ is reached. In each generation, the algorithm produces new solutions (offspring) and updates the parent population. The diversity of solutions in a population is an important factor in reaching a near-optimal Pareto-front. To reach such diversity, the NSGA-III uses the tournament selection, in which it disregards low and keeps solutions with high fitness value found by then (elitism). To make (possibly better) new offspring solutions, the algorithm benefits from the linear crossover and uniform mutation procedures [4]. Then, it creates reference sets according to the reference points and associates each Pareto-solution a reference value. To create new Pareto-fronts, it starts by determining the closest solutions (in the combined parent and offspring populations) to the reference points using a niche-preservation operation and places them in the fronts accordingly [67].

Although reaching an optimal balance between the tournament, crossover, and mutation procedures is a challenging issue, it can, however, be managed by some proper control parameter settings, such as probabilistic execution [87]. A majority of good solutions survive and the search space remains unexplored when the probability of calling the tournament selection procedure is very high. On the other hand, most of the objective space is explored, but the probability of neglecting a majority of good solutions is relatively high when the probability of calling linear crossover and uniform mutation procedures is high [88]. To also ensure the satisfaction of all constraints described in the optimization models, the NSGA-III in each generation applies a constraint handling (repairing) procedure on infeasible solutions produced [89]. For more information about this algorithm, the reader is referred to [4, 9, 67, 89].

Algorithm 4.2: The procedure of the NSGA-III.

1 Initialize $\mathcal{Q}$ feasible parent solutions (see Section 4.1.4.6.2 for prosumers and Section 4.2.1.2.1 for the aggregator);
2 Evaluate the fitness pair of these $\mathcal{Q}$ parent solutions through the relevant objective functions (see Equations (4.26) and (4.27) for prosumers and Equations (4.32) and (4.33) for the aggregator);
3 **for** $\mathscr{A} = 1$ *to* $\mathcal{W}$ **do**
4 Run the tournament selection to select $\mathscr{U} \in \mathbb{N}$ parent solutions (elitism);
5 Run the linear crossover with the probability of $pc \in (0,1)$ to produce $\mathscr{C} \in \mathbb{N}$ offspring solutions;
6 Run the uniform mutation with the probability of $pm \in (0,1)$ to produce $\mathscr{M} \in \mathbb{N}$ offspring solutions;
7 Run the constraint handling (repairing) procedure on infeasible $\mathscr{C} + \mathscr{M}$ offspring solutions (in the worst case);
8 Evaluate the fitness value of these $\mathscr{C} + \mathscr{M}$ offspring solutions (see Line 2);
9 Create reference sets according to the reference points;
10 Associate each Pareto-solution (in $\mathcal{Q} + \mathscr{U} + \mathscr{C} + \mathscr{M}$ solutions) a reference value;
11 Determine the closest solutions to the reference points using a niche-preservation operation and place them in the fronts accordingly;
12 Replace the parent solutions with the first $\mathcal{Q}$ solutions out of the sorted $\mathcal{Q} + \mathscr{U} + \mathscr{C} + \mathscr{M}$ solutions;
13 **end**
14 Provide a set of feasible desired solutions existing on the first Pareto-front;

4.1.4.6.2 Pareto-Solution Formulation for Prosumers

This section defines a generic formulation for Pareto-solutions of prosumers, which is independent of the algorithm used to produce them. Let

$$\ell_i(t) = \begin{bmatrix} \mathcal{BP}_i^1(t) & \mathcal{SI}_i^1(t) \\ \vdots & \vdots \\ \mathcal{BP}_i^K(t) & \mathcal{SI}_i^K(t) \end{bmatrix}, \tag{4.29}$$

$$\mathcal{BP}_i^k(t) \triangleq \left(\aleph_i^k(t), P_i^k(t)\right), \tag{4.30}$$

$$\mathcal{SI}_i^k(t) \triangleq \begin{cases} \frac{\aleph_i^k(t)}{\overline{\aleph}_i(t)} + \frac{P_i^k(t)}{\overline{P^{\mathrm{u}}}(t)} & \aleph_i^k(t) > 0, \\ \frac{\aleph_i^k(t)}{\underline{\aleph}_i(t)} + \frac{P^{\mathrm{l}}(t)}{P_i^k(t)} & \aleph_i^k(t) < 0, \end{cases} \tag{4.31}$$

where $\ell_i(t)$ at each time interval is produced once and is comprised of feasible *behavior pairs* $\mathcal{BP}_i^k(t), 1 \leq k \leq K$. Fig. 4.14 shows how these behavior pairs, according to Equations (4.1)-(4.25) and (4.28), are randomly generated. $\ell_i(t)$ is sorted descendingly by its second column. *Satisfaction index* $\mathcal{SI}_i^k(t) \in (0,2]$ is a measure of to which extent $\mathcal{BP}_i^k(t)$ optimizes the prosumer's conflicting objectives [9,40,41].

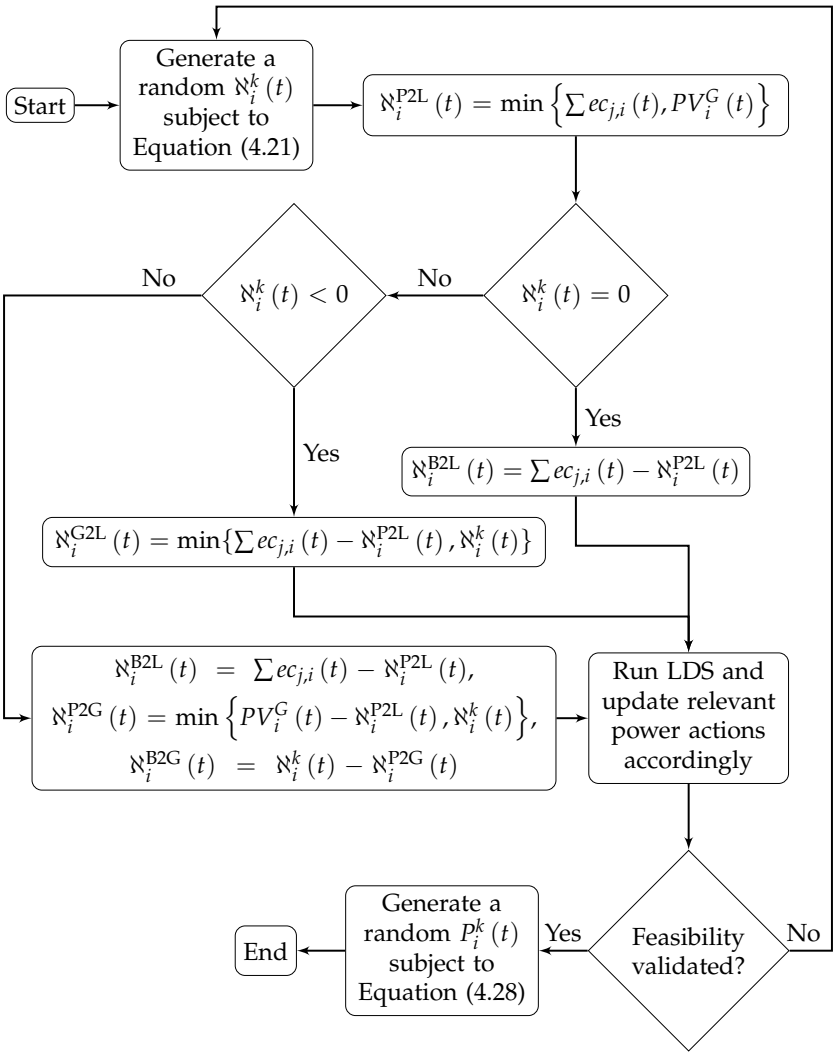


Fig. 4.14: Flow chart of generating a feasible behavior pair $\mathcal{BP}_i^k(t)$. $\sum ec_{j,i}(t)$ is the summation of load demands of appliances with $flex_{j,i}(t) = 0$. Note that the LDS mechanism is applied on appliances with $flex_{j,i}(t) = 1$. Choosing proper power actions to update after running the LDS depends on $\aleph_i^k(t)$ (see Section 4.1.4.4 for different power exchange scenarios) [9].

4.2 Aggregator

The aggregator, which can also act as a balance responsible party [53], holds no physical connection with the grid and is only responsible for, by trading prosumers' flexibilities, making feasible and profitable contracts with them [9].

4.2.1 Operating Management System

Fig. 4.15 shows the system model and connections of the OMS of the aggregator. The aggregator interacts with its OMS through a GUI. This GUI provides the aggregator with the possibility of adjusting the ECT policy and negotiation functions (will be described in Chapter 5). Similar to the EMS of prosumers, the OMS is a software application running on a gateway (connected to the Internet). The OMS through the Internet communicates with the grid operator and a VPP, which is acting on behalf of prosumers (will be described in Chapter 5). The grid operator adjusts its own price offers and indirectly influences the ECT policy. The aggregator's MO-MINLP optimization model, similar to prosumers', is solved through the NSGA-III algorithm. Next parts elaborate the content of the OMS more precisely.

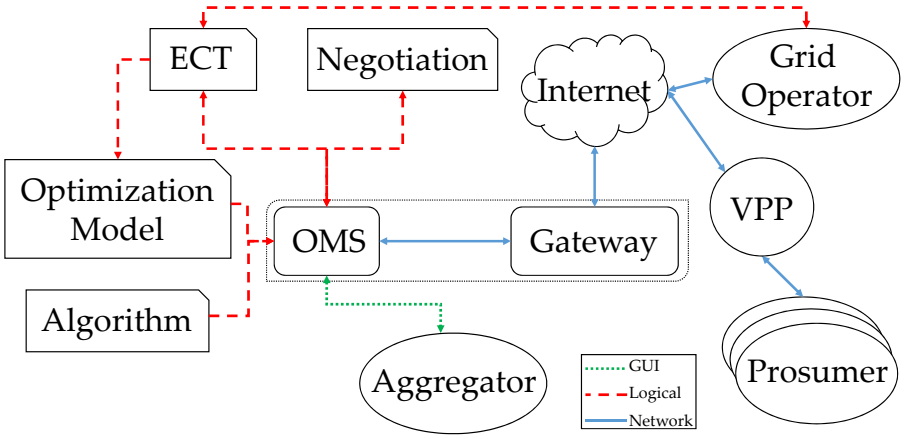


Fig. 4.15: System model and connections of the OMS of the aggregator.

4.2.1.1 Optimization Model

To enable the aggregator to make the decisions in response to prosumers' load demands, it runs an optimization model. Let

$$\underset{\left\{\widetilde{\aleph}_i(t),\widetilde{P}_i(t)\right\}_{i=1}^{h}}{\text{maximize}} \ \Delta\tau \times \sum_{i=1}^{h} \begin{cases} \widetilde{\aleph}_i(t) \times \left(P_{\mathcal{G}}^{\mathbb{1}}(t) - \widetilde{P}_i(t)\right) & \widetilde{\aleph}_i(t) > 0, \\ \widetilde{\aleph}_i(t) \times \left(\widetilde{P}_i(t) - P_{\mathcal{G}}^{\mathbb{u}}(t)\right) & \widetilde{\aleph}_i(t) < 0, \end{cases} \tag{4.32}$$

$$\underset{\left\{\widetilde{\aleph}_i(t)\right\}_{i=1}^{h}}{\text{minimize}} \ -\sum_{i=1}^{h} \widetilde{\aleph}_i(t), \tag{4.33}$$

subject to

$$\begin{cases} \begin{bmatrix} 0 < \widetilde{\aleph}_i(t) \leq \max\limits_{\forall \rho_i \in \mathcal{P}} \overline{\aleph}_i(t) \\ P^{\mathbb{1}}(t) \leq \widetilde{P}_i(t) \leq P_{\mathcal{G}}^{\mathbb{1}}(t) \end{bmatrix} & \aleph_i(t) > 0, \forall \rho_i \in \mathcal{P}, \\ \begin{bmatrix} \min\limits_{\forall \rho_i \in \mathcal{P}} \underline{\aleph}_i(t) \leq \widetilde{\aleph}_i(t) < 0 \\ P_{\mathcal{G}}^{\mathbb{u}}(t) \leq \widetilde{P}_i(t) \leq P^{\mathbb{u}}(t) \end{bmatrix} & \aleph_i(t) < 0, \forall \rho_i \in \mathcal{P}, \end{cases} \tag{4.34}$$

$$\sum_{i=1}^{h} \widetilde{\aleph}_i(t) + \widetilde{\aleph}_{\mathcal{A}}(t) = 0, \tag{4.35}$$

$$\left| \sum_{i=1}^{h} \widetilde{\aleph}_i(t) \right| \leq ect_{\mathcal{A}}^{\mathscr{H}}(t), \tag{4.36}$$

where Equation (4.32), by matching the surplus energy with energy shortage, attempts to maximize its profit while Equation (4.33) aims at minimizing the grid purchase. These objectives are in conflict with each other, since selling more to buyer prosumers and buying less from seller prosumers lead to buying more from the grid. $\widetilde{\aleph}_i(t) \in \mathbb{R}$ (kW) is the amount of power the aggregator trades with prosumer ρ_i coupled with a price offer $\widetilde{P}_i(t) \in \mathbb{R}_{>0}$ (\$/kWh). Note that Equation (4.34) prevents the aggregator from requesting buyer prosumers to sell and vice versa. The aggregator has to exchange $\widetilde{\aleph}_{\mathcal{A}}(t) \in \mathbb{R}$ (kW) amount of electric power at time interval t with the grid since, with respect to Equation (4.35), the supply and demand at each time interval must match in the grid. $P_{\mathcal{G}}^{\mathbb{1}}(t), P_{\mathcal{G}}^{\mathbb{u}}(t) \in \mathbb{R}_{>0}$ (\$/kWh) are grid's prices for buying/selling energy from/to the aggregator, respectively. $ect_{\mathcal{A}}^{\mathscr{H}}(t) \in \mathbb{R}_{>0}$ (kW) is the hard ECT (e.g., a substation's capacity) adjusted by the aggregator in collaboration with the grid operator.

4.2.1.2 Optimization Algorithm

Similar to prosumers, the aggregator, to randomly generate a set of feasible non-dominated solutions to its optimization problem, benefits from the

NSGA-III (see Section 4.1.4.6 for more information about the algorithm).

4.2.1.2.1 Pareto-Solution Formulation for the Aggregator

This dissertation, similar to the generic formulation developed for producing Pareto-solutions of prosumers, proposes an algorithm-independent model for producing Pareto-solutions of the aggregator's optimization model. Let

$$\widetilde{\ell}_{\mathcal{A}}(t) = \begin{bmatrix} \widetilde{\mathcal{BM}}^{1}_{\mathcal{A}}(t) & \mathcal{SI}^{1}_{\mathcal{A}}(t) \\ \vdots & \vdots \\ \widetilde{\mathcal{BM}}^{\widetilde{K}}_{\mathcal{A}}(t) & \mathcal{SI}^{\widetilde{K}}_{\mathcal{A}}(t) \end{bmatrix}, \tag{4.37}$$

$$\widetilde{\mathcal{BM}}^{\widetilde{k}}_{\mathcal{A}}(t) \triangleq \begin{bmatrix} \widetilde{\aleph}^{\widetilde{k}}_{1}(t) & \widetilde{P}^{\widetilde{k}}_{1}(t) \\ \vdots & \vdots \\ \widetilde{\aleph}^{\widetilde{k}}_{h}(t) & \widetilde{P}^{\widetilde{k}}_{h}(t) \end{bmatrix}, \tag{4.38}$$

$$\mathcal{SI}^{\widetilde{k}}_{\mathcal{A}}(t) \triangleq \frac{1}{h} \times \sum_{i=1}^{h} \begin{cases} \frac{\widetilde{\aleph}^{\widetilde{k}}_{i}(t)}{\max\limits_{\forall \rho_i \in \mathcal{P}} \overline{\aleph}_i(t)} + \frac{P^{\mathtt{n}}(t)}{\widetilde{P}^{\widetilde{k}}_{i}(t)} & \widetilde{\aleph}^{\widetilde{k}}_{i}(t) > 0, \\ \frac{\widetilde{\aleph}^{\widetilde{k}}_{i}(t)}{\min\limits_{\forall \rho_i \in \mathcal{P}} \underline{\aleph}_i(t)} + \frac{\widetilde{P}^{\widetilde{k}}_{i}(t)}{P^{\mathtt{u}}(t)} & \widetilde{\aleph}^{\widetilde{k}}_{i}(t) < 0, \end{cases} \tag{4.39}$$

where $\widetilde{\ell}_{\mathcal{A}}(t)$ (produced once at each time interval) defines actions that the aggregator makes regarding prosumers' behaviors. Fig. 4.16 shows how *behavior matrices* $\widetilde{\mathcal{BM}}^{\widetilde{k}}_{\mathcal{A}}(t)$, $1 \leq \widetilde{k} \leq \widetilde{K}$ are generated randomly according to Equations (4.34)-(4.36). The tilde symbol in all notations are related to the aggregator. $\widetilde{\ell}_{\mathcal{A}}(t)$ is sorted descendingly by satisfaction index $\mathcal{SI}^{\widetilde{k}}_{\mathcal{A}}(t) \in (0, 2]$ measuring to which extent the aggregator is satisfied with $\widetilde{\mathcal{BM}}^{\widetilde{k}}_{\mathcal{A}}(t)$.

4.3 Conclusions

This chapter develops a holistic SALSA system, which is derived from the formal framework proposed in the previous chapter. It targets two customers and operations domain. In the first domain, prosumers, as the main actors, own a set of smart appliances, a PV, and a BESS. To operate these equipment, they interact with EMSs. The aggregator, by using its OMS, responds requests of OMSs. Therefore, this chapter has led to the following Application Level Contributions:

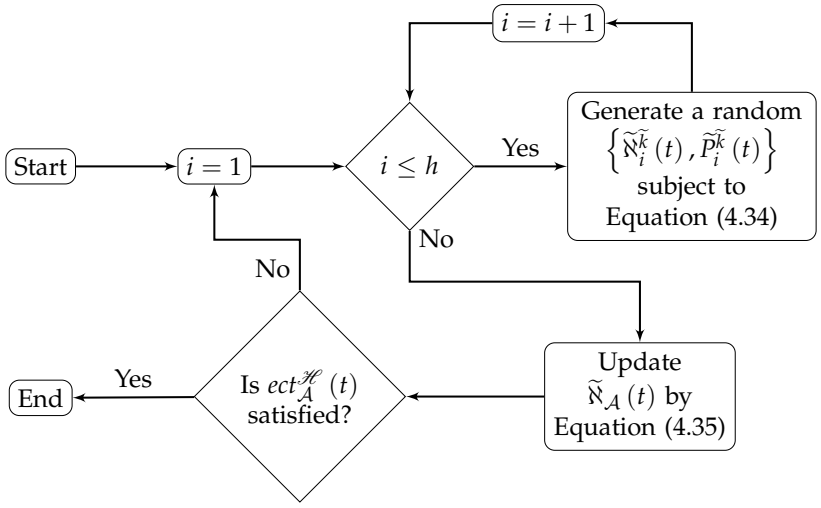


Fig. 4.16: Flow chart of generating a feasible behavior matrix $\widetilde{\mathcal{BM}}_{\mathcal{A}}^{\tilde{k}}(t)$.

ALC-1: Design, implementation, and evaluation of the SALSA system consisting of the following high-level features: i) conforming to the hierarchical grid infrastructure; ii) operating using no forecasting services and historical data; iii) performing in real-time (one minute to one hour); and iv) following an easily expandable agent-based modeling design;

ALC-2: Design and support different LDS mechanisms to enable the load shifting from peak to off-peak periods and develop a novel load demand buffering sub-system to serve hundreds of thousands of prosumers (including diverse appliance types);

ALC-3: Model, develop, and implement a novel concept of flexibility zone for synthesizing diverse flexibility characteristics by the integration of PVs with BESSs in two MO-MINLP models for prosumers to schedule their appliances and share their surplus energy with the grid and for the aggregator to efficiently match prosumers' demands with surpluses;

ALC-4: Design and implement dynamic ECT policies for prosumers and the aggregator (complied with the smart grid operating regimes defined by the USEF) following day-ahead peak consumption, real-time aggregated load demands, and day-ahead electricity prices.

Chapter 5

Agent-Based Negotiation Approach

Prosumers' rational behaviors are more pronounced when their uncertainty about the decision space of the aggregator increases. Due to the promising outlook of introducing prosumers into the smart grid, this dissertation employs an approach to enable the concurrent negotiation on *power* and *price* issues with *packaged offers* given that the negotiators have no prior knowledge about the flexibility information and utility functions of each other [68]. To model such approach, the following key elements are needed: i) notion of a solution to the negotiation problem, and ii) negotiation protocol and strategy. The negotiation protocol and strategy define how negotiators provide and prepare offers, respectively. The content of this chapter originates and adapts from the following publications:

[9] Armin Ghasem Azar, Hamidreza Nazaripouya, Behnam Khaki, Chi-Cheng Chu, Rajit Gadh, and Rune Hylsberg Jacobsen, "A Non-Cooperative Framework for Coordinating a Neighborhood of Distributed Prosumers," Submitted to *IEEE Transactions on Smart Grid*, 2017

5.1 Negotiators

Fig. 5.1 depicts the behavior of work-flow executions in the framework. The negotiation procedure is conducted between time intervals t and $t + \Delta\tau$ for maximum $\mathcal{T} \in \mathbb{N}$ iterations (set arbitrarily).

5.1.1 Virtual Power Plant

The framework, to alleviate the challenges of h parallel bilateral negotiations between prosumers and the aggregator, where each negotiation in the worst

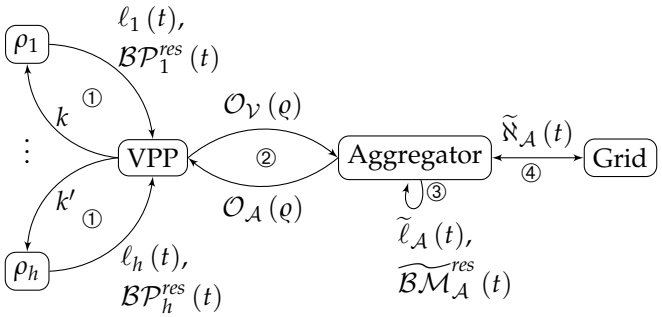


Fig. 5.1: Behavior work-flow of the agents in the framework between each t and $t + \Delta\tau$. Procedures ①, ③, and ④ are done only once while ② takes maximum $\mathcal{T}$ iterations. Notations are described in Sections 5.1 and 5.2 [9].

case to reach an agreement can take $\mathcal{T}$ iterations, utilizes an intermediate VPP to negotiate, on behalf of prosumers. Let

$$\mathcal{O}_{\mathcal{V}}(\varrho) = \begin{bmatrix} \mathcal{BP}_1^k(t) \\ \vdots \\ \mathcal{BP}_h^{k'}(t) \end{bmatrix}, \tag{5.1}$$

where $\mathcal{O}_{\mathcal{V}}(\varrho)$ is an *offer package* sent from the VPP to the aggregator at negotiation iteration $1 \leq \varrho \leq \mathcal{T}$. Behavior pairs $\mathcal{BP}_1^k(t)$ and $\mathcal{BP}_h^{k'}(t)$ point to rows k and k' (not necessarily equivalent) in $\ell_1(t)$ and $\ell_h(t)$, respectively (see Equation (4.29)). Let

$$U_{\mathcal{V}}(\mathcal{SI}_{\mathcal{V}}(\varrho)) = 1 - \frac{\sum_{i=1}^{h} \left(\frac{1}{2} \times \mathcal{SI}_i^k(t)\right)^2}{h}, \tag{5.2}$$

$$\mathcal{SI}_{\mathcal{V}}(\varrho) \triangleq \bigcup_{i=1}^{h} \mathcal{SI}_i^k(t), 1 \leq k \leq K, \tag{5.3}$$

where $U_{\mathcal{V}} \in [0, 1)$ is the VPP's utility function. We assume a very general hyper-quadric utility function [90] for negotiators, which is private, continuous, and strictly concave [68]. By "private," negotiators have no knowledge about other negotiator's utility function. $\mathcal{SI}_{\mathcal{V}}(\varrho)$ is the *union* of satisfaction indexes of prosumers' behavior pairs. Superscripts k in $\mathcal{SI}_i^k(t), \forall \rho_i \in \mathcal{P}$ are not necessarily equivalent. Let

$$\mathcal{O}_{\mathcal{V}}^{res}(t) = \begin{bmatrix} \mathcal{BP}_1^{res}(t) \\ \vdots \\ \mathcal{BP}_h^{res}(t) \end{bmatrix}, \tag{5.4}$$

$$\mathcal{BP}_i^{res}(t) = \left(\aleph_i^{res}(t), P_i^{res}(t)\right), \tag{5.5}$$

$$\aleph_i^{res}(t) = \begin{cases} \underline{\aleph}_i(t) & \sum_{\substack{flex_{j,i}(t)=0, \\ \forall sa_{j,i} \in SA_i}} ec_{j,i}(t) \leq PV_i^G(t), \\ \overline{\aleph}_i(t) & \text{otherwise}, \end{cases} \tag{5.6}$$

$$P_i^{res}(t) = \begin{cases} P^{\mathbb{l}}(t) & \aleph_i^{res}(t) > 0, \\ P^{\mathbb{u}}(t) & \aleph_i^{res}(t) < 0, \end{cases} \tag{5.7}$$

where $\mathcal{O}_{\mathcal{V}}^{res}(t)$ is the reservation offer package of the VPP including prosumers' reservation behavior pair $\mathcal{BP}_i^{res}(t)$. $\aleph_i^{res}(t)$ and $P_i^{res}(t)$ are the reservation power and price offers of prosumer ρ_i at time interval t. Prosumers in the worst case have to: i) satisfy appliances with no flexibility remained, and ii) utilize the electric power generated by the PV completely. The reservation price offer equals to either the lowest ($\aleph_i^{res}(t) > 0$) or highest ($\aleph_i^{res}(t) < 0$) possible offerable electricity price, respectively. $\mathcal{SI}_{\mathcal{V}}^{res}(t)$, as the satisfaction index of $\mathcal{O}_{\mathcal{V}}^{res}(t)$, is union of $\mathcal{SI}_i^{res}(t) \in (0, 2]$ (calculated by Equation (4.31)) associated with $\mathcal{BP}_i^{res}(t)$. Any offer package with the utility value less than $U_{\mathcal{V}}\left(\mathcal{SI}_{\mathcal{V}}^{res}(t)\right)$ is unacceptable to VPP. The VPP in the end of negotiation returns indexes of agreed behavior pairs, i.e., $k \leq K, \forall \rho_i \in \mathcal{P}$, to the prosumers (see Fig. 5.1).

5.1.2 Aggregator

Let

$$\mathcal{O}_{\mathcal{A}}(\varrho) = \widetilde{\mathcal{BM}}_{\mathcal{A}}^{\tilde{k}}(t), \tag{5.8}$$

where $\mathcal{O}_{\mathcal{A}}(\varrho)$, equivalent to a behavior matrix in $\widetilde{\ell}_{\mathcal{A}}(t)$, is an offer package sent from the aggregator to the VPP. Let

$$U_{\mathcal{A}}\left(\mathcal{SI}_{\mathcal{A}}^{\tilde{k}}(t)\right) = 1 - \left(\frac{1}{2} \times \mathcal{SI}_{\mathcal{A}}^{\tilde{k}}(t)\right)^2, \tag{5.9}$$

where $U_{\mathcal{A}} \in [0, 1)$ is the aggregator's utility function, which follows the same rule as prosumers' [68]. $\mathcal{SI}_{\mathcal{A}}^{\tilde{k}}(t)$ is the satisfaction index of behavior matrix $\widetilde{\mathcal{BP}}_{\mathcal{A}}^{\tilde{k}}(t)$. Let

$$\mathcal{O}_{\mathcal{A}}^{res}(t) = \begin{bmatrix} \widetilde{\mathcal{BP}}_1^{res}(t) \\ \vdots \\ \widetilde{\mathcal{BP}}_h^{res}(t) \end{bmatrix}, \tag{5.10}$$

$$\widetilde{\mathcal{BP}}_i^{res}(t) = \left(\min_{\forall \rho_i \in \mathcal{P}} \underline{\aleph}_i(t), P^{\mathbb{l}}(t)\right), \tag{5.11}$$

where $\mathcal{O}_{\mathcal{A}}^{res}(t)$ is the reservation offer package of the aggregator denoting $h \times \min_{\forall \rho_i \in \mathcal{P}} \underline{\aleph}_i(t)$ amount of electric power must be exchanged (in the worst case) with the grid for $P^{\mathbb{1}}(t)$. This reservation offer package is coupled with a satisfaction index $\mathcal{SI}_{\mathcal{A}}^{res}(t) \in (0, 2]$ (calculated by Equation (4.39)). Similar to the VPP, the aggregator will not accept any offer package with the utility value less than $U_{\mathcal{A}}(\mathcal{SI}_{\mathcal{A}}^{res}(t))$. The following part explains the protocol and strategy the negotiators follow during the negotiation process.

5.2 Negotiation Protocol and Strategy

We employ an *alternating-offer protocol* [91], where the VPP produces an offer and the aggregator either accepts it or produces a new one. The negotiation begins with offer packages produced with the highest possible utility values and continues with offer packages with lower utility values. It terminates when: i) an offer on the table is acceptable to both negotiators, or ii) it reaches iteration $\mathcal{T}$ with no offer accepted. Let

$$\mathcal{O}_{\mathcal{V}}(1) = \begin{bmatrix} \mathcal{BP}_1^1(t) \\ \vdots \\ \mathcal{BP}_h^1(t) \end{bmatrix}, \tag{5.12}$$

$$\mathcal{O}_{\mathcal{A}}(1) = \widetilde{\mathcal{BM}}_{\mathcal{A}}^{1}(t), \tag{5.13}$$

$$U_{\mathcal{V}}(\mathcal{SI}_{\mathcal{V}}(\varrho - 1)) \leq U_{\mathcal{V}}(\mathcal{SI}_{\mathcal{V}}(\varrho)) \leq U_{\mathcal{V}}(\mathcal{SI}_{\mathcal{V}}^{res}(t)), \tag{5.14}$$

$$U_{\mathcal{A}}(\mathcal{SI}_{\mathcal{A}}(\varrho - 1)) \leq U_{\mathcal{A}}(\mathcal{SI}_{\mathcal{A}}(\varrho)) \leq U_{\mathcal{A}}(\mathcal{SI}_{\mathcal{A}}^{res}(t)), \tag{5.15}$$

where $\mathcal{O}_{\mathcal{V}}(1)$ and $\mathcal{O}_{\mathcal{A}}(1)$ are initial preferred offer packages of the VPP and aggregator, respectively. Since $\ell_i(t), \forall \rho_i \in \mathcal{P}$ and $\widetilde{\ell}_{\mathcal{A}}(t)$ are sorted descendingly, the initial offer packages provide the highest utility value. Negotiators gradually produce offer packages with lower utility values over negotiation iterations. They neither propose nor accept any offer package with utility value lower than their reservation utility value. To propose a new offer package, they follow the following two consecutive procedures:

5.2.1 Reactive Utility Value Concession

The negotiation approach assumes each negotiator's utility value obtained by an agreement is higher than the one with no agreement. Therefore, they prefer to concede over risking negotiation breakdown. Let

$$\mu_{\mathcal{V}}(\varrho) = U_{\mathcal{V}}(\mathcal{O}_{\mathcal{V}}(1)) - \mathcal{O}_{\mathcal{V}}^{res}(t) \times \left(\frac{\varrho}{\mathcal{T}}\right)^{\frac{1}{\epsilon}}, \tag{5.16}$$

$$\mu_{\mathcal{A}}(\varrho) = U_{\mathcal{A}}\left(\mathcal{O}^{1}_{\mathcal{A}}(t)\right) - \mathcal{O}^{res}_{\mathcal{A}}(t) \times \left(\frac{\varrho}{\mathcal{T}}\right)^{\frac{1}{\epsilon}}, \tag{5.17}$$

where $\mu_{\mathcal{V}}(\varrho), \mu_{\mathcal{A}}(\varrho) \in [0,1)$ are monotonically decreasing time-dependent concession values of the VPP and aggregator, respectively [68]. Their values only depend on each negotiator's reservation utility value and the number of negotiation iterations passed so far with the decay rate $\epsilon \in \mathbb{R}_{>0}$ [92]. As the second assumption, negotiators are assumed to be reactive. Hence, their concession rate should depend on their perception of the utility value of other party's offer packages given: i) whether the current offer of the opponent negotiator provides higher utility value than the negotiator's reservation utility value, and ii) the negotiator's perception of how much the other party has conceded. One reason for a negotiator to stop decreasing its desired utility value over time is to gain higher utility. This happens if the other negotiator, without realizing that the negotiator has stopped conceding, accepts time-dependent concession values at all negotiation iterations. This behavior is called the "deliberate stopping of concession." As a result, Let

$$\kappa_{\mathcal{V}}(\varrho) = \left(U_{\mathcal{V}}\left(\mathcal{SI}^{temp}_{\mathcal{V}}\right) - U_{\mathcal{V}}\left(\mathcal{SI}^{temp'}_{\mathcal{V}}\right)\right)^{+}, \tag{5.18}$$

$$\kappa_{\mathcal{A}}(\varrho) = \left(U_{\mathcal{A}}\left(\mathcal{SI}^{temp}_{\mathcal{A}}\right) - U_{\mathcal{A}}\left(\mathcal{SI}^{temp'}_{\mathcal{A}}\right)\right)^{+}, \tag{5.19}$$

where $\kappa_{\mathcal{V}}(\varrho), \kappa_{\mathcal{A}}(\varrho) \in [0,1)$ are reactive concession values of the VPP and aggregator, respectively, and $y^{+} = \max\{0, y\}$. The VPP, using Equation (4.31), calculates $\mathcal{SI}^{temp}_{\mathcal{V}}$ and $\mathcal{SI}^{temp'}_{\mathcal{V}}$ for $\mathcal{O}_{\mathcal{A}}(\varrho)$ and $\mathcal{O}_{\mathcal{A}}(\varrho - 1)$, respectively. The aggregator, by using Equation (4.39), follows a similar procedure. Then, let

$$\Pi_{\mathcal{V}}(\varrho) = \min\{\mu_{\mathcal{V}}(\varrho), \Pi_{\mathcal{V}}(\varrho - 1) - \kappa_{\mathcal{V}}(\varrho)\}, \tag{5.20}$$

$$\Pi_{\mathcal{A}}(\varrho) = \min\{\mu_{\mathcal{A}}(\varrho), \Pi_{\mathcal{A}}(\varrho - 1) - \kappa_{\mathcal{A}}(\varrho)\}, \tag{5.21}$$

where $\Pi_{\mathcal{V}}(\varrho), \Pi_{\mathcal{A}}(\varrho) \in [0,1)$ are desired utility values of the VPP and the aggregator at iteration ϱ, respectively. Negotiators only accept an offer package that provides a utility value equal to or higher than their desired utility value at that iteration.

5.2.2 New Offer Package Generation

Let us assume $Z_{\mathcal{V}}(t)$ (including maximum K^{h} offer packages, see Equation (4.29)) and $Z_{\mathcal{A}}(t)$ (including maximum $\widetilde{K}$ possible feasible offer packages, see Equation (4.37)) are the convex *feasible offer package sets* of the VPP and the aggregator, respectively. These offer packages provide negotiators with utility value equal to or no less than their reservation offer package's utility value. For an agreement to exist, let $\mathscr{Z}(t) = Z_{\mathcal{V}}(t) \bigcap Z_{\mathcal{A}}(t) \neq \emptyset, \forall t$ remain unchanged during the negotiation, where $\mathscr{Z}(t)$ is the zone of agree-

ment denoting the common intersection of the feasible offer package sets. If an offer package is within $\mathscr{Z}(t)$, a negotiator may not accept it if it yields a utility value lower than the negotiator's *current* desired utility value. To make an acceptable agreement, negotiators keep conceding to their reservation utility values subject to the nonempty zone of agreement at each time interval. Thus, geometrically speaking, in negotiation, the negotiators' goal is to find a point in the zone of agreement, under the restriction that this zone is unknown to negotiators and none of them has any explicit knowledge about each other's utility functions [68].

Let ϱ be the negotiation iteration when it is the VPP's turn to produce a new offer package. Let $\mathcal{BP}_i^k(t) \in \mathcal{O}_{\mathcal{V}}(\varrho-1)$. The VPP (temporarily) updates $\mathcal{O}_{\mathcal{V}}(\varrho-1)$ with behavior pairs $\mathcal{BP}_i^{k'}(t), \forall k+1 \leq k' \leq K$ and expands $\overline{Z}_{\mathcal{V}}(\varrho)$ with the updated offer packages *individually only if* each returns a utility value equivalent to $\Pi_{\mathcal{V}}(\varrho)$. $\overline{Z}_{\mathcal{V}}(\varrho) \subseteq Z_{\mathcal{V}}(t), \forall \varrho \leq \mathcal{T}$ is the *continuously expanding* feasible offer package subset of the VPP. The aggregator at iteration $\varrho+1$ determines $\widetilde{k}$ where $\widetilde{\mathcal{BM}}_{\mathcal{A}}^{\widetilde{k}}(t) \in \widetilde{\ell}_{\mathcal{A}}(t)$. Then, it updates $\overline{Z}_{\mathcal{A}}(\varrho+1)$ with new offer packages $\widetilde{\mathcal{BM}}_{\mathcal{A}}^{\widetilde{k}+1}(t), \forall \widetilde{k}+1 \leq \widetilde{K}$, where each provides the aggregator with a utility value equal to $\Pi_{\mathcal{A}}(\varrho+1)$. $\overline{Z}_{\mathcal{A}}(\varrho) \subseteq Z_{\mathcal{A}}(t), \forall \varrho \leq \mathcal{T}$ is the continuously expanding feasible offer package subset of the aggregator. Let

$$\mathcal{O}_{\mathcal{V}}(\varrho) = \vartheta_{\overline{Z}_{\mathcal{V}}(\varrho)}[\xi(\varrho)] = \underset{q \in \overline{Z}_{\mathcal{V}}(\varrho)}{\arg\min} \|q - \xi(\varrho)\|, \tag{5.22}$$

$$\mathcal{O}_{\mathcal{A}}(\varrho) = \vartheta_{\overline{Z}_{\mathcal{A}}(\varrho)}[\xi(\varrho)] = \underset{q \in \overline{Z}_{\mathcal{A}}(\varrho)}{\arg\min} \|q - \xi(\varrho)\|, \tag{5.23}$$

$$\xi(\varrho) = \varphi_{\mathcal{V}}(\varrho) \times \mathcal{O}_{\mathcal{V}}(\varrho-1) + \varphi_{\mathcal{A}}(\varrho) \times \mathcal{O}_{\mathcal{A}}(\varrho-1), \tag{5.24}$$

$$\varphi_{\mathcal{V}}(\varrho) + \varphi_{\mathcal{A}}(\varrho) = 1, \tag{5.25}$$

where ϑ is the operator of projecting the weighted offer package $\xi(\varrho)$, created based on the latest offers made by all agents, on current continuously expanding feasible offer package subsets $\overline{Z}_{\mathcal{V}}(\varrho)$ and $\overline{Z}_{\mathcal{A}}(\varrho)$ [93]. $\arg\min \|\cdot\|$ is the Frobenius norm with argument of minimum. Note this method generates an offer that is acceptable to the negotiator and is closest (in terms of Euclidean distance) to the weighted offer package $\xi(\varrho)$. $\varphi_{\mathcal{V}}(\varrho), \varphi_{\mathcal{A}}(\varrho) \in (0,1)$ are the weights that each negotiator puts on the other's offer package.

In Algorithms 5.1 and 5.2, we provide the pseudo-code for the overall communication steps and the negotiation approach, respectively. Steps in the former are in line with the data flow diagram depicted in Fig. 5.1. Fig. 5.2 illustrates a conceptual example of the offer package space during the negotiation and shows how the VPP and the aggregator negotiate with each other over, for example, $\mathcal{T} = 9$ iterations [68]. Offer packages existing on each concession curve have equal utility values. The negotiation terminates when $\max\{\|\mathcal{O}_{\mathcal{V}}(9) - \xi(9)\|, \|\mathcal{O}_{\mathcal{A}}(9) - \xi(9)\|\} < \nabla$, which denotes that if the Euclidean distances between the current iteration's offer packages and

the weighted offer package are less than a convergence tolerance $\nabla \in \mathbb{R}_{>0}$.

Algorithm 5.1: Communication steps in the framework between time intervals t and $t + \Delta\tau$ [9].

```
   // Prosumers' part;
1  foreach ρ_i ∈ P do
2  |  See line 12 in Algorithm 4.1;
3  end
   // VPP's part (i);
4  Determine the reservation offer package O_V^res (t);
5  Produce the first offer package O_V (1);
   // Aggregator's part;
6  Run the NSGA-III to produce ℓ̃_A (t);
7  Determine the reservation offer package O_A^res (t);
8  Produce the first offer package O_A (1);
   // Negotiation approach;
9  Run Algorithm 5.2 // VPP's part (ii);
10 Return the indexes of agreed behavior pairs to prosumers (see line 13
     in Algorithm 4.1);
```

5.3 Solution Concept for the Negotiation Approach

The use of the solution concept in this negotiation approach, where the negotiators have no information about their opponents, is in the spirit of Herbert Simon [94]. Through computational experiments, The authors in [68] have demonstrated that such solution concept proposed in the negotiation approach yields performance sufficiently close to the Nash bargaining solution, which is a different definition proposed for a proper negotiation solution [95]. The set of points that satisfy Nash bargaining solution's requirements are all subsets of the zone of agreement. However, computing them requires that all the negotiators have complete knowledge of the preference structure and utility function of the opponents.

The authors in [68] have also analytically proved that: i) the scale of the utility value of each negotiator is of no critical importance, as long as the reservation utility value and the scale of concession are consistent with it; ii) the negotiators, by utilizing the utility value concession strategy described earlier, converge to an agreement acceptable to all in maximum $\mathcal{T}$ iterations, if the zone of agreement is nonempty and they concede to reservation utility values in the worst case; and iii) the convergence holds for general concave

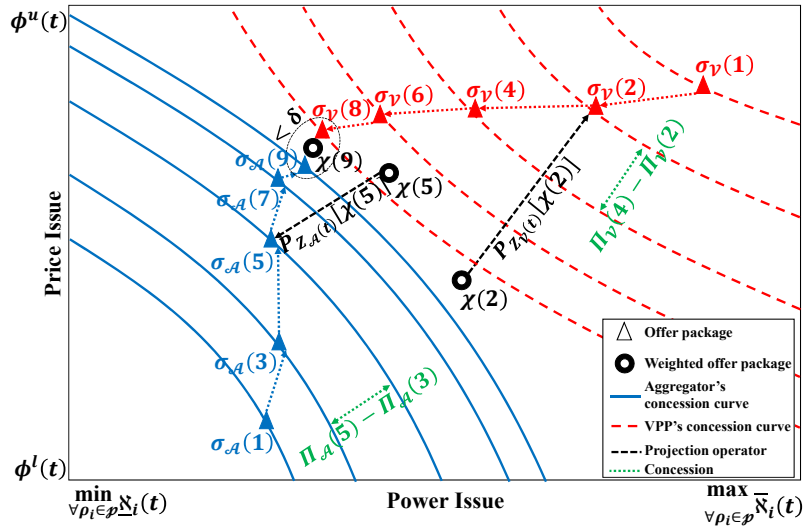


Fig. 5.2: Conceptual example of the offer package space during the negotiation [9].

utility functions as long as all the negotiators concede to their reservation utilities, irrespective of the specific concession strategy they adopt.

5.4 Conclusions

This chapter proposes a non-cooperative approach for coordinating a neighborhood of prosumers. To take advantage of their consumption and generation flexibilities, they individually communicating with an aggregator. The approach, by using the two MO-MINLP models for prosumers and the aggregator developed in the previous chapter, prosumers are able to schedule their appliances and share surplus power with the grid while the aggregator controls the power matching over time. To relieve the burden of parallel bilateral communications between prosumers and the aggregator, this approach employs a VPP, working on behalf of prosumers, to negotiate on packaged power and price offers with the aggregator subject to having no knowledge about each other's preferences and utility functions. Thus, this chapter has made the following Application Level Contributions:

ALC-5: Materialize the trading of prosumers' heterogeneous flexibilities through the negotiation approach;

ALC-6: Model and implement a bilateral multi-issue negotiation approach to enable the negotiation between a VPP (on behalf of prosumers) and the aggregator.

Algorithm 5.2: The negotiation algorithm [9].

IsConverge=False;
$\varrho = 2$;
while $\varrho \leq \mathcal{T}$ *and IsConverge=False* **do**
  Determine agent's turn by $\mathscr{W} = \mathrm{mod}(\varrho, 2)$;
  if $\mathscr{W} = 0$ **then** `// VPP's turn`
    $\mathcal{O}_{\mathcal{A}}(\varrho) = \mathcal{O}_{\mathcal{A}}(\varrho - 1)$;
    $\Pi_{\mathcal{A}}(\varrho) = \Pi_{\mathcal{A}}(\varrho - 1)$;
    Calculate $\Pi_{\mathcal{V}}(\varrho)$ by Equation (5.20);
    for $i = 1$ *to* h **do**
      Determine k, where $\mathcal{BP}_i^k(t) \in \mathcal{O}_{\mathcal{V}}(\varrho - 1)$;
      $\mathcal{O}_{\mathcal{V}}^{temp} = \mathcal{O}_{\mathcal{V}}(\varrho - 1)$;
      Found=0;
      while $k + 1 \leq K$ **do**
        Update $\mathcal{O}_{\mathcal{V}}^{temp}$ with $\mathcal{BP}_i^{k+1}(t)$;
        Calculate $\mathcal{SI}_{\mathcal{V}}^{temp}$ for $\mathcal{O}_{\mathcal{V}}^{temp}$ by Equation (5.3);
        if $U_{\mathcal{V}}\left(\mathcal{SI}_{\mathcal{V}}^{temp}\right) = \Pi_{\mathcal{V}}(\varrho)$ **then**
          Add $\mathcal{O}_{\mathcal{V}}^{temp}$ to $\overline{Z}_{\mathcal{V}}(\varrho)$;
          Found=1;
          $k = k + 1$;
        else if *Found=1* **then**
          $k = K$;
        else
          $k = k + 1$;
        end
      end
    end
    Set $\mathcal{O}_{\mathcal{V}}(\varrho)$ by Equation (5.22);
  else `// Aggregator's turn`
    $\mathcal{O}_{\mathcal{V}}(\varrho) = \mathcal{O}_{\mathcal{V}}(\varrho - 1)$;
    $\Pi_{\mathcal{V}}(\varrho) = \Pi_{\mathcal{V}}(\varrho - 1)$;
    Set $\Pi_{\mathcal{A}}(\varrho)$ by Equation (5.21);
    Determine $\widetilde{k}$, where $\widetilde{\mathcal{BM}}_{\mathcal{A}}^{\widetilde{k}} \in \widetilde{\ell}(t)$;
    Found=0;
    while $\widetilde{k} + 1 \leq \widetilde{K}$ **do**
      if $\mathcal{SI}_{\mathcal{A}}^{\widetilde{k}+1}(t) = \Pi_{\mathcal{A}}(\varrho)$ **then**
        Add $\mathcal{O}_{\mathcal{A}}^{\widetilde{k}+1}$ to $\overline{Z}_{\mathcal{A}}(\varrho)$;
        Found=1;
        $\widetilde{k} = \widetilde{k} + 1$;
      else if *Found=1* **then**
        $k = K$;
      else
        $\widetilde{k} = \widetilde{k} + 1$;
      end
    end
    Set $\mathcal{O}_{\mathcal{A}}(\varrho)$ by Equation (5.23);
  end
  Set $\xi(\varrho)$ by Equation (5.24);
  if $\max\{\|\mathcal{O}_{\mathcal{V}}(\varrho) - \xi(\varrho)\|, \|\mathcal{O}_{\mathcal{A}}(\varrho) - \xi(\varrho)\|\} < \nabla$ **then**
    IsConverge=True;
  else
    $\varrho = \varrho + 1$;
  end
end

Chapter 6

Simulation and Discussion

This chapter presents the simulation results and evaluates the performance of the proposed contributions. It is divided into two main sections: i) defining economic and environmental assessment metrics for performance evaluation, and ii) proposing two case studies based on the research progress made during the three-year PhD study. The content of this chapter partially originates and adapts from the following publications:

[1] Armin Ghasem Azar, Rune Hylsberg Jacobsen, and Qi Zhang, "Aggregated Load Scheduling for Residential Multi-Class Appliances: Peak Demand Reduction," In *IEEE International Conference on the European Energy Market (EEM)*, 2015, pages 1-6, doi: 10.1109/EEM.2015.7216702

[5] Armin Ghasem Azar and Rune Hylsberg Jacobsen, "Appliance Scheduling Optimization for Demand Response," *International Journal on Advances in Intelligent Systems*, vol. 2, no 1&2, 2016, pages 50-64, Link to paper

[8] Armin Ghasem Azar and Rune Hylsberg Jacobsen, "Agent-Based Charging Scheduling of Electric Vehicles," In *IEEE Online Conference on Green Communications (OnlineGreenComm)*, 2016, pages 64-69, doi: 10.1109/OnlineGreenCom.2016.7805408

[9] Armin Ghasem Azar, Hamidreza Nazaripouya, Behnam Khaki, Chi-Cheng Chu, Rajit Gadh, and Rune Hylsberg Jacobsen, "A Non-Cooperative Framework for Coordinating a Neighborhood of Distributed Prosumers," Submitted to *IEEE Transactions on Smart Grid*, 2017

6.1 Evaluation Metrics

To verify the effectiveness of the contributions in different directions, the following assessment metrics are defined.

6.1.1 Environmental Metrics

Environmental metrics are designed to assess the environmental impact of the SALSA system on the power grid. The impact is primarily related to using prosumers' flexibility capabilities and RESs.

6.1.1.1 Peak Demand Reduction

This metric determines how much the proposed framework is successful in shaving the peak demand. Let

$$\mathrm{PDR} \triangleq \left(1 - \frac{\max\limits_{\forall t} \sum\limits_{i=1}^{h} \sum\limits_{j=1}^{\mathfrak{M}_i} ec_{j,i}(t) \times dec_{j,i}(t)}{\max\limits_{\forall t'} \sum\limits_{i=1}^{h} \sum\limits_{j=1}^{\mathfrak{M}_i} ec_{j,i}(t')} \right) \times 100, \tag{6.1}$$

where t and t' are time intervals, at which the grid confronts the maximum peak demand *with* and *without* using the proposed framework, respectively. PDR equals to zero, when $t = t'$.

6.1.1.2 Peak-to-Average Ratio

This metric measures how much higher the peak demand is than average demands over a single simulation. A high PAR means a large fluctuation in daily load demand. Let

$$\mathrm{PAR} \triangleq \frac{\max\limits_{\forall t \leq T} \sum\limits_{i=1}^{h} \sum\limits_{j=1}^{\mathfrak{M}_i} ec_{j,i}(t) \times dec_{j,i}(t)}{\sum\limits_{t=1}^{T} \sum\limits_{i=1}^{h} \sum\limits_{j=1}^{\mathfrak{M}_i} ec_{j,i}(t) \Big/ T}. \tag{6.2}$$

6.1.2 Economic and Profit Metrics

These metrics allow the analysis of economic performance and impact of the proposed SALSA system on prosumers' daily life (in terms of load consumption). Their main purpose is to provide the quantitative information needed to make a judgment on deployment of the SALSA system.

6.1.2.1 Average Appliance Operation Delay

This metric calculates the delay in delivering appliances in the completed status. Let

$$\mathrm{AOD} \triangleq \frac{\sum_{i=1}^{h} \sum_{j=1}^{\mathfrak{M}_i} \left(\varepsilon_{j,i} - \beta_{j,i}\right)}{\sum_{i=1}^{h} \mathfrak{M}_i}. \tag{6.3}$$

6.1.2.2 Average Flexibility Usage Rate

This metric considers how much of prosumers' flexibilities are traded in the market. Let

$$\mathrm{FUR} \triangleq \frac{\sum_{i=1}^{h} \sum_{j=1}^{\mathfrak{M}_i} \frac{\varepsilon_{j,i} - \beta_{j,i}}{oft_{j,i} - ost_{j,i}}}{\sum_{i=1}^{h} \mathfrak{M}_i} \times 100. \tag{6.4}$$

6.1.2.3 Average Prosumer Cost-Benefit

This metric evaluates the cost-effectiveness of the framework for prosumers. It studies how much money they averagely earn/spend *with* and *without* negotiating and exchanging power with the grid. Let

$$\mathrm{PCB} \triangleq \left(1 - \sum_{t=1}^{T} \sum_{i=1}^{h} \frac{\aleph_i^k(t) \times P_i^k(t)}{\sum_{j=1}^{\mathfrak{M}_i} ec_{j,i}(t) \times P_{\mathcal{G}}^{\mathfrak{u}}(t)}\right) \times 100, \tag{6.5}$$

where $k \leq K_i$ is the behavior pair index of prosumer ρ_i, on which the VPP in the end of negotiation process at time interval t has agreed with the aggregator. $T \equiv \texttt{HH:MM:SS}$ is the last simulated time interval.

6.1.2.4 Average Self Load-Satisfaction Rate

This metric studies the local energy utilization for prosumers. Let

$$\mathrm{SLR} \triangleq \frac{\sum_{t=1}^{T} \sum_{i=1}^{h} \frac{\aleph_i^{\mathrm{P2L}}(t) + \aleph_i^{\mathrm{B2L}}(t)}{\sum_{j=1}^{\mathfrak{M}_i} ec_{j,i}(t) + PV_i^{\mathcal{G}}(t) + B_i^{\overline{d\text{-}chg}}}}{h \times T} \times 100. \tag{6.6}$$

6.1.2.5 Average Self Sufficiency Rate

This metric evaluates PVs' capability in maximizing the comfort level of prosumers without purchasing any amount of power from the grid. Let

$$\text{SSR} \triangleq \frac{\sum_{t=1}^{T}\sum_{i=1}^{h}\frac{\aleph_i^{\text{P2L}}(t)+\aleph_i^{\text{P2B}}(t)}{\sum_{j=1}^{\mathfrak{M}_i} ec_{j,i}(t)+PV_i^G(t)}}{h\times T}\times 100. \tag{6.7}$$

6.1.3 Computation Time

Measuring the CPU time (or process time) of different parts of the SALSA is to quantify the overall busyness of the system. This is the time taken from the start of until the end of a specific part as measured by an ordinary clock. This metric measures the computation time of the LDS mechanism, NSGA-III algorithm, negotiation approach, and the total simulation. Note the presence of each measurement differs in each case study.

6.2 Simulation Case Studies

As Table 6.1 describes, this section presents two simulation case studies according to the contributions made during the PhD study. The second case study is dependent on the first case study. Case studies will be further explained in Sections 6.2.2.1 and 6.2.2.2.

Table 6.1: Description of simulation case studies.

Case study	PhD period	Feature highlights	Simulation environment
CS-1	1^{st} and 2^{nd} years	• Centralized • LDS mechanism: All • ECT policy: All • Objective: Comfort	• Matlab R2015b • 4 Intel 2.00 GHz Core i7-3537U CPUs • 12 GB memory
CS-2	3^{rd} year	• Decentralized • LDS mechanism: EDF • ECT policy: ECT-P2 • Objectives: Comfort and profit • Integration of PVs and BESSs with the SALSA • Bilateral automated multi-issue negotiation	• Matlab R2017a • 16 Intel 2.3 GHz Xeon E5-2686 CPUs • 64 GB memory

6.2.1 Simulation Data and Setting

Table 6.2 lists the inputs to simulations, which are assumed constant unless otherwise stated. For the PV generation profile, the real data captured from the UCLA Ackerman Union is scaled down from the capacity of 35 kW to 7 kW [10]. Real-time hourly electricity prices are captured from Nord Pool Spot [96], where $\left\{P^{\mathbb{l}}(t), P^{\mathbb{u}}(t)\right\}, \forall t$ and $\left\{P^{\mathbb{l}}_{\mathcal{G}}(t), P^{\mathbb{u}}_{\mathcal{G}}(t)\right\}, \forall t$ are adjusted by fluctuation rates of $\pm 50\%$ and $\pm 20\%$, respectively [63]. Analyzing the price formation is beyond the scope of this dissertation. Table 6.3 describes how consumption scenarios for appliances are created. Start, end, and deadline flexibility times are randomly generated by the normal distribution $\mathcal{N}(\mu, \sigma^2)$ with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$. Load profiles of appliances are captured from [75, 97] with the time resolution of $\Delta\tau = 1$ hour. Refrigerator operates uninterruptedly with no end and flexibility times. Nissan Altra is chosen as the electric vehicle with an empty battery at arrival and fully charged battery at departure [97]. The deadline flexibility concept from the perspective of the air conditioner is the *comfortable temperature range* [75], where 25°C and ±3°C are prosumers' desired temperature set point and flexibility, respectively. Values for ECTs are defined in each case study.

Table 6.2: Constant input values for the simulation case studies [9].

Parameter	Value	Parameter	Value	Parameter	Value
$\Delta\tau$	1 hour	$^{*}PV_i^{cap}$	7 kW	$^{*}B_i^{cap}$	13.2 kWh
$^{*}B_i^{\overline{chg}}$	5 kW	$^{*}B_i^{\overline{d\text{-}chg}}$	5 kW	$\mathcal{T}$	50
$\mathcal{Q}$	100	$\mathscr{U}$	20	pc	0.8
pm	0.2	ϵ	0.8	∇	0.01
$^{*}\widetilde{\hbar}_i$	±3 °C	$^{*}\Delta\hbar_i$	0.05 °C	$^{*}tsp_i$	25 °C
$^{\dagger}\varphi_{\mathcal{V}}(\varrho)$	0.5	$^{\dagger}\varphi_{\mathcal{A}}(\varrho)$	0.5	$^{\dagger}\mathcal{X}(t)$	$\{20, 40, 60, 80\}\%$

$^{*}\forall\rho_i \in \mathcal{P}$, $^{\dagger}\forall\varrho \leq \mathcal{T}$.

Table 6.3: Timetable of generating load demand scenarios of appliances [9].

	$sa_{j,i} \in SA_i, \forall\rho_i \in \mathcal{P}$	$ost_{j,i}$	$\beta_{j,i}$	$ofl_{j,i}$
$\mathcal{NS}_i$	Refrigerator	00:00	N/A	N/A
$\mathcal{S}_i$	Washing Machine	$\mathcal{N}(10,3)$	$ost_{j,i} + 02{:}00$	$\mathcal{N}(16,4)$
	Laundry Dryer	$\mathcal{N}(15,1)$	$ost_{j,i} + 01{:}30$	$\mathcal{N}(21,5)$
	Dishwasher	$\mathcal{N}(17,2)$	$ost_{j,i} + 01{:}40$	$\mathcal{N}(23,2)$
	Electric Vehicle	$\mathcal{N}(19,10)$	$ost_{j,i} + 05{:}00$	$^{*}\mathcal{N}(7.5,1)$
	HVAC	$\mathcal{N}(9,1)$	$\mathcal{N}(21,2)$	25 °C ± 3 °C

*The next day.

6.2.2 Simulation Analysis and Discussion

All statistical results have been averaged over 100 independent simulation runs. The following parts analyzes the simulation results of the two proposed case studies (see Table 6.1 for their features). Note that metrics defined in Sections 6.1.2.4 and 6.1.2.5 are not applicable to the first case study.

6.2.2.1 Case Study 1

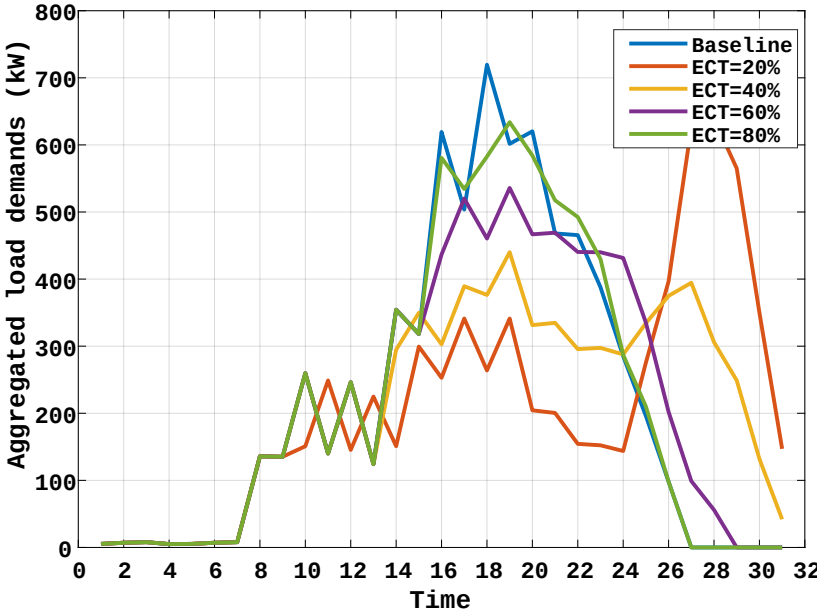
Fig. 6.1 shows fluctuations in the aggregated load demands of 100 prosumers (with and without using the proposed SALSA system) subject to three ECT policies. Table 6.4 also compares the results of applying these policies on 100 prosumers against the evaluation metrics.

Table 6.4: Comparing three ECT policies against the evaluation metrics. For the LDS mechanism, 0-1 Knapsack is used (Dynamic Programming).

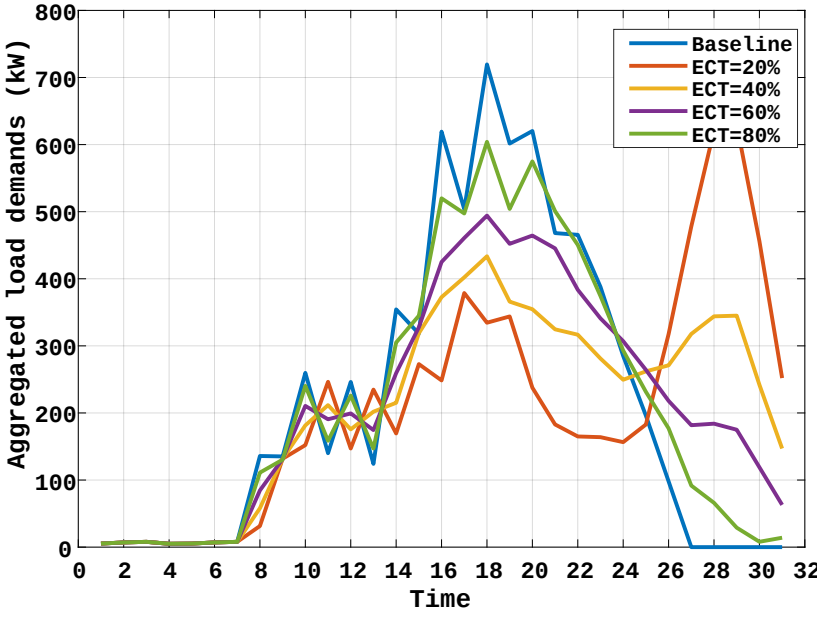
ECT	$^{*}\mathcal{X}_i\,(t)$ (%)	PDR (%)	PAR	AOD (hrs)	FUR (%)	PCB (%)
ECT-P1	20	11.18	3.00	2.05	24.01	38.92
	40	38.83	2.06	1.18	18.71	36.80
	60	25.52	2.25	0.48	9.04	34.82
	80	11.89	2.45	0.13	2.86	33.81
ECT-P2	20	9.51	2.96	2.16	33.68	38.78
	40	37.73	2.04	2.15	28.30	37.45
	60	29.02	2.32	1.05	18.56	36.03
	80	13.22	2.81	1.10	13.23	34.81
ECT-P3	–	23.80	2.57	1.48	21.78	42.45
–	100	0	3.21	0	0	–

$^{*}\forall \rho_i \in \mathcal{P}, \forall t \leq T$.

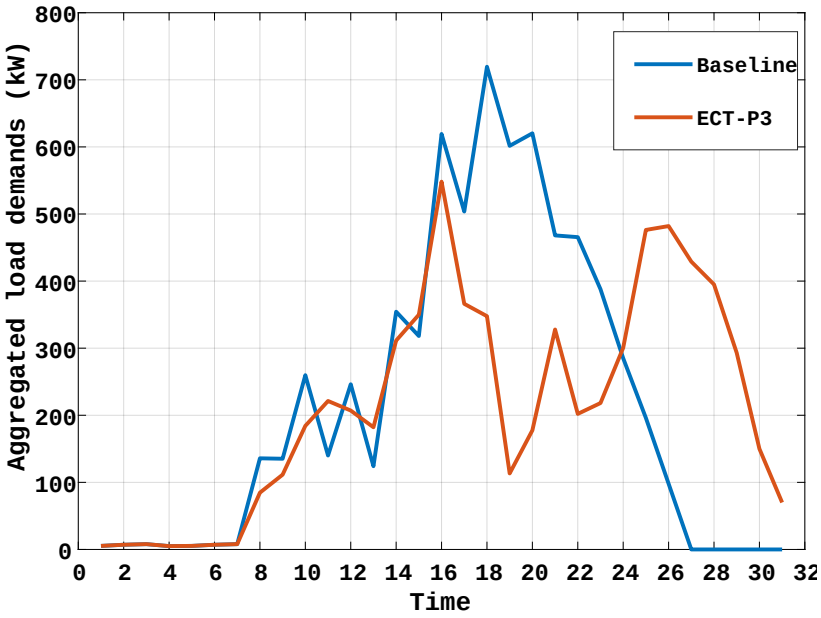
The peak electricity consumption (in baseline) equals to almost 723 kW occurring at 18:00. As Fig. 6.1(a) demonstrates, when $\mathcal{X} = 20\%$, the SALSA has been successfully flattened the aggregated consumption between 14:00 and 24:00. However, this flattening causes another peak period happening between 02:00 to 06:00 (the next day). The reasons are the non-shiftability feature and deadline/temperature flexibility of some appliances, as the FUR is higher compared to other ECT percentages (see FUR column of ECT-P1 part in Table 4.1). Furthermore, in this situation, the Knapsack cannot find any solution for the remaining load demands at some time intervals due to the low remaining ECT. Consequently, the SALSA has to shift all the remaining loads to the next time interval. According to the deadline flexibility constraint, it must allow some loads to start or to continue their operation



(a) ECT-P1 (keeping the aggregated load demands below a constant ECT).



(b) ECT-P2 (keeping the aggregated load demands below the multiplication of the aggregation of load demands at each time interval and a constant ECT).



(c) ECT-P3 (following the day-ahead normalized electricity prices over time).

Fig. 6.1: Fluctuations in the aggregated load demands of 100 prosumers (with and without using the proposed SALSA system) subject to three ECT policies.

apart from the remaining ECT at the next time intervals, which produces another peak time.

When $\mathcal{X} = 40\%$, the Knapsack procedure can permit most of loads to start or to continue their work at corresponding time intervals, and accordingly, the SALSA should shift only a few remaining load demands to the next interval. This will decrease the aggregated consumption at peak times and flatten the aggregated consumption by 38.83%, as represented in Fig. 6.1(b) and Table 6.4. In addition, no significant achievement is found, when the ECT increases to 60% 80%. The reason is that most of load demands are permitted to start or to continue their operation at the time they request. In this situation, the peak reduction ratio is reduced to 25.52% and 11.89%, respectively.

The SALSA system behaves similarly, when ECT-P2 is applied. As Fig. 6.1(b) pictures and the relevant part in Table 4.1 analyzes, ECT-P2 benefits from prosumers' flexibilities more and provides them with more cost reduction (see PCB column of ECT-P2 part in Table 4.1). However, in average, PDRs are slightly lower compared to ECT-P1. Finally, regarding ECT-P3, as Fig. 6.1(c), although this policy is not as successful as the previous two policies, however, prosumers face higher PCB. The reason is that the SALSA attempts to shift load demands from periods with high electricity prices (e.g., 17:00 to 22:00)

to periods with low prices (e.g., 24:00 to 06:00 in the next day).

Fig. 6.2 compares the computation time of applying the first two ECT policies on the simulated prosumers. The most important point about the SALSA system in terms of computational expenses is that it is independent of the ECT policy applied over time. Computation time of applying ECT-P3 is equivalent to $\mathcal{X} = 60\%$ in ECT-P1. Note that, according to the definition provided in Section 6.1.3, only parameters ① and ④ are analyzable.

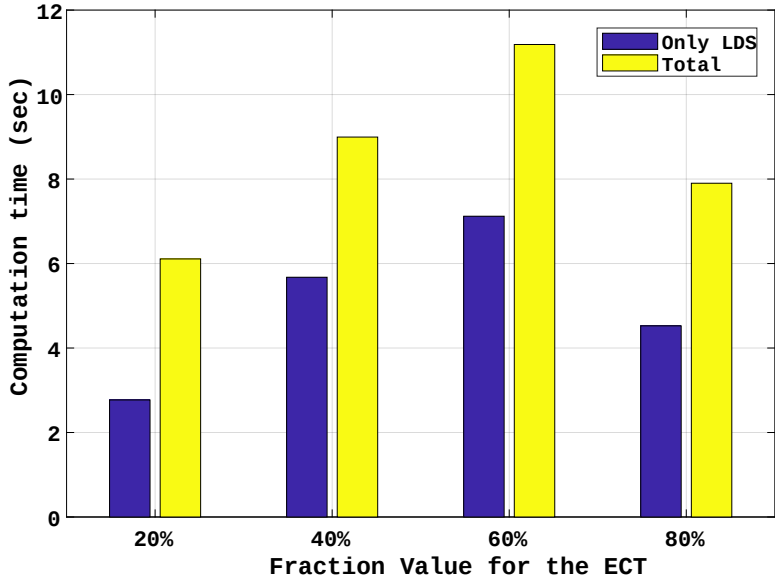
Fig. 6.3 pictures the fluctuations in the aggregated load demands of 100 prosumers with respect to various LDS mechanisms. Table 6.5 compares applying various LDS mechanisms against the applicable assessment metrics. Fig. 6.2 compares the computation time of applying each of these LDS mechanisms. Interestingly, FIFO dominates other mechanism in terms of PDR but fails in terms of PCB. Except the 0-1 Knapsack mechanism, others consumer less computation time, which also results in lower total computation time. The reason is the NP-Completeness of the Knapsack problem. Utilizing the Genetic Algorithm due its repetitive nature for finding the optimal solution in a polynomial time causes the highest computation time compared to other LDS mechanisms.

Table 6.5: Comparing applying various LDS mechanisms against the applicable assessment metrics (ECT-P2 with $\mathcal{X} = 40\%$).

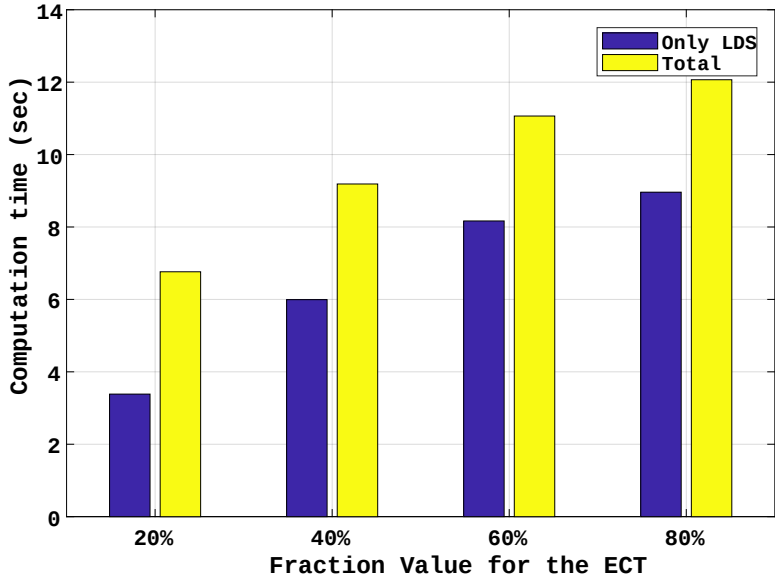
LDS	PDR (%)	PAR	AOD (hrs)	FUR (%)	PCB (%)
0-1 Knapsack (*DP)	37.73	2.04	2.15	28.30	37.45
0-1 Knapsack (†GA)	34.12	2.38	2.29	22.04	34.06
EDF	36.89	2.09	2.19	29.65	36.72
LST	28.97	2.35	2.52	34.40	36.56
LRT	25.04	2.47	1.30	24.36	37.16
RMS	28.37	2.43	3.37	46.97	36.30
FIFO	41.06	1.95	2.01	35.17	31.42
–	0	3.21	0	0	–

* Dynamic Programming, † Genetic Algorithm (Matlab Toolbox).

Fig. 6.5 evaluates the trade-off between number of prosumers, time interval resolution, and total simulation time, when ECT-P2 with $\mathcal{X} = 40\%$ and EDF mechanism are assumed. Table 6.6 compares the utilization of various time interval resolutions against the applicable assessment metrics. As time interval resolution increases, total computation time decreases. Nevertheless, such increase causes: i) the grid to confront lower PDR and higher PAR, and ii) prosumers to experience lower FUR, PCB, and AOD.



(a) ECT-P1.



(b) ECT-P2.

Fig. 6.2: Computation time of scheduling load demands of 100 prosumers subject to ECT-P1&2.

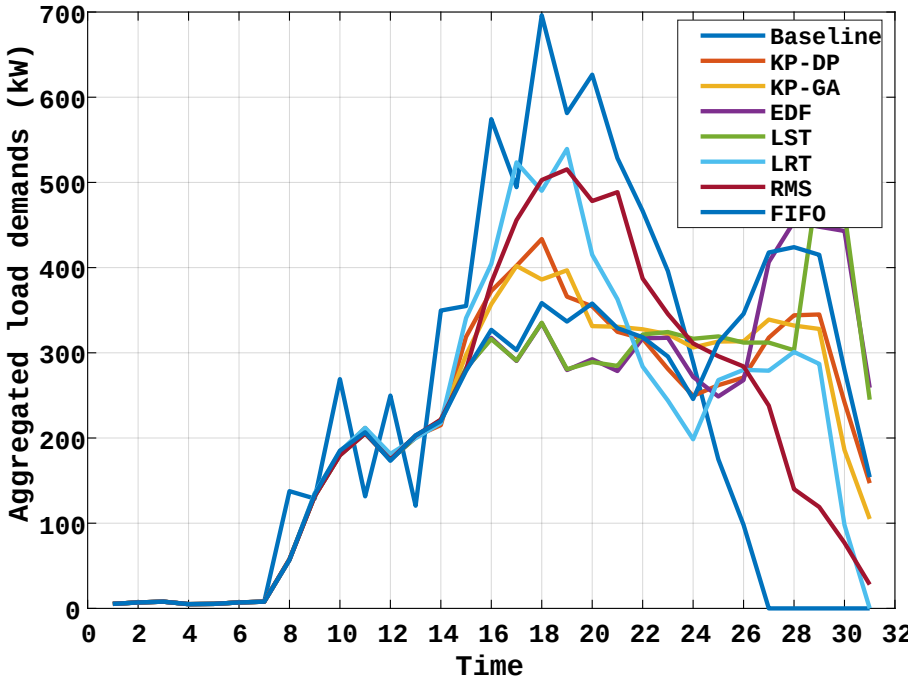


Fig. 6.3: Fluctuations in the aggregated load demands of 100 prosumers with respect to various LDS mechanisms (ECT-P2 with $\mathcal{X} = 40\%$).

Table 6.6: Comparing the utilization of various time interval resolutions against the applicable assessment metrics (ECT-P2 with $\mathcal{X} = 40\%$ and EDF mechanism are assumed).

$\Delta\tau$ (min)	PDR (%)	PAR	AOD (hrs)	FUR (%)	PCB (%)
15	33.84	2.85	2.02	29.58	37.68
30	33.61	2.95	1.49	24.02	34.05
45	33.20	3.06	1.33	20.19	32.39
60	32.98	3.16	1.05	16.46	31.35

6.2.2.1.1 Analysis of the Impact of the SALSA System on Charging Scheduling of Electric Vehicles

Recently, the electrification of the current transportation system has emerged as one of the most crucial challenges in the power grid control. Integrating an increasing number of environmentally friendly EVs with the current grid has yielded to the largest share of total energy consumption growth in the world [43]. Because of this large impact, charging scheduling of EVs is an important research topic since their uncoordinated charging process can jeopardize the efficiency and reliability of the power grid. For more information,

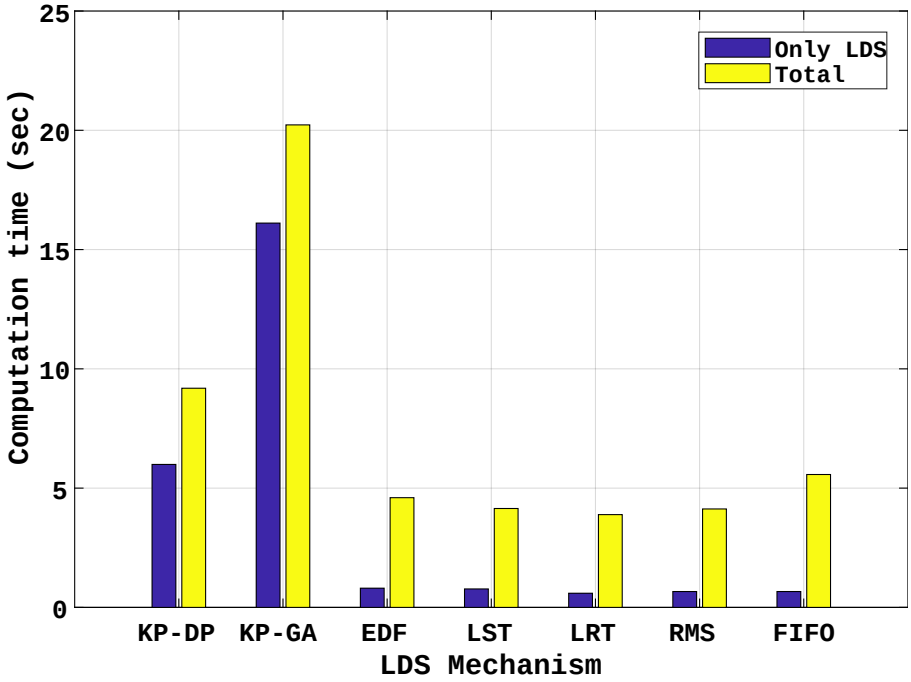


Fig. 6.4: Computation time of each LDS mechanism applied on 100 prosumers (ECT-P2 with $\mathcal{X} = 40\%$).

the reader is referred to [8].

Fig. 6.6 pictures the aggregated charging loads of 100 EVs before scheduling (uncoordinated). Fig. 6.7 illustrates EVs' probability density and real-time hourly basis electricity prices of Scandinavian countries [96]. Fig. 6.8 shows the aggregated charging loads of 100 EVs before and after scheduling subject to ECT-P3 applied over time. The comparison is made according to electricity prices of four Scandinavian countries adjusted in a specific date (May 17, 2016). Table 6.7 analyzes the scheduling performance against the evaluation metrics defined earlier.

For Denmark, its highest and lowest electricity prices are adjusted almost after departure and arrival times, respectively (see Fig. 6.8(a)). Their difference ratio is 1.95. This results in averagely 3.45% Average Prosumer Cost-Benefit (PCB) for each prosumer while averagely 5.22% Peak Demand Reduction (PDR), as pictured in Fig. 6.8(a). This PDR is the result of 10% Average Flexibility Usage Rate (FUR) and 94.19 minutes of Average Appliance Operation Delay (AOD). Residents provide averagely 932.79 minutes of flexibility. For instance, a Nissan Altra arrives at 19:00 and will depart at 08:00 the next day. This includes 780 minutes of flexibility. According to its charging duration (300 minutes [97]), although its battery can be fully

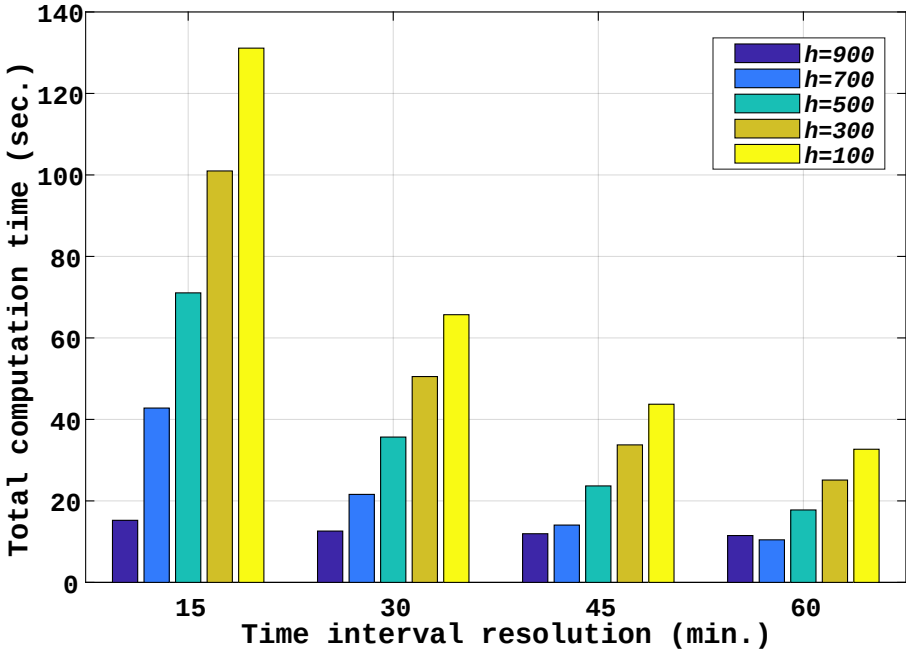


Fig. 6.5: Comparing the trade-off between number of prosumers, time interval resolution, and total simulation time (ECT-P2 with $\mathcal{X} = 40\%$ and EDF mechanism are assumed).

charged until 24:00, however, charging is continued until almost 01:34 the next day.

Table 6.7: Performance analysis of the SALSA applied on 100 prosumers (owning only EVs) against the evaluation metrics [8].

	PDR (%)	PAR	AOD (hrs)	FUR (%)	PCB (%)
Denmark	5.22	3.12	1.34	10.30	3.45
Norway	20.20	2.45	4.40	28.23	3.58
Finland	-11.87	3.89	3.07	20.65	19.53
Sweden	6.63	3.05	1.24	9.40	2.68

For Norway, as Fig. 6.8(b) pictures, PDR is 20.20%. The reason is the low difference ratio 1.29 in electricity prices (see Fig. 6.7). Having an inconsiderable fluctuation in electricity prices, ECTs are relatively low (compared to Denmark), which helps the scheduler shift more charging loads from the first day to the second day. However, this influences AOD, in which EVs are delayed averagely 263.40 minutes to fully charge their batteries. PCB is averagely 3.58%, which is due to the low difference ratio. However, PCB for

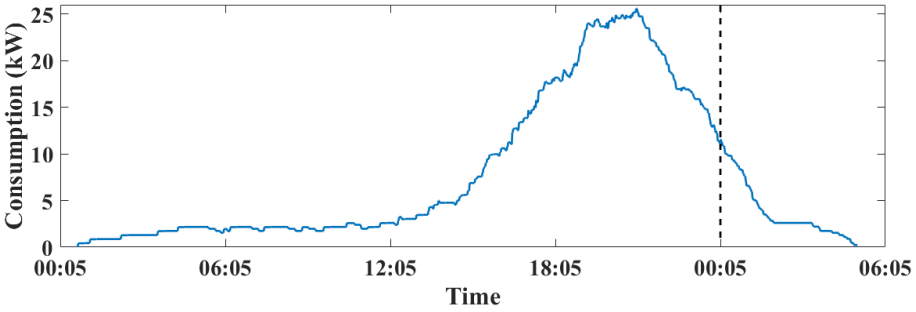


Fig. 6.6: Aggregated charging loads of 100 EVs. Two consecutive days are divided by a vertical dashed line. EVs arrive in the first day with probability of $\mathcal{N}(19, 10)$ and depart the day after with the probability of $\mathcal{N}(7.5, 1)$ [8].

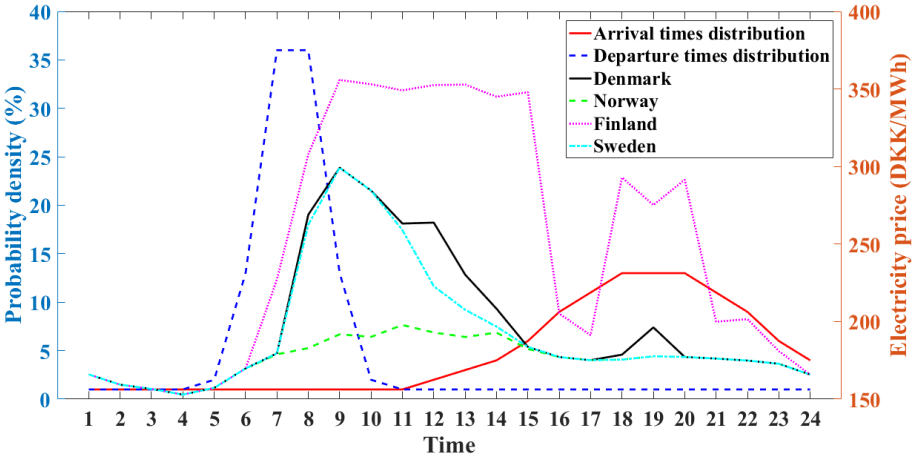
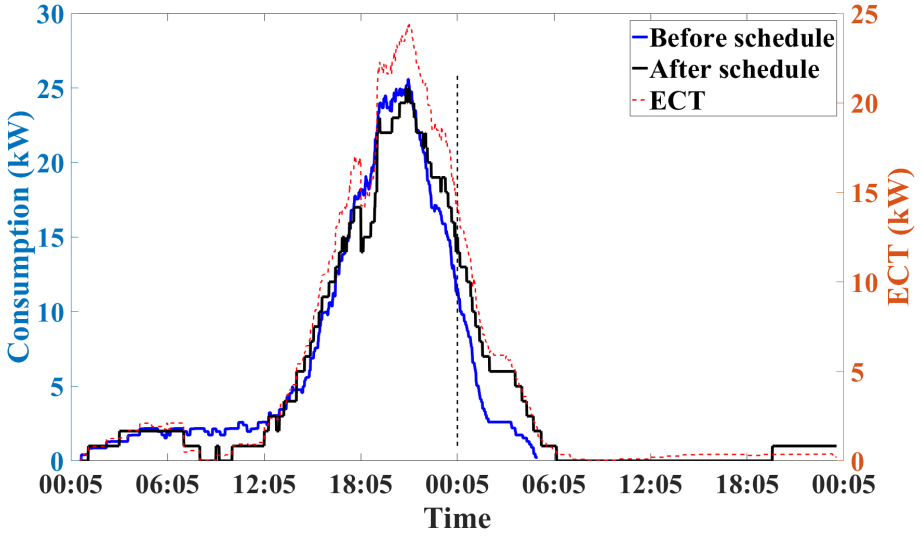


Fig. 6.7: EVs' probability density and hourly basis electricity prices of four Scandinavian countries (May 17, 2016) [8].

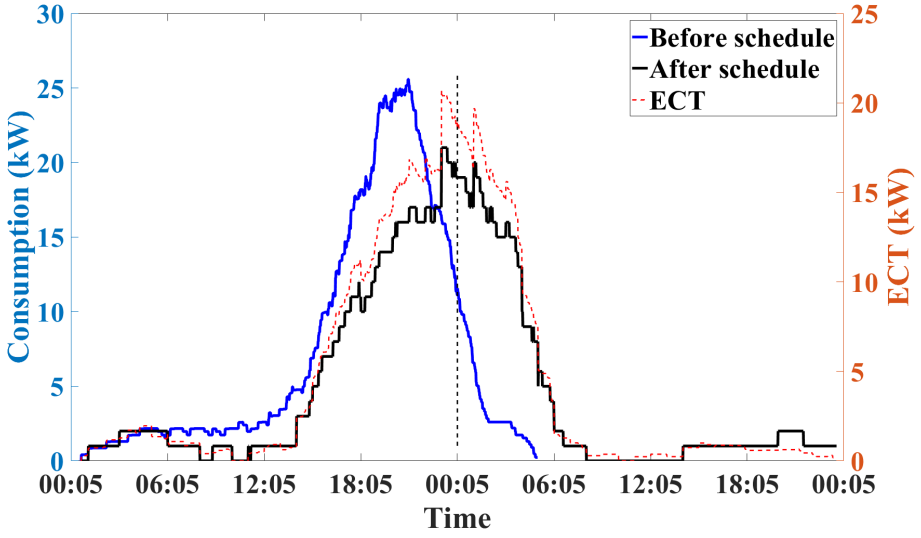
Danish residents is lower compared to Norwegian residents, which depends on PDR.

For Finland, as Fig. 6.8(c) shows, there is a significant trade-off between PCB and PDR. Here, the different ratio is 2.32, which results in 19.53% PCB. This reduction is the result of using 20% FUR meaning averagely 263.40 minutes of AOD. Finland's electricity prices are higher than other discussing countries and more fluctuating during arrival times (see Fig. 6.7). This causes the scheduler not to succeed in shaving the peak, which produces a significant rebound peak with -11.87% PDR. This analysis proves that electricity-price-dependent ECTs significantly behaves differently in these countries.

For Sweden, as Fig. 6.8(d) shows, the evaluation almost equals to the Denmark (ratio: 1.95). However, the small difference in results is caused by a small fluctuation in the electricity prices during arrival times.



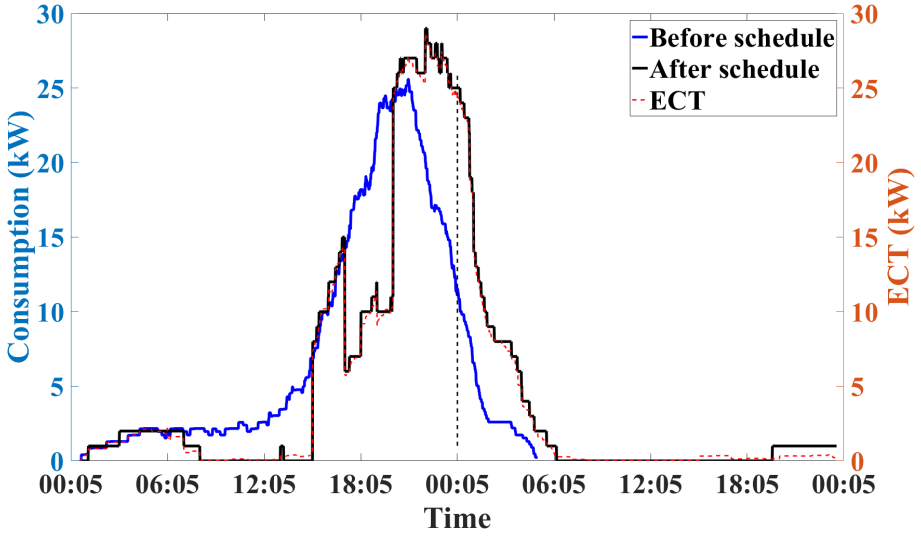
(a) Denmark.



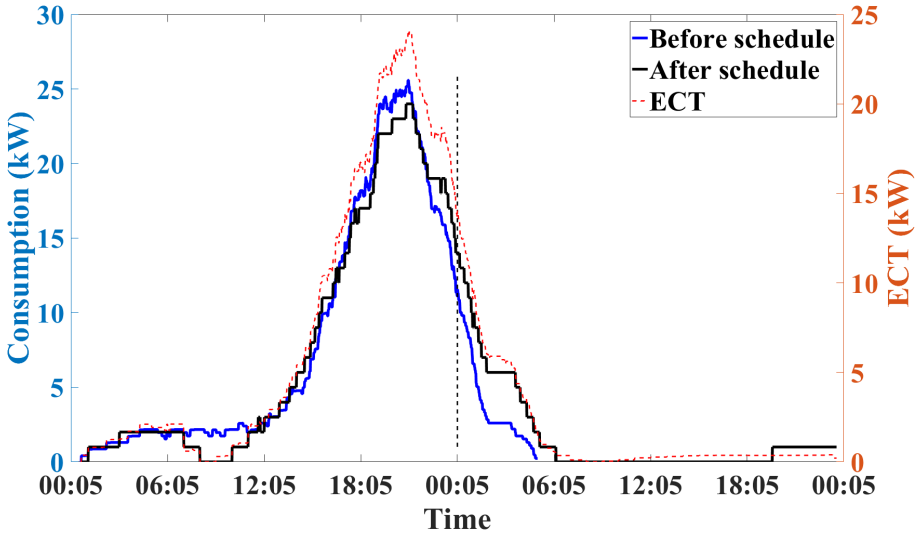
(b) Norway.

Fig. 6.9 shows the distribution of charging EVs over time (reflected by Fig. 6.8). Before scheduling, 61 EVs make the peak demand. PDRs, mentioned in Table 6.7, also represent the reduction in the number of charging EVs. This analysis helps operators configure the grid's structure efficiently.

Fig. 6.10 presents the charging evolution of a Nissan Altra's battery. It arrives at 18:07 on May 17 and departs at 02:32 on May 19. After scheduling, the charging period is split into 571 one-minute slots starting from 17:01 on



(c) Finland.



(d) Sweden.

Fig. 6.8: Aggregated charging loads of 100 EVs before and after applying the SALSA system subject to price-driven ECTs (ECT-P3) [8].

May 18 ending to 02:31 on May 19. From 18:00 to 22:33 on May 18, the battery remains with 20.28% charge.

As the final step, Fig. 6.11 demonstrates how increasing the number of prosumers impacts on the performance. Fig. 6.11(a) shows how FUR and AOD are changed when the number of prosumers increases from 100 to 1000.

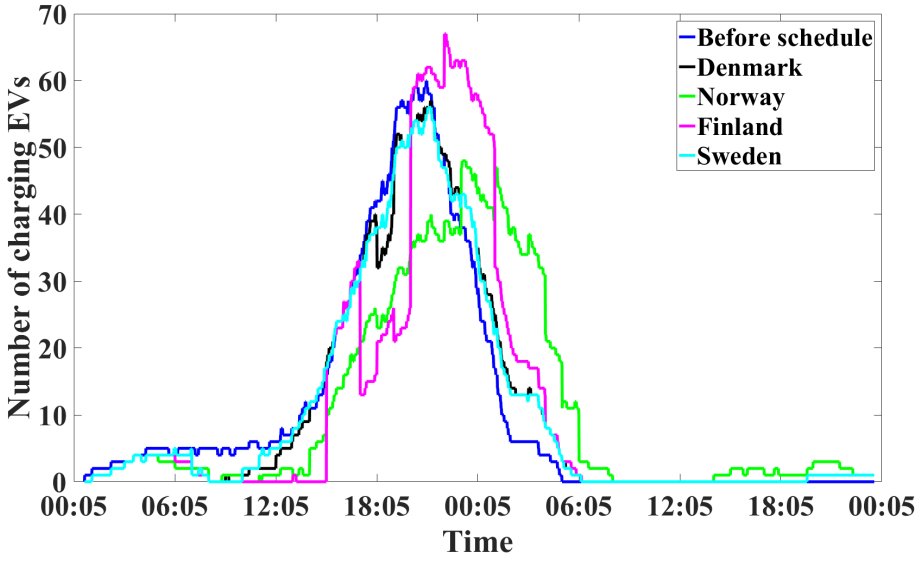


Fig. 6.9: Distribution of charging EVs [8].

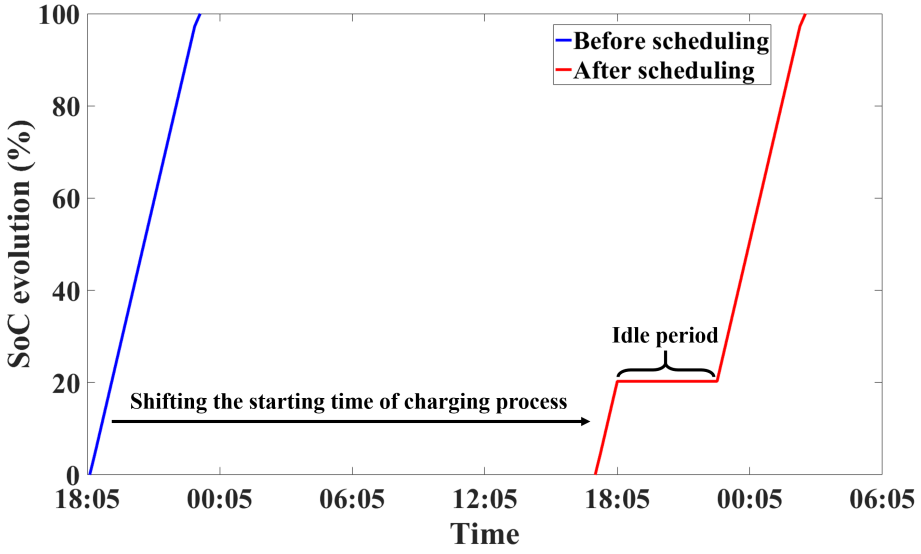


Fig. 6.10: Charging evolution of a Nissal Altra EV [8].

Although FURs and AODs behave reversely, however, the result shows that prosumers benefit more (in terms of AOD) when the DR participation percentage increases. FURs increase since the scheduler uses more flexibility to be able to shave the peak. Nevertheless, since ECTs are higher than the aggregated load consumptions at some intervals (because of previously denied

and newly received events), more events are responded to operate, which consequently, leads to a reduction in AODs.

Fig. 6.11(b) complements this analysis based on PCBs and Maximum Computation Times (MCTs). MCT reports how long it takes for the SALSA system to return decisions in the worst case. For instance, prosumers in Denmark, to receive a response to their load demand requests, averagely wait for 0.28 seconds. The fluctuating nature of PCBs is due to having: 1) high ECTs because of the number of prosumers, and 2) more complexity in selecting an optimal subset of events at each time interval. PCBs do not follow a normal and expected behavior compared to others. The ascending slope of MCTs is a consequence of having more events in the waiting buffer at each time interval. This indeed makes the scheduling algorithm take longer time than anticipated to return decisions. However, taking almost only 4 seconds to schedule 1000 events in the worst case scenario is surprisingly interesting, which confirms the scalability of the algorithm. In Danish electricity grid's infrastructure, averagely 100 prosumers are serviced via a single aggregator (substation). According to Table 6.7, each prosumer will see a delay of 0.28 seconds to receive a response to each event it sends. Since decision making takes place at each sub-station independently, the framework is able to accommodate for a large number of requests in a fair manner.

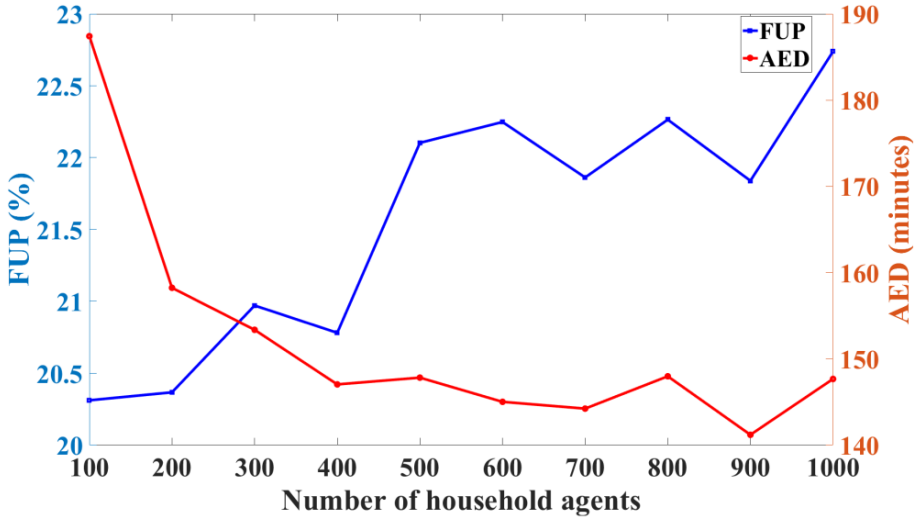
6.2.2.2 Case Study 2

The following parts evaluate and analyze the second case study from different perspectives.

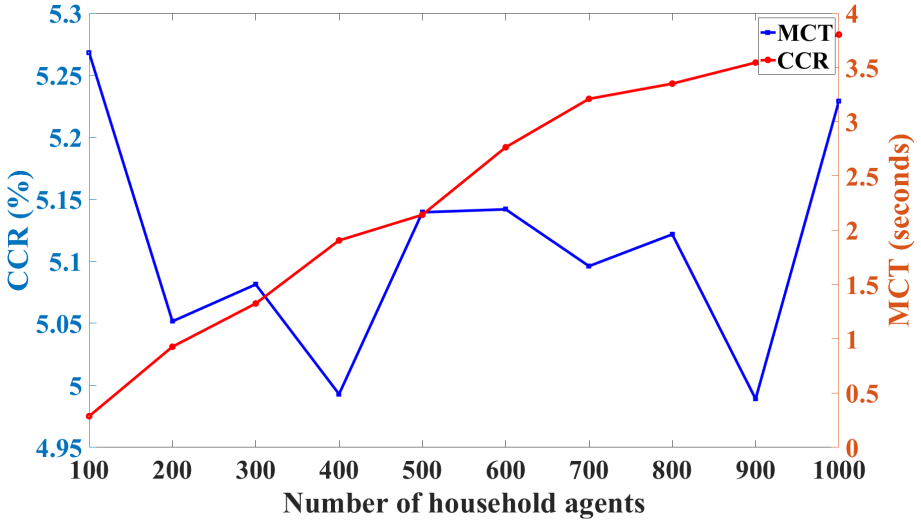
6.2.2.2.1 Impact of Status of Prosumers on the Negotiation

Fig. 6.12 shows the offer package and utility value (unitless, see (5.2) and (5.9)) concession spaces of randomly picked time interval in different circumstances. In Fig. 6.12(a), no PVs and BESSs are considered. The VPP, for example at negotiation iteration $\varrho = 15$, is interested in buying 1630 kW of electric power for 0.0145 \$/kWh. The aggregator, at negotiation iteration $\varrho = 16$, rejects this offer and makes a new one intending to sell 2180 kW of electric power for 0.022 \$/kWh. They continue negotiating until iteration $\varrho = 31$, at which they come to an agreement on exchanging 2000 kW of electric power for 0.016 \$/kWh. Fig. 6.12(b) shows the negotiation process, where all prosumers own PV and BESS. They reach an agreement after exactly 100 negotiation iterations. Having the same utility value of 0.76 at negotiation iteration $\varrho = 65$ does not terminate the process since the VPP provides an offer package with selling 8385 kW of electric power for 0.0227 \$/kWh while the aggregator returns another offer package with buying 4738 kW of electric power for 0.0212 \$/kWh.

Fig. 6.12(c) experiences the same setting as Fig. 6.12(b) does, where negotiators reach an agreement after 71 negotiation iterations. Reasons for having



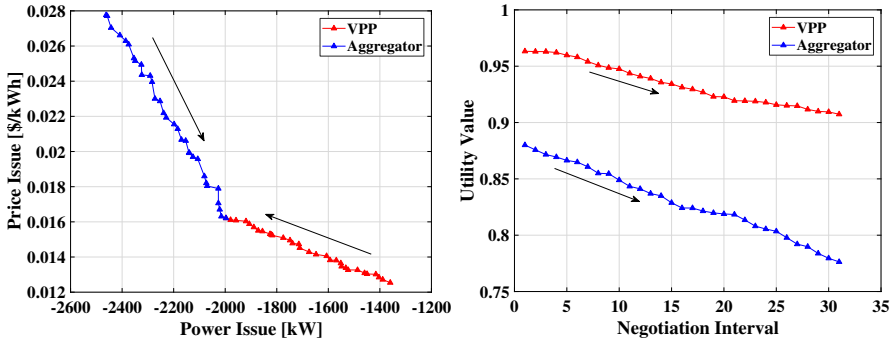
(a) FUR (FUP) and AOD (AED).



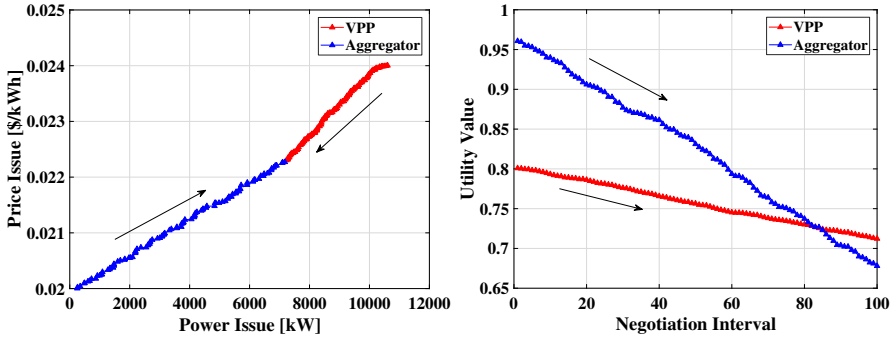
(b) PCB (CCR) and MCT.

Fig. 6.11: Performance analysis based on increasing the number of prosumers [8].

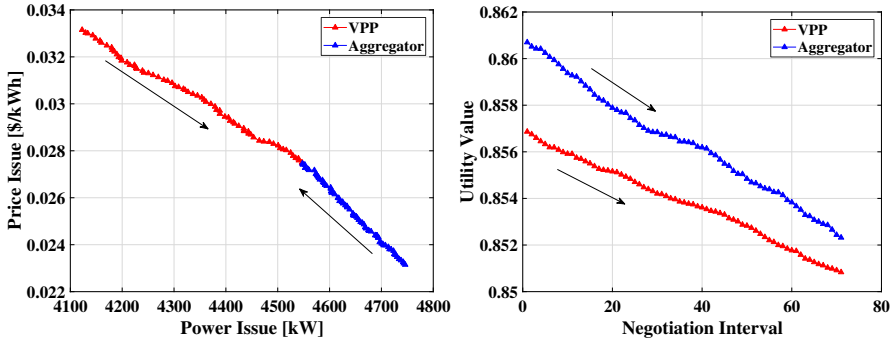
a mixed number of buyer and seller prosumers at this interval are the absence of PV generation (outside of the PV generation period), presence of BESSs with average SOC value of 0.48, and having all refrigerators, 23 dishwashers, 12 newly arrived electric vehicles, and all air conditioners in operation. Therefore, the VPP has to averagely increase the amount of power to sell and decrease the price offer while the aggregator behaves the other way around.



(a) All prosumers are buyers.



(b) All prosumers are sellers.



(c) Prosumers are mixed.

Fig. 6.12: Offer package (left) and utility value concession (right) spaces in different situations. Symbols in the offer package spaces, for the sake of simplicity, represent the average values of columns in the behavior matrices [9].

6.2.2.2.2 Impact of Penetration of Prosumers on the Negotiation Process and the Grid

Fig. 6.13 demonstrates how increasing the number of prosumers influences the computation time and negotiation convergence iteration. Running an individual instance of NSGA-III for each prosumer and the aggregator, where population size and number of generations equal 100, takes approximately 7 seconds. To evaluate the practicality of the negotiation approach employed in the proposed framework with $h = 900$ prosumers, we simulate two setups: i) parallel bilateral negotiations between prosumers and the aggregato (with no VPP), and ii) a single bilateral negotiation between the VPP and the aggregator (introduced here). In the former, CPU and memory usages are 79% and 42 GB, respectively, and reaching agreement at each time interval takes approximately 75 seconds. In the latter, these values for the gateway of each party (each prosumer, the VPP, and the aggregator) are 34.6% (of a single core CPU) and 960 MB, respectively, and the negotiation converges in approximately 39 seconds.

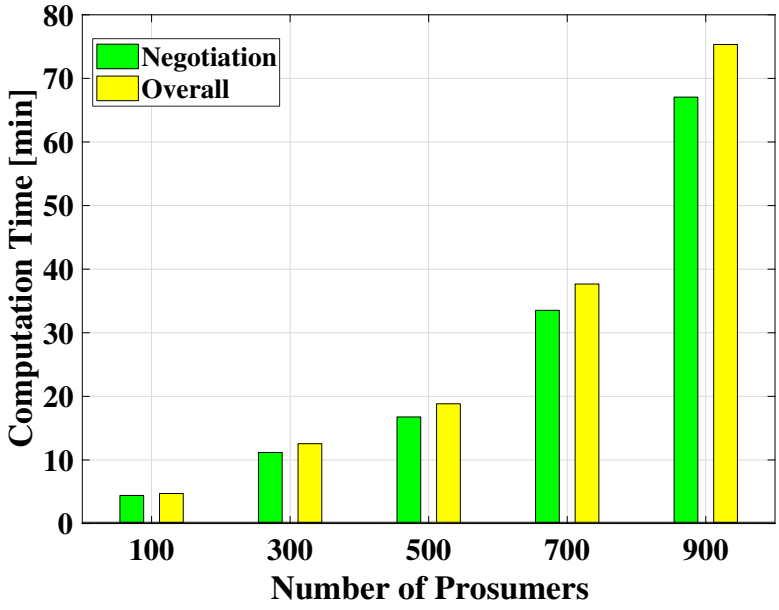
Table 6.8 evaluates the assessment metrics according to different penetration rates of prosumers. These assessment metrics, due to the presence of conflicting objectives in the framework, provide prosumers and the aggregator with trade-offs in making decisions. As the number of prosumers increases, mathematically speaking, the size of the convex feasible offer packages set $Z_{\mathcal{V}}(t), \forall t$ of the VPP also increases (including maximum K^h offer packages). This provides the VPP with more opportunities in utilizing prosumers' flexibilities, which enables it to: i) decrease the delay in satisfying load demands of appliances in average, ii) increase the PCB, and iii) increase the PDR. Increasing rates of SLR and SSR also depend on: i) the generation profiles of PVs in different weather conditions and the BESS capacities, as discussed in the next section, and ii) decrease in FUR.

Table 6.8: Evaluating the assessment metrics according to different penetration rates of prosumers in the grid [9].

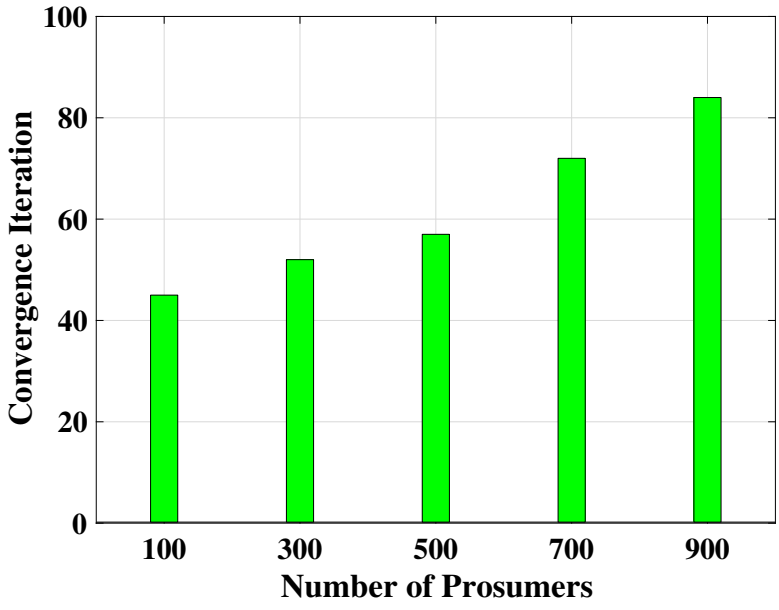
h	PDR (%)	PAR	AOD (hrs)	FUR (%)	PCB (%)	SLR (%)	SSR (%)
100	15.19	2.76	3.30	42.00	65.40	10.10	14.03
300	25.90	2.23	2.57	37.39	99.34	12.47	21.29
500	32.19	2.13	2.29	31.91	147.97	15.08	32.51
700	34.78	2.37	1.43	23.12	179.49	19.17	47.46
900	38.46	2.01	1.13	18.75	209.47	24.45	51.21

6.2.2.2.3 Impact of Penetration of PVs and BESSs on Prosumers and the Grid

Table 6.9 evaluates to which extent the "random distribution of PVs and BESSs" impacts on the values of assessment metrics. Compared to the setting,



(a) Total computation time.



(b) Average negotiation convergence.

Fig. 6.13: Total computation and average negotiation convergence times with different number of prosumers [9].

where all prosumers own PVs and BESSs (see the first row in Table 6.8), here, the grid experience lower PDR since the amount of flexibility is restricted. Decrease in AOD and FUR (due to limited flexibility) increases SLR and SSR (desire to increase the comfort), since the VPP cooperates with the aggregator to increase the PDR and PCB.

Table 6.9: Evaluating the assessment metrics according to the presence of PVs and BESSs in the grid [9].

$\forall \rho_i \in \mathcal{P}$	PDR (%)	PAR	AOD (hrs)	FUR (%)	PCB (%)	SLR (%)	SSR (%)
Only PV	11.43	2.95	1.56	28.18	46.73	20.16	20.16
Only BESS	33.29	2.08	1.20	11.67	35.46	0.83	0.00
Random	16.32	2.60	1.25	12.91	39.97	6.06	6.51

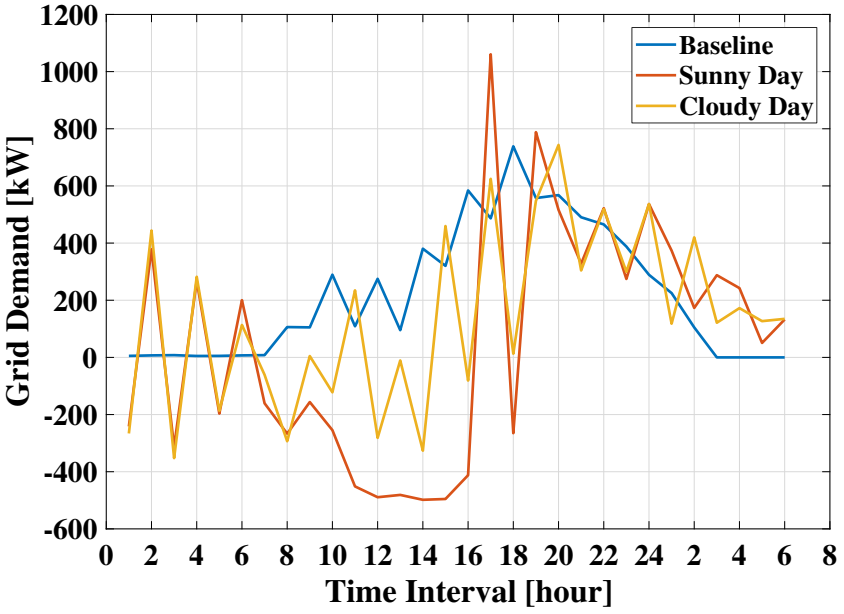
Table 6.10 analyzes the framework in different weather conditions. Obviously, fluctuations in the PV generation limits the VPP, in terms of available flexibility, in negeotiation.

Table 6.10: Evaluating the assessment metrics according to the PV generation profile in different weather conditions [9].

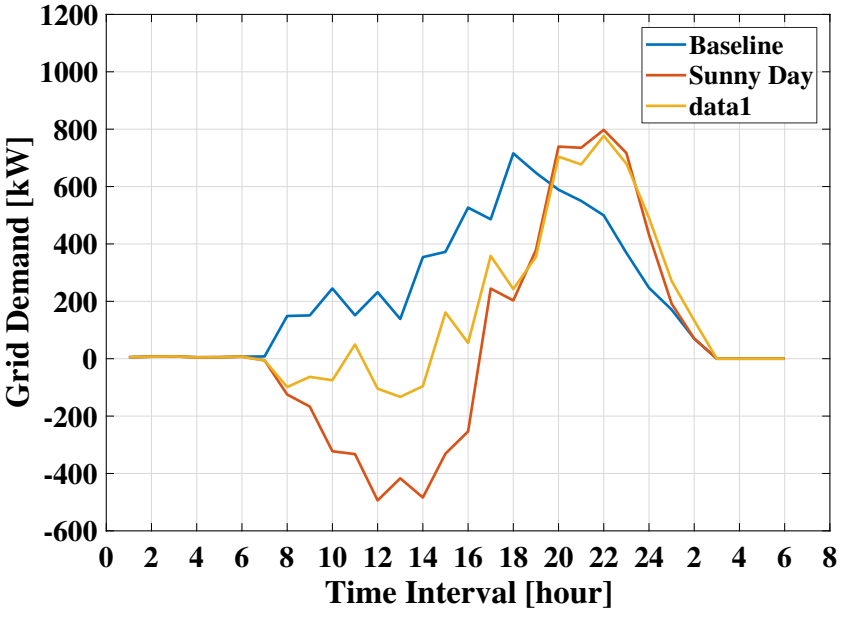
Weather	PDR (%)	PAR	AOD (hrs)	FUR (%)	PCB (%)	SLR (%)	SSR (%)
Sunny	15.19	2.76	3.30	42.00	65.40	10.10	14.03
Cloudy	09.42	2.99	1.33	14.04	37.78	07.15	13.30

Fig. 6.14 shows grid demands for different PV and BESS penetration levels in two sunny and cloudy days. In Fig. 6.14(a), until the time at which PVs start the power generation (i.e., `07:00`), BESSs, by consecutive charging and discharging, try to regulate the grid demand. The grid confronts lower demand fluctuation, when there are only PVs in the system (see Fig. 6.14(b)). However, this setting results in lower PDR for the grid and PCB for prosumers. The reason is that prosumers, due to having no storage flexibility, are unable to provide the VPP with more flexibility. As Fig. 6.14(c) demonstrates, prosumers experience lower AOD and PCB. Similarly, the reason is the very limited amount of flexibility (only consumption flexibility). Table 6.11 evaluates the assessment metrics and Fig. 6.15 shows the average SOCs according to the various BESS capacities. High BESS capacity provides prosumers with: i) more flexibility in storing energy, ii) lower AOD, and iii) higher PCB by selling more to the grid. The VPP, by such increase in the capacity, is able to provide the grid with more flexibility, which in turn, results in having higher PDR. BESSs with different capacities behave dissimilarly after PVs stop generating the electric power (see Fig. 6.15 for `20:00` to `07:00`-next day). The main reason is the arrival of the majority of electric vehicles, which impose higher load demands to the grid compared to other appliances.

Figs. 6.16 and 6.17 picture the generation profile and utilization distribution of a PV and a BESS, respectively. Considering Table 6.3, the prosumer

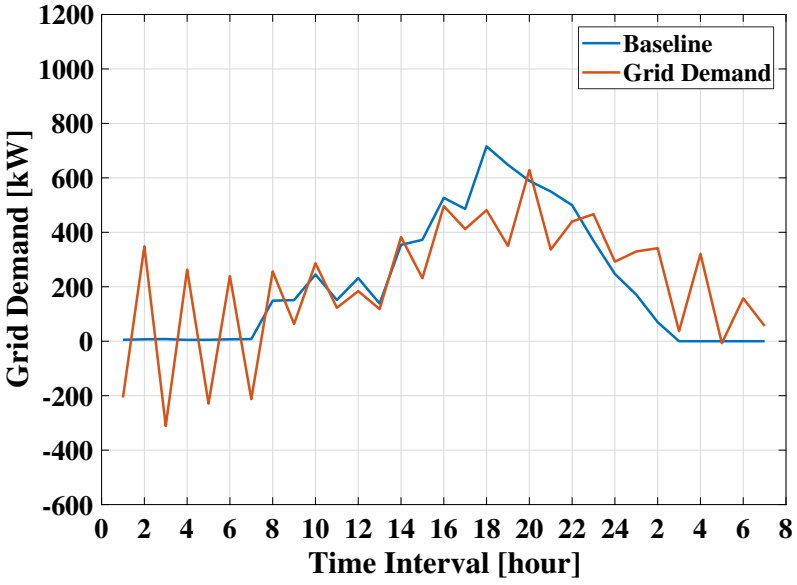


(a) Full penetration of PVs and BESSs.



(b) Full penetration of only PVs.

at `11:00`, `15:00`, and `19:00` endeavors to satisfy load demands of washing machine, laundry dryer, and dishwasher, respectively, with PV generation.



(c) Full penetration of only BESSs.

Fig. 6.14: Grid demand with different penetration levels of PVs and BESSs in two sunny and cloudy days. Baseline shows load demands of appliances [9].

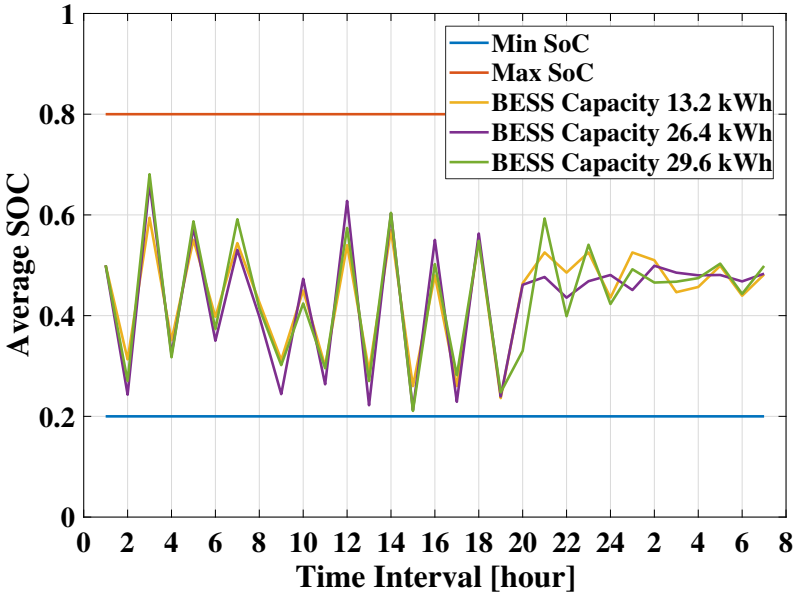


Fig. 6.15: Average SOC of BESSs with different capacities [9].

Table 6.11: Evaluating the assessment metrics according to various BESSs capacities [9].

$B_i^{cap}, \forall \rho_i \in \mathcal{P}$	PDR (%)	PAR	AOD (hrs)	FUR (%)	PCB (%)	SLR (%)	SSR (%)
13.2 kWh	15.19	2.76	3.30	42.00	65.40	10.10	14.03
26.4 kWh	24.36	2.38	2.00	34.68	102.72	14.84	14.66
39.6 kWh	39.62	1.98	0.58	25.36	165.57	18.12	14.34

At other hours, most of the PV generation is sold to the grid. These time intervals are also reflected in Fig. 6.17.

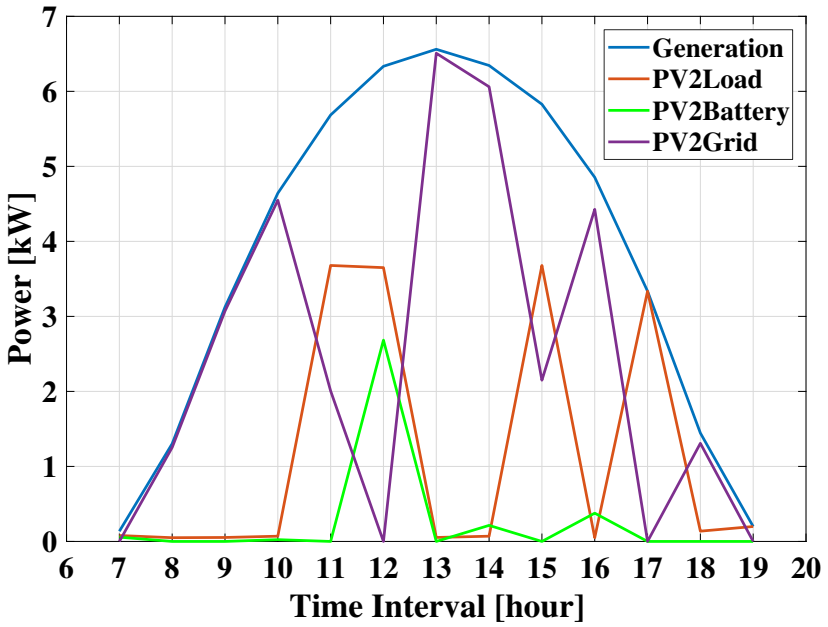


Fig. 6.16: Generation and utilization profiles of the PV of a prosumer [9].

6.2.2.2.4 Impact of Consumption Flexibility of Prosumers on the Negotiation Process and the Grid

Table 6.12 evaluates the assessment metrics, where different sets of appliances are simulated. Considering only a refrigerator for each prosumer yields no PDR and delay due to its non-shiftability feature. Adding more shiftable appliances help prosumers provide the VPP with more consumption flexibilities. This increase has a direct correlation with the AOD and PCB, where prosumers benefit more while waiting for a longer time to receive their appliances in the completed status. A shiftable appliance contributes to the grid's PDR and the prosumer's PCB with averagely 0.1% and 0.37%, respectively.

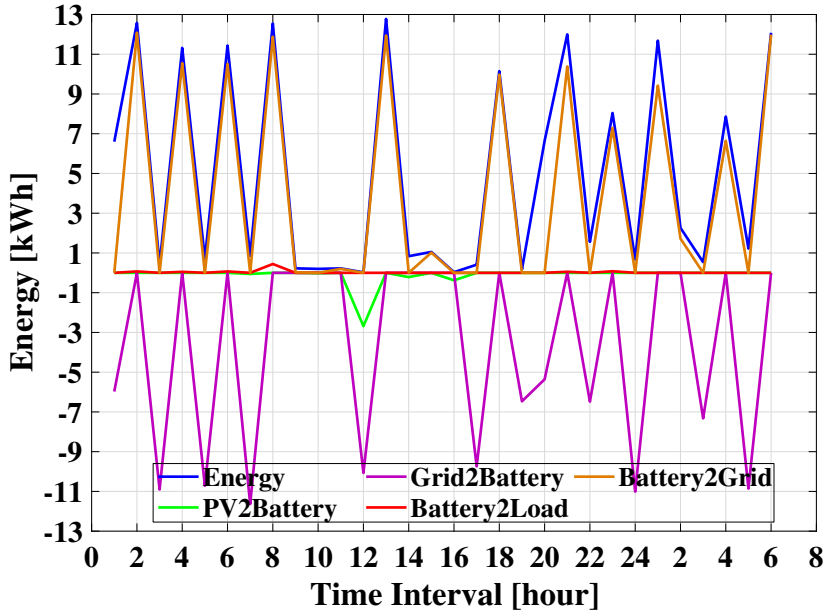


Fig. 6.17: Energy and utilization profiles of the BESS of a prosumer [9].

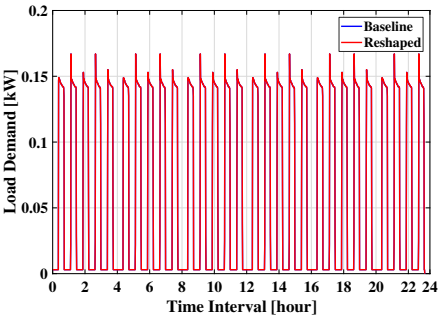
Table 6.12: Evaluating the assessment metrics according to different sets of appliances [9].

$SA_i, \forall \rho_i \in \mathcal{P}$	PDR (%)	PDR	AOD (hrs)	FUR (%)	PCB (%)	SLR (%)	SSR (%)
{RG}	0	3.15	0.00	0.00	6.10	2.94	16.58
{RG, WM}	03.42	3.11	0.14	2.39	10.68	3.32	16.35
{RG, WM, LD}	09.22	3.01	0.58	13.19	26.98	6.20	15.39
{RG, WM, LD, DW}	11.63	2.92	1.37	20.90	34.33	8.06	15.14
{RG, WM, LD, DW, EV}	13.24	2.85	2.26	30.27	49.88	9.44	12.24
{RG, WM, LD, DW, EV, AC}	15.19	2.76	3.30	42.00	65.40	10.10	14.03

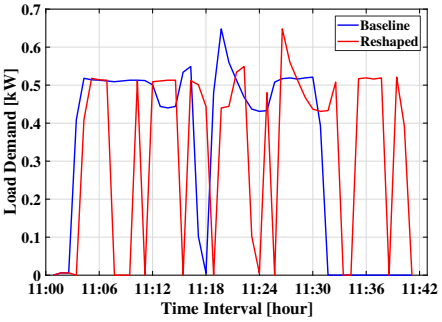
RG: Refrigerator, WM: Washing Machine, LD: Landry Dryer,
DW: Dish Washer, EV: Electric Vehicle, and AC: Heating, Ventilation, and Air Conditioning.

Fig. 6.18 demonstrates the baseline and reshaped load profiles of appliances of a prosumer (only hours in charge). Consecutive fluctuations in the baseline profile of the air conditioner is to keep the temperature constant at 25°C. Air conditioner starts using its temperature flexibility due to the load demand overlap between the laundry dryer (partly), dishwasher, and the electric vehicle. For example, the air conditioner between `19:20` and `19:40` attempts to increase the temperature since the laundry dryer has just finished operating and the operation of dishwasher has been interrupted.

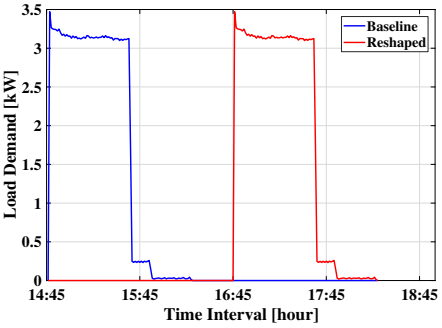
Fig. 6.19 shows the hourly benefit/cost of prosumers with respect to real-time electricity prices. The baseline points to the case, where there are no PVs and BESSs simulated. One prosumer, for instance, to satisfy its load demands without any PV and BESS, has to daily spend (-)$2.59 while holding



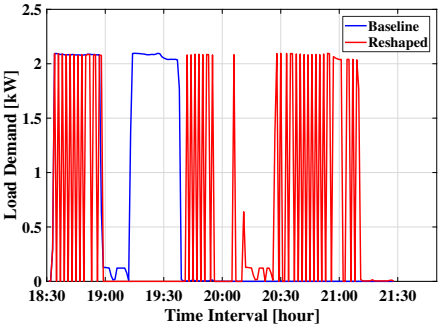
(a) Refrigerator (AOD=0 min. and FUR=N/A).



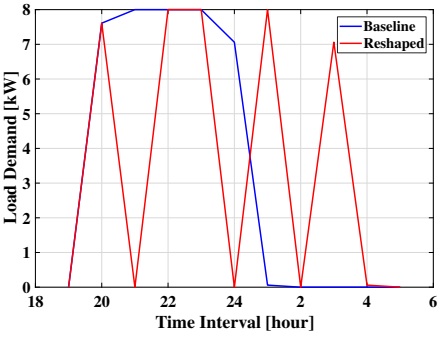
(b) Washing machine (AOD=11 min. and FUR=2.57%).



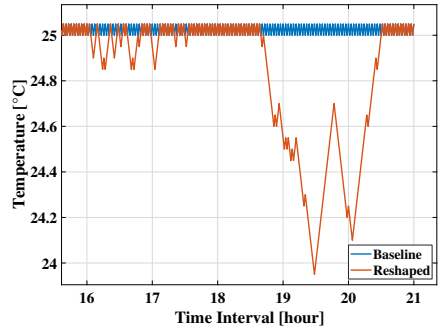
(c) Laundry dryer (AOD=119 min. and FUR=99.16%).



(d) Dishwasher (AOD=92 min. and FUR=33.57%).



(e) Electric vehicle (AOD=294 min. and FUR=59.63%).



(f) Air conditioner (temperature).

Fig. 6.18: Baseline and reshaped load profiles of appliances of a prosumer [9].

such equipment results in making a benefit of $2.78. Therefore, according to (6.5), PCB for this prosumer equals to 207.41%. The results confirm that the prosumers are interested in buying less from the grid, when the electricity

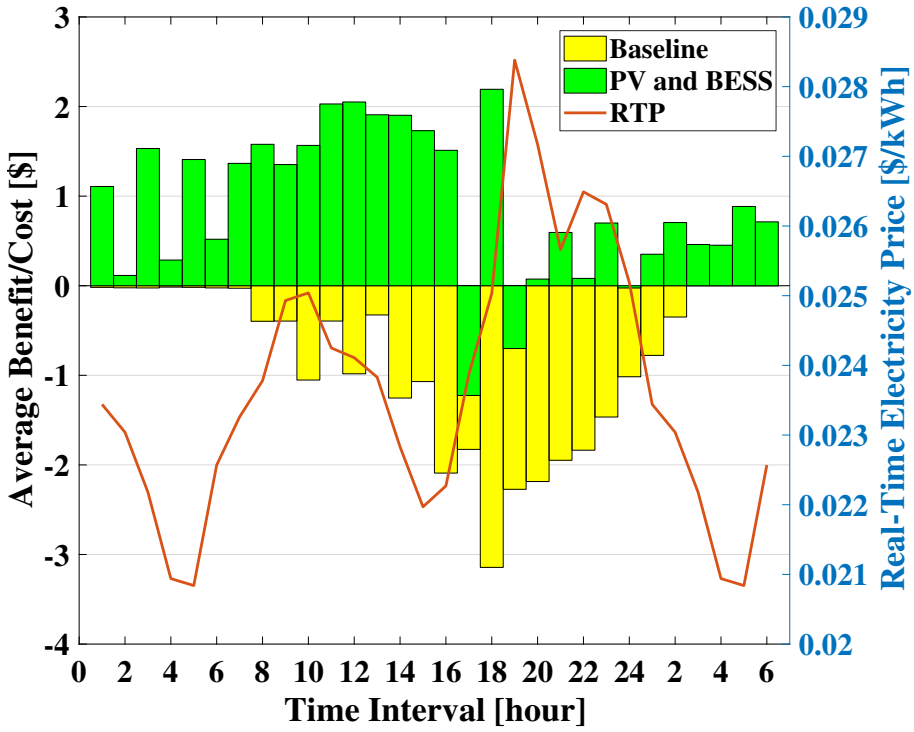


Fig. 6.19: Hourly benefit/cost of prosumers [9].

prices are relatively high (see `18:00` to `24:00`). The reason for buying power from the grid at `17:00` is the low PV generation, the start time of majority of dishwashers, and the arrival of some of electric vehicles. The VPP at the next hour changes its behavior and compensates the cost imposed at the previous hour. However, since PVs stop generating at `19:00`, prosumers have to buy from the grid since the majority of electric vehicles arrive and intend to charge immediately. From then on, prosumers, to satisfy their load demands, utilize their BESSs while trying to sell the surplus energy to the grid simultaneously.

6.2.2.2.5 Scalability of the SALSA

The research, conducted during PhD study, has been in collaboration with the SEMIAH project. SEMIAH, to determine the potential performance of the impact of a scalable number of prosumers on the grid, has promised to deliver a scalable simulation infrastructure for the aggregation of 200.000 prosumers. The deployment of SEMIAH in these households would allow the shifting of 90 GWh/year of electrical consumption from fossil fuels to RESs, thereby, reducing the gap between RES produced and consumed [3].

In order to validate and confirm the scalability of the SALSA system for the SEMIAH, 1000 aggregators, each of which serving 200 prosumers, are instantiated. Time resolution for running the SALSA is set to 15 minutes. ECT-P2 with percentage of 60% is used. Fig. 6.20 demonstrates the computation time of scheduling a scalable number of prosumers with the SALSA.

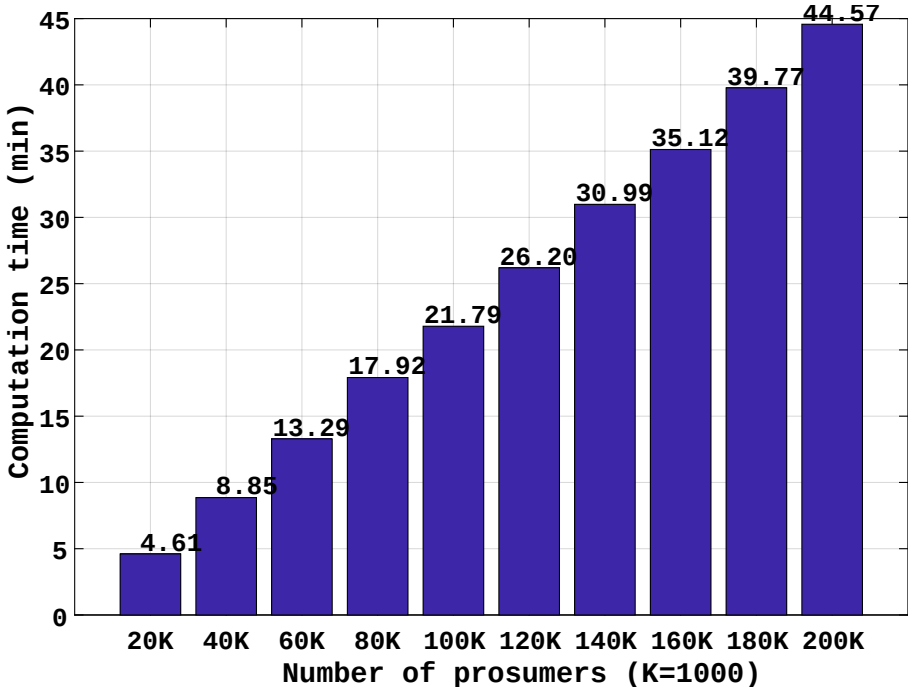


Fig. 6.20: The computation time of scheduling a scalable number of prosumers with the SALSA.

Referring to simulation environment listed in Table 6.1, once the simulation starts, Matlab picks 16 random aggregators and assigns each aggretgator to one CPU. Then, each CPU schedules its own prosumers. Once a CPU finishes scheduling, another random aggregtor is assigned to it. Therefore, at maximum, $16 \times 200 = 3.200$ prosumers are scheduled simultaneously. In fact, a number of back-and-forth messages in each simulation per CPU is $200 \times 1.538 \times 15.396 = 153.600$, where "8" is the total of six appliances (listed in Table 6.3) with a PV, and a BESS. Also, "96" is denoted as the number of intervals during a 24-hour day. Therefore, at each time, at which 16 CPUs are running the scheduling simultaneously, $16 \times 153.600 = 2.457.600$ messages are transferred. Three main parameters influence the simulation time: i) number of aggregators, ii) number of prosumers per each aggregator, and iii) the time resolution. By decreasing the number of aggregators and increasing the number of prosumers at the same time, the SALSA moves further towards the complete decentralized solution. For instance, let us consider

there are 16 aggregators and 12.500 prosumers per each substation. In the simulation, each aggregator is assigned to each CPU. When the simulation starts, it runs only once since $16 * 12.500 = 200.000$. Therefore, 200.000 households are scheduled simultaneously in a completely decentralized way. The interesting achievement here is that by fluctuating the time resolution the total simulation time also changes. As a result, there is a three-dimensional trade-off between these three parameters.

6.2.2.2.6 Network Complexity of the SALSA

To analyze the network complexity of the SALSA in a realistic scenario, consider a world with $h = 20.000$ prosumers interacting with one aggregator (no VPP is assumed). The size of the private type of a typical prosumer with six appliances, one PV, and one BESS is approximated to 16 kB at most. As for the communication network topology, current last-mile technologies include wireless 4G LTE and fiber optic cables. Nevertheless, as a worst case scenario, the former is considered. The wireless 4G LTE technology claims 1 Gb/s (125 Mb/s) peak data rate shared among all prosumers within a cell; which it is assumed to have the capacity of 200 active prosumers (i.e., one cell per 200 active prosumers). Note that time resolution is set to 15 minutes.

One message of size 16 kB is sent over one two-way communication channel by each prosumer to the aggregator, which take approxiamtely 3 seconds. The highest (and only) computational burden is at the aggregator node, which has polynomial time complexity and requires an execution time of approximately 28 seconds.[1] The highest required storage at the aggregator node is also $0.016 \times 20.000 = 320$ MB.. Then, the aggregator returns individual responses to prosumers in one message over each two-way communication channel. The time required to send all the 320 MB messages back to the agents in each round would be (320 MB/message$\times$200 message/cell)/125 MB/s$\approx$ 512 s/cell.

In summary, assuming negligible latency in a 4G LTE network, the whole process can be achieved in approximately (3 sec.+ 28 sec.+512 sec.)/60 sec/min = 9.05 minutes which is not an unreasonable computation time in a realistic setting with 20.000 prosumers.

[1] The simulation results on the hardware listed in Table 6.1 are just an example for instantiation purposes. For actual systems, the running time would of course be different but as well as the hardware.

Chapter 7

Web Applications

This chapter describes the web applications developed during the PhD studies. It includes two sections: i) the first section proposes an interactive web service for the SALSA system, and ii) the second section provides a brief description about an interactive "Microgrid Online Training Center" developed during the six-month stay at UCLA.

7.1 SALSA Web Service

The web application of the SALSA, developed for the SEMIAH project, is accessed by browsing salsa.semiah.eu. Fig. 7.1 shows a snapshot of this application. As Fig. 7.2 pictures, navigating to "Simulation Case Study" part on the left side, the case study, entitled "SALSA: A Formal Hierarchical Optimization Framework for Smart Grid," appears.

By clicking on the "Grid Information" button, as Fig. 7.3 shows, the "Population" button enables the modification of the set of *prosumers* and *aggregators* in the grid while the "Appliance Set" button enables the incorporation of various appliances and their penetration rates to prosumers. For the sake of simplicity, only one aggregator is allowed to be instantiated. For each appliance, e.g., "Refrigerator," it is possible to enable/disable its availability and define its penetration rate accordingly. Once the "Number of Prosumers" is defined, more simulation settings appear, as Fig. 7.4 depicts. Each appliance has a specific setting, as Fig. 7.5 illustrates. Referring to the "Scheduling Information," it defines the ECT policy ("Grid Constraints") and "Scheduling Time Resolution," as Fig. 7.6 shows.

By clicking on the "Start Scheduling button, a JavaScript code running in the background forwards the information to the Matlab installed in the server. If the "individual household analysis," is checked prior to starting the scheduling, a simulation analysis for the household of each prosumer is made and shown afterwards. Once scheduling starts, the screen is locked until the

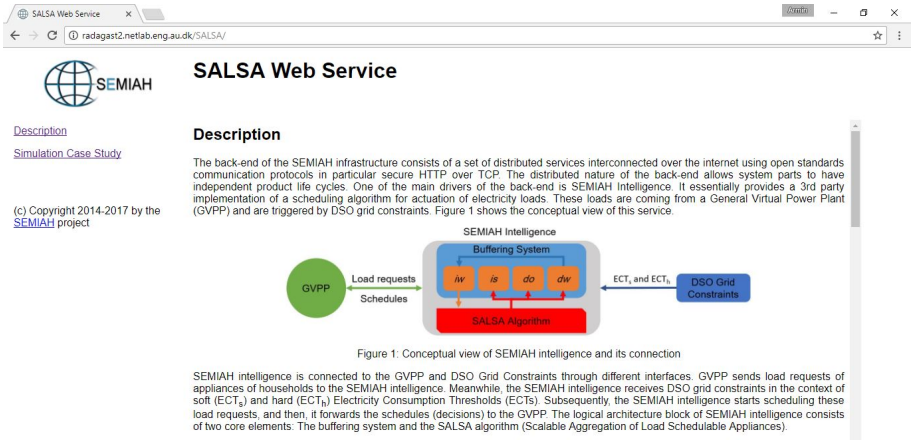


Fig. 7.1: Snapshot of the web application of the SALSA.

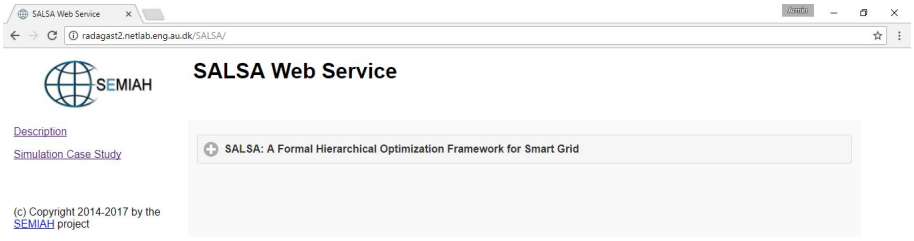
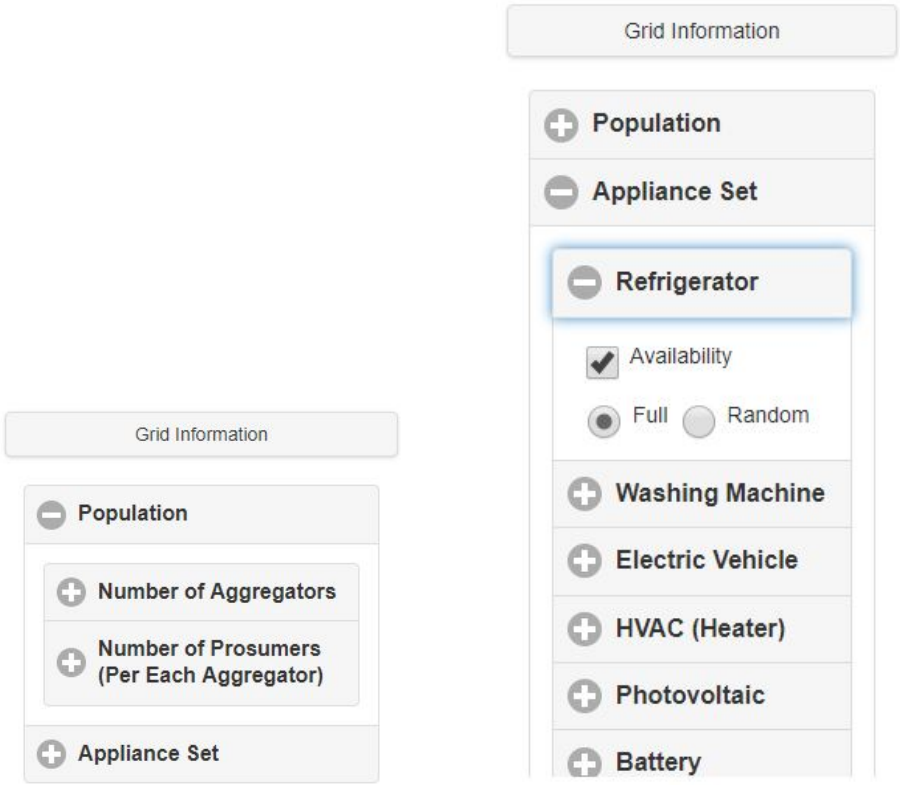


Fig. 7.2: Snapshot of the case study developed for the SALSA during the three-year PhD study.

Matlab returns the results. Waiting time depends on the number prosumers, appliance set, and time resolution adjusted. Results include the evaluation of different economic and environmental criteria along with the Excel file of schedules of the whole grid and each of households. Fig. 7.8 shows scheduling results of two prosumers with full penetration of all appliances. It is also possible to download the Excel file of schedules, as Fig. 7.9 shows its snapshot (both the aggregated and individual versions). This file includes five columns. The first column lists the time intervals. Second column lists aggregated load demands before scheduling. Third column lists aggregated load demands after scheduling, which have been satisfied through the grid. Fourth column lists aggregated load demands after scheduling, which have been satisfied through PVs installed in households. Finally, fifth column lists aggregated load demands after scheduling, which have been satisfied through BESSs installed in households. Fig. 7.10 shows an example of results for one household.



(a) Population. **(b)** Appliance Set.

Fig. 7.3: Snapshot of adjusting the "Grid Information."

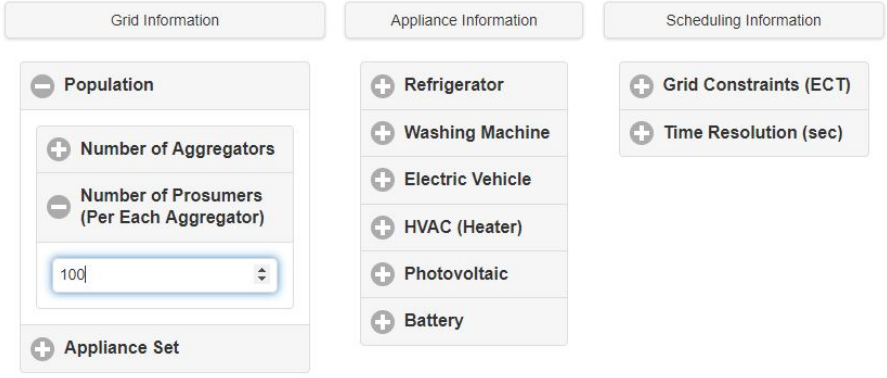
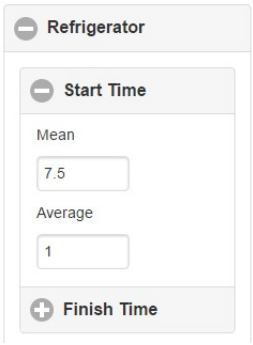
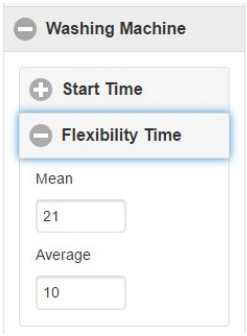


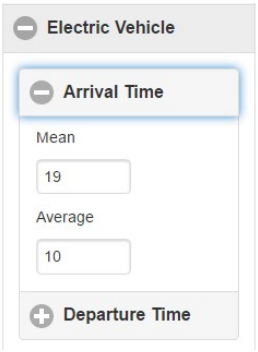
Fig. 7.4: Snapshot of different scheduling setting after defining the "Number of Prosumers."



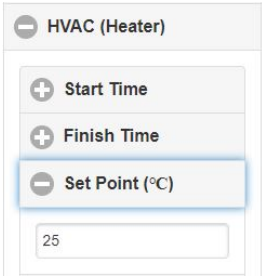
(a) Refrigerator.



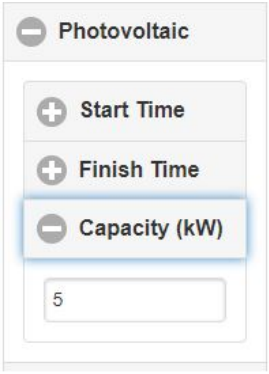
(b) Washing Machine.



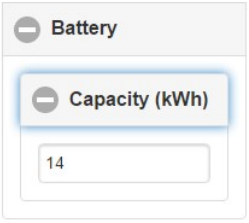
(c) Electric Vehicle.



(d) HVAC.

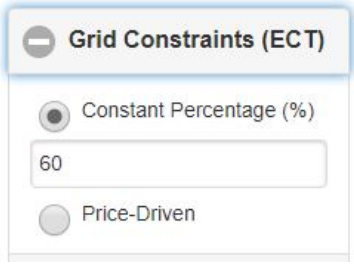


(e) PV.

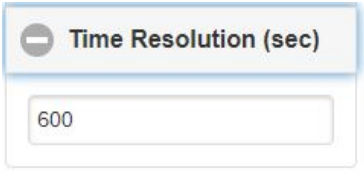


(f) BESS.

Fig. 7.5: Snapshot of the specific setting for each appliance.



(a) ECT Policy (1&3).



(b) Time Resolution.

Fig. 7.6: Snapshot of adjusting the "Scheduling Information."

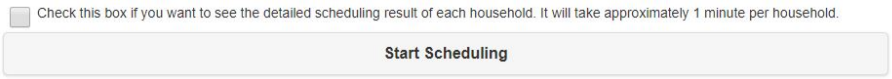
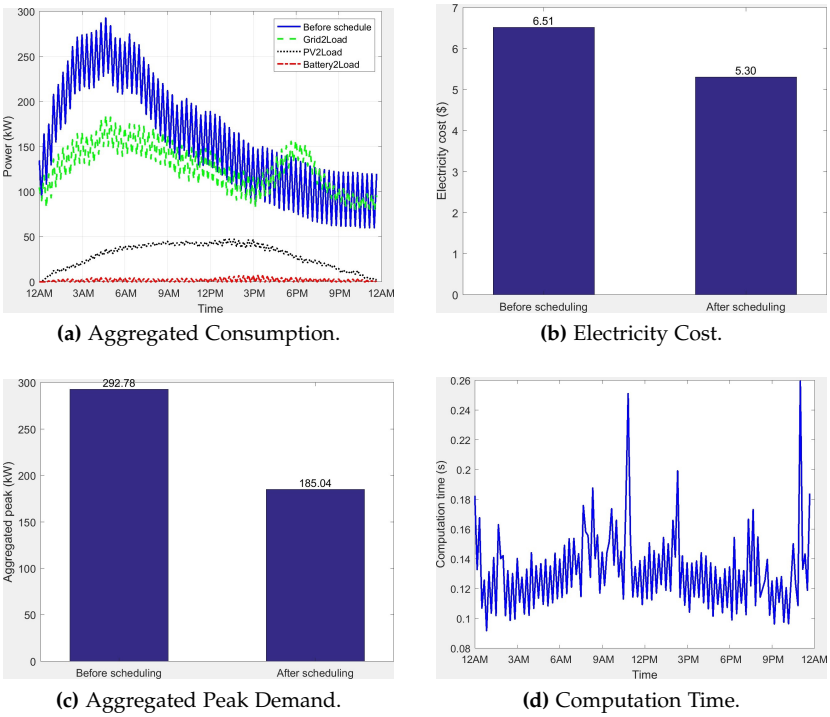


Fig. 7.7: Snapshot of the "Start Scheduling" button.



(a) Aggregated Consumption. **(b)** Electricity Cost.

(c) Aggregated Peak Demand. **(d)** Computation Time.

Fig. 7.8: Performance of the SALSA from grid's point of view according to different criteria.

7.2 Microgrid Online Training Center

The web address for the developed web application has been excluded due to data export regulations in United States. Nowadays, the demand for electric vehicles is increasing. Although this will add new features to the distribution grid management, however, it can also bring challenging issues at the same time. The "Microgrid Online Training Center" is a user-interactive web service that is designed to help non-expert users understand the energy flow, timing, and performance of a microgrid. It simulates a microgrid integrated with three different components: PVs, BESSs, and Electric Vehicle Supply Equipments (EVSEs). For information about the components, the reader is referred to Sections 4.1.2, Section 4.1.3, and [98], respectively. Fig. 7.11 shows a snapshot of the Microgrid Online Training Center.

	A	B	C	D	E
1	Time	Before Schedule (kW)	After Schedule-Grid2Load (kW)	After Schedule-PV2Load (kW)	After Schedule-Battery2Load (kW)
2	10-Jan-2017 00:00:00	2.4	1.8	0	0
3	10-Jan-2017 00:10:00	1.2	1.2	0	0
4	10-Jan-2017 00:20:00	2.4	1.8	0	0
5	10-Jan-2017 00:30:00	1.2	1.8	0	0
6	10-Jan-2017 00:40:00	2.4	1.8	0	0
7	10-Jan-2017 00:50:00	1.2	1.8	0	0
8	10-Jan-2017 01:00:00	2.4	1.8	0	0
9	10-Jan-2017 01:10:00	1.2	1.8	0	0
10	10-Jan-2017 01:20:00	2.4	1.8	0	0
11	10-Jan-2017 01:30:00	1.2	1.8	0	0
12	10-Jan-2017 01:40:00	2.4	1.8	0	0
13	10-Jan-2017 01:50:00	1.2	1.8	0	0
14	10-Jan-2017 02:00:00	2.4	1.8	0	0
15	10-Jan-2017 02:10:00	1.461	1.8	0	0
16	10-Jan-2017 02:20:00	2.625	2.061	0	0
17	10-Jan-2017 02:30:00	1.443	1.8	0	0
18	10-Jan-2017 02:40:00	2.603	2.025	0	0
19	10-Jan-2017 02:50:00	1.2	1.8	0	0
20	10-Jan-2017 03:00:00	2.4	2.043	0	0
21	10-Jan-2017 03:10:00	1.2	1.8	0	0
22	10-Jan-2017 03:20:00	2.4	2.003	0	0
23	10-Jan-2017 03:30:00	1.2	1.8	0	0
24	10-Jan-2017 03:40:00	2.4	1.8	0	0
25	10-Jan-2017 03:50:00	1.2	1.8	0	0
26	10-Jan-2017 04:00:00	2.4	1.8	0	0
27	10-Jan-2017 04:10:00	1.2	1.8	0	0
28	10-Jan-2017 04:20:00	6	2.4	0	0

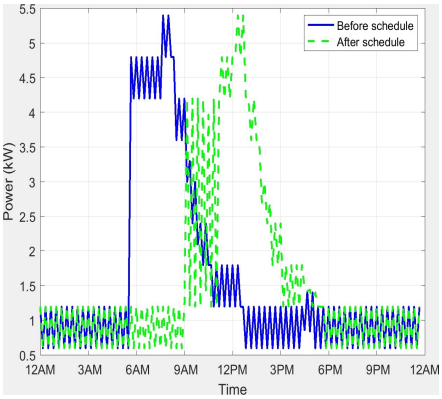
Sheet 1

Fig. 7.9: Snapshot of the Excel file of schedules.

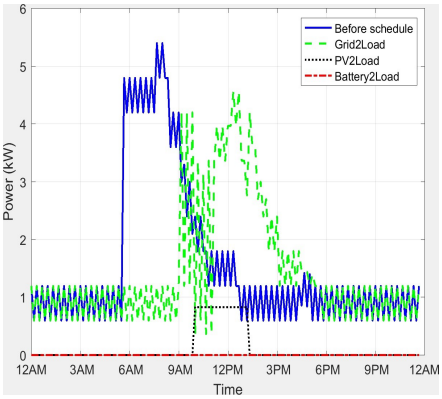
These components are managed through a *Control System*. The overall goal of the control system is to reduce the EVSE demand by optimally managing energy fed from the PV and BESS. The control system, to satisfy the EVSE demand, will try to utilize the energy from the PV and BESS (if possible). This applications employs two variations of the SALSA system, named "Real-Time" and "PV Power Integration." The first one attempts to satisfy the EVSE load consecutively over time while the second one shifts all demands until a certain percentage of PV peak production is reached. Lastly, to meet the remaining EV demand, it would utilize the energy from the grid. The control center allows users to adjust parameters of each component. Once parameters are entered, the back-end of the Online Training Center will start simulating, and then, displaying simulation results through a user-interactive plot. The user can study the impacts of such components on the microgrid according to different simulation parameters. The overall goal of the Online Training Center is to provide users with visual simulation results to have a fundamental understanding about microgrid. Fig. 7.12 shows the control center configuration. Fig. 7.13 shows the simulation result. PV capacity is 10 kW. A 10kWh BESS is assumed to discharge and charge once over a day. Charging is done during the midnight when electricity prices and peak demand are low. Its minimum and maximum SOCs are set to 10% and 100%, respectively.

7.3 Conclusion

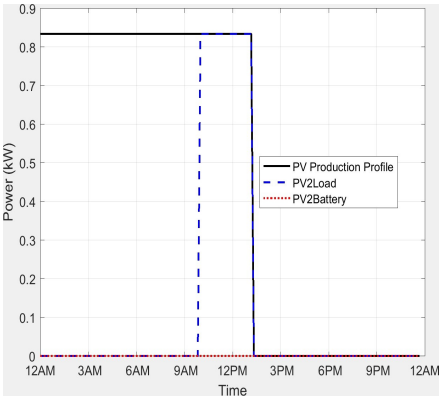
This chapter has led to the following Application Level Contribution:



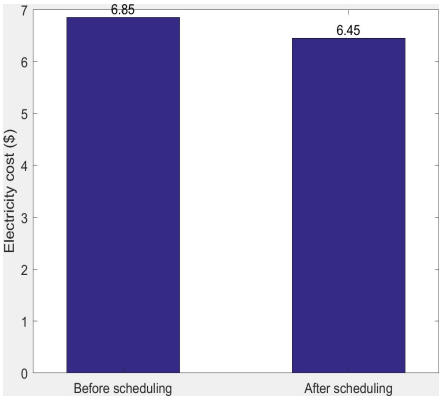
(a) Aggregated Consumption.



(b) Resource Usage.

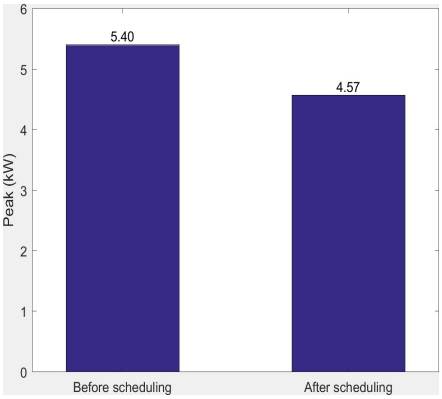


(c) PV Profile and Usage.

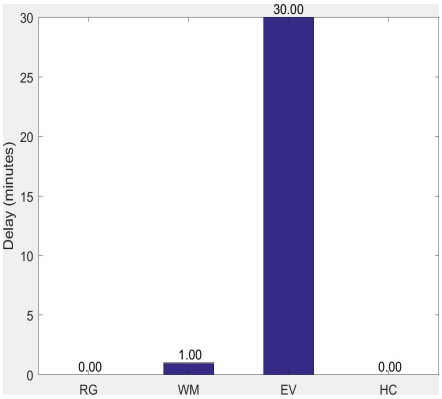


(d) Electricity Consumption Cost.

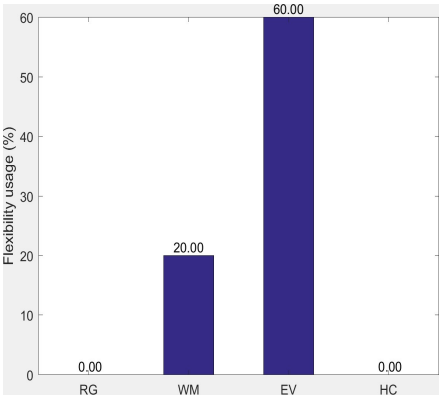
ALC-7: Develop and implement two interactive web services for: i) the SALSA system, which is accessible at salsa.semiah.eu, and ii) a microgrid online training center hosted at UCLA SMERC.



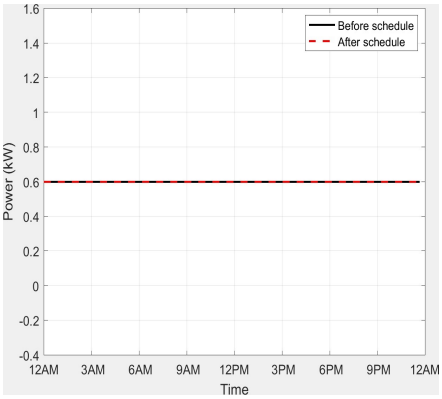
(e) Peak Consumption.



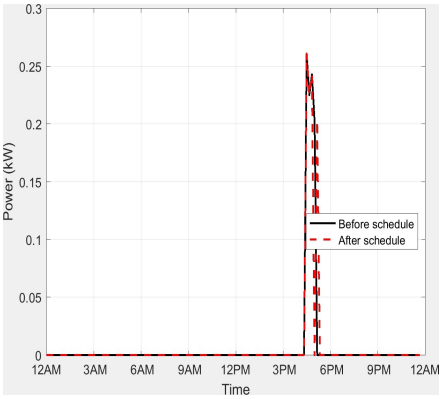
(f) Appliance Serving Delay.



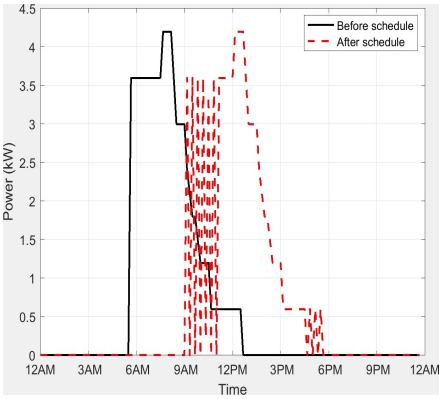
(g) Appliance Flexibility Usage.



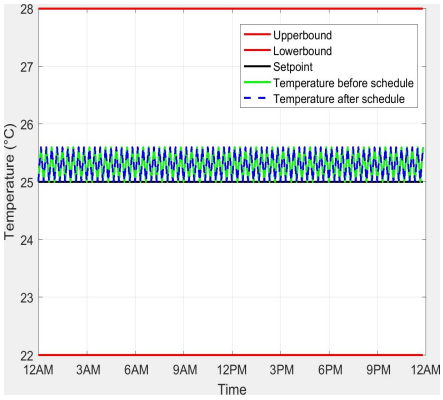
(h) Refrigerator Profile.



(i) Washing Machine Profile.



(j) Electric Vehicle Profile.



(k) House Temperature.

Fig. 7.10: Performance and impact of the SALSA on a prosumer's daily consumption.

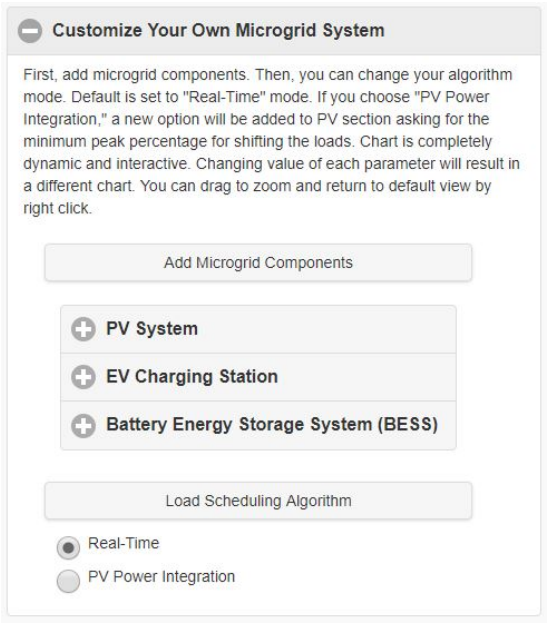


Fig. 7.11: Microgrid Online Training Center.

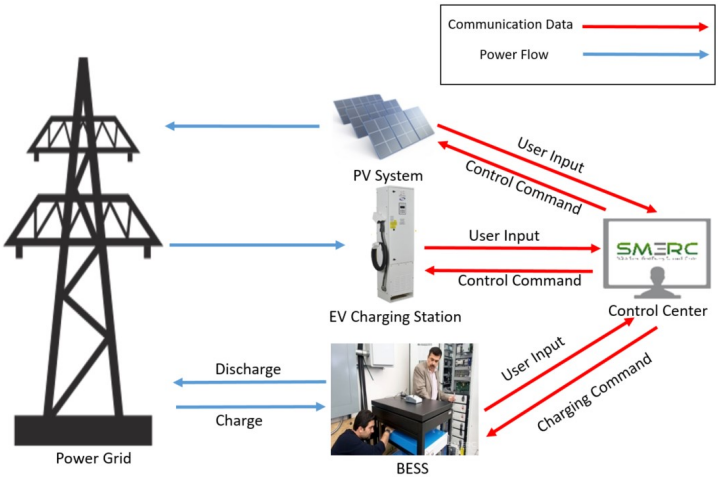
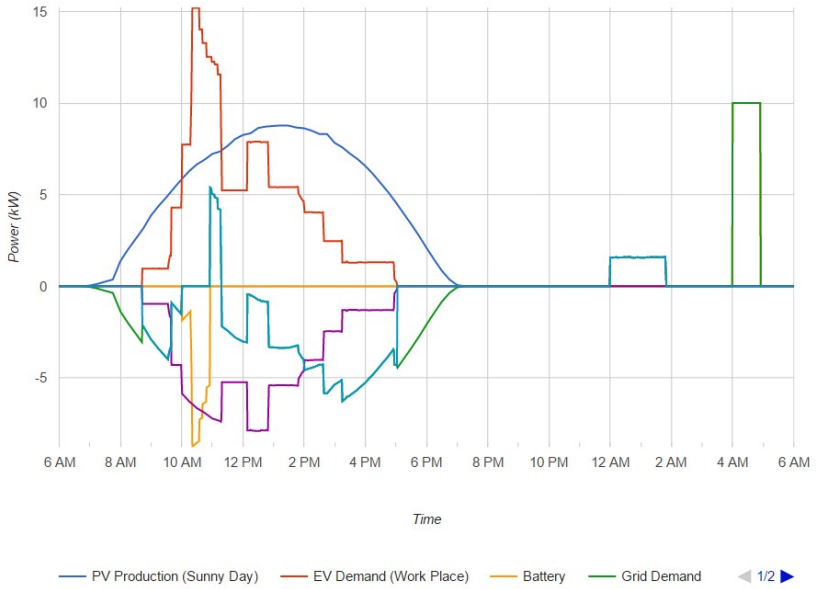
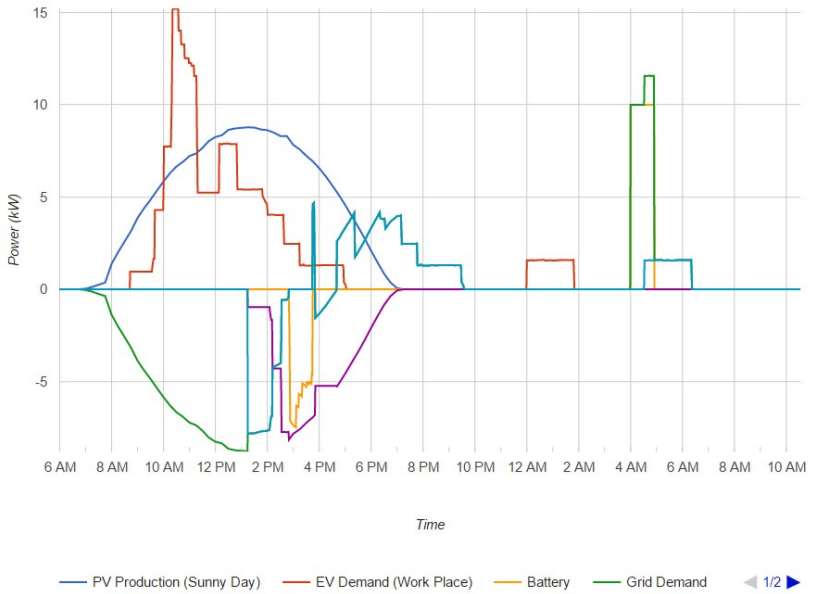


Fig. 7.12: Control Center Configuration.



(a) Real-time algorithm.



(b) PV power integration algorithm (Maximum peak percentage for shifting is 100%).

Fig. 7.13: Grid's status using different algorithms.

Chapter 8

Conclusions and Future Work

This chapter concludes the summary part of this dissertation and presents an overview of the contributions. It is finalized with a discussion on future work.

8.1 Contributions Overview

The research hypotheses of the dissertation, as described in Section 1.2, have been addressed with 10 scientific contributions. Each of these contributions has been presented in publications, as shown in Table 8.1. The table shows the level that each publication has supported each contribution: Low (L), Medium (M), or High (H). Note that all of these 10 contributions have mainly been addresses in this dissertation with a high level.

The contributions fall in the following categories according to three research hypotheses: i) formal framework for smart grid applications, which is addressed in Chapter 3, ii) the SALSA system, which is addressed in Chapters 4 and 7, and iii) the bilateral multi-issue negotiation approach, which is addressed in Chapter 5. Fig. 8.1 shows an overview of the different contributions and how they relate to each other. Fig. 8.2 illustrates the decision tree of relationship between research hypotheses and corresponding contributions through doughnut pie charts. The relationship of parts to the whole in each chart defines the time-driven impact and influence of the contribution on the relevant research sub-hypothesis. The 10 contributions of this PhD dissertation have addressed different aspects of the research hypotheses. The research carried out has been evaluated in Chapter 6 following different case studies and assessment criterion. For all of these reasons, the sole author of this document believes that the hypotheses have been validated. The following parts briefly overview the contributions.

Table 8.1: Traceability between research contributions and publications. Each publication can support a contribution at three different levels: High (H), Medium (M) and Low (L).

Res. Cont.	[1]	[2]	[3]	[5]	[6]	[7]	[8]	[9]	PhD Diss.
SLC-1	–	–	–	L	L	H	M	M	H
SLC-2	–	–	–	–	–	H	L	L	H
SLC-3	–	–	–	–	–	H	L	L	H
ALC-1	M	L	L	H	M	M	H	H	H
ALC-2	L	L	–	M	L	L	M	H	H
ALC-3	L	L	L	M	M	L	L	H	H
ALC-4	H	H	L	H	L	M	H	H	H
ALC-5	–	–	–	–	–	–	–	H	H
ALC-6	–	–	–	–	–	–	–	H	H
ALC-7	–	–	–	–	–	H	–	–	H

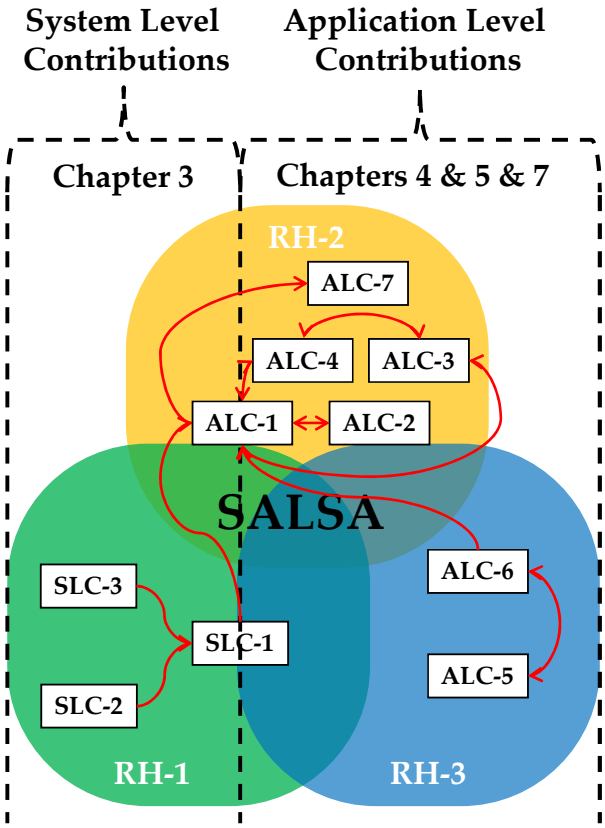


Fig. 8.1: Overview of the contributions of this PhD dissertation and their linking.

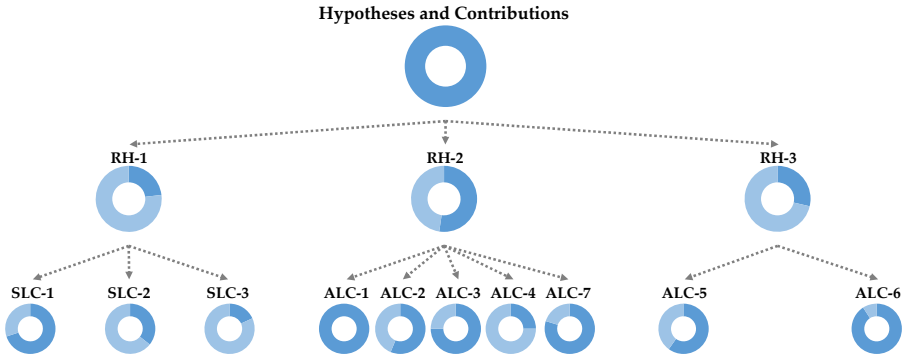


Fig. 8.2: Decision tree of research hypotheses and contributions.

8.1.1 Formal Framework for Smart Grid

This category of research hypothesis made the following contributions:

SLC-1: Propose a generic formal framework by providing a smooth way to describe smart grid elements, domains, and their interactions;

SLC-2: Develop and formulate three essential smart grid aspects, i.e., hardware, software, and network, to demonstrate the formal framework's scalability, reusability, interoperability, and updatability;

SLC-3: Model, design, and develop two novel UML smart grid profiles to map the formal framework into various smart grid applications.

A formal framework for modeling the main semantics of smart grid systems, with an emphasis on the customers and operations domains, have been defined and mathematically formalized (see Section 3.2 for more information). Two novel UML profiles, shown in Figs. 3.3 and 3.4, were developed to integrate smart grid aspects into the generic formal framework profile and to support the implementation of the framework, reflect the mathematical formulas, and create formal grid models. To prove the validity of the formal framework in building an efficient simulator for smart grid applications, a case study was proposed in Chapter 3 demonstrating how to synthesize the formal framework into an executable code.

8.1.2 The SALSA System

This category of research hypothesis made the following contributions:

ALC-1: Design, implementation, and evaluation of the SALSA system consisting of the following high-level features: i) conforming to the hierarchical grid infrastructure; ii) operating using no forecasting services and historical data; iii) performing in real-time (one minute to one hour); and iv) following an easily expandable agent-based modeling design;

ALC-2: Design and support different LDS mechanisms to enable the load shifting from peak to off-peak periods and develop a novel load demand buffering sub-system to serve hundreds of thousands of prosumers (including diverse appliance types);

ALC-3: Model, develop, and implement a novel concept of flexibility zone for synthesizing diverse flexibility characteristics by the integration of PVs with BESSs in two MO-MINLP models for prosumers to schedule their appliances and share their surplus energy with the grid and for the aggregator to efficiently match prosumers' demands with surpluses;

ALC-4: Design and implement dynamic ECT policies for prosumers and the aggregator (complied with the smart grid operating regimes defined by the USEF) following day-ahead peak consumption, real-time aggregated load demands, and day-ahead electricity prices,

ALC-7: Develop and implement two interactive web services for: i) the SALSA system, which is accessible at salsa.semiah.eu, and ii) a microgrid online training center hosted at UCLA SMERC.

The SALSA system has been derived from the formal framework proposed in Chapter 3. This system conforms to the hierarchical grid infrastructure and enables an easy expansion approach due to its agent-based modeling design. It has three main features: i) it operates without using any forecasting services, ii) it is independent of any historical data (in both prosumers and aggregators layers), and iii) it is capable of performing in real-time. SALSA is able to manage multi-class smart appliances. The shiftability feature of appliances helps it incentivize prosumers to modify their load consumption scenarios over time. This modification provides the grid with an opportunity to flatten the grid demand by shifting loads from peak to off-peak periods. This is done using a negotiation between prosumers and an aggregator. Beside proposing consumption-based flexibility types (shiftability), integrating PVs and BESSs with the SALSA system provides prosumers with more flexibility types, i.e., generation and storing. This integration has resulted in a new concept for prosumers named flexibility zone. This zone mathematically formulates the feasible integrated region of flexibilities over time. Prosumers, by modifying their load consumption scenarios, aim at reducing their electricity cost. However, some appliances are non-shiftable and some may have limited amount of flexibility. Therefore, the SALSA system follows a bound-

ary to keep the aggregated load demands below dynamic ECTs over time. This concept supports various policies, such as coupling with electricity pricing strategies or PV production profile.

To enable the SALSA to serve a scalable number of appliances in the grid, this dissertations developed a load demand buffering sub-system. It classifies load demands according to their shiftability feature and remaining flexibility. Non-shiftable appliances and those, which have insufficient amount of flexibility remained, are immediately allowed to operate. Others wait until EMSs makes decisions according to the grid situation at that time interval.

Incorporating all contributions made so far, the SALSA system developed a decentralized Multi-Objective Mixed Integer Nonlinear Programming optimization model for prosumers. This model is responsible for providing prosumers with a set of feasible solutions. Each solution defines actions to be taken in response to load demands, PV generation, and BESS status at each time interval. This optimization model is also integrated with diverse LDS mechanism, e.g., 0-1 Knapsack, EDF, etc. Each mechanism has specific features for scheduling load demands. For instance, the 0-1 Knapsack approach tries to operate as many load demands as possible subject to 1its capacity (equivalent to the ECT). On the other side, the EDF mechanism sorts load demands according to their remaining flexibility from the current time interval. For more information regarding other mechanism, the reader is referred to Section 4.1.4.3. Achievements provided in Section 4.2.1.1 have proposed a similar MO-MINLP optimization model for the aggregator. This model, by aiming at maximizing the profit and minimizing the grid purchase simultaneously, is responsible for matching prosumers' demand with surplus subject to ECTs over time.

Finally, the SALSA provides two web services, where a combination of JavaScript, PHP, and Matlab code orchestrate various contributions in an interactive way.

8.1.3 The Bilateral Multi-Issue Negotiation Approach

This category of research hypothesis made the following contributions:

ALC-5: Materialize the trading of prosumers' heterogeneous flexibilities through the negotiation approach.

ALC-6: Model and implement a bilateral multi-issue negotiation approach to enable the negotiation between a VPP (on behalf of prosumers) and the aggregator.

This category proposed a bilateral multi-issue negotiation approach in a non-cooperative environment. This approach follows the two MO-MINLP optimization models studied in the previous section. To relieve the burden of parallel bilateral communications between prosumers and the aggregator,

this approach employs an efficient negotiation approach, in which a Virtual Power Plant, on behalf of prosumers, negotiate on packaged power and price offers with the aggregator subject to having no knowledge about each other's preferences and utility functions. This approach utilizes an alternating offer package production protocol and a reactive utility value concession strategy, where none of the negotiators has incentive to deliberately stop conceding while the zone of agreement must remain nonempty.

8.2 Future Work

There are different directions of future work that either build on top of the developed research or explore new challenges brought to light during this PhD studies.

As possible future work for the first research hypothesis, to confirm the generality and re-usability of the framework in different directions, more systems, similar to the SALSA, should be investigated. Furthermore, to evaluate how efficient the framework behaves in a simulated ICT-based power grid environment, smart grid communication protocols will be necessary to integrate with the framework (updating the network aspect). At the moment, the code is not automatically generated by the framework, but manually mapped from the UML model to Matlab code. This method can be promoted by an automatic code generation tool, e.g., Acceleo, which is an open-source code generator allowing researchers to use a model-driven approach to build applications.

For the SALSA system, it is left for future work to: i) apply learning and forecasting approaches to enhance the load demand scheduling efficiency, ii) investigate the correlation between the flexibility zone, PV capacity and BESS sizing, and iii) analyze the sensitivity of rapid change in prosumers' flexibilities on their comfort level and profit as well as on the grid status.

Finally, for the third research hypothesis, future work will focus on adding a negotiation level between aggregators, integrating industrial and commercial prosumers, where their intermittent load consumptions and power generations can lead to a difficulty in balancing supply and demand, and investigating the network performance and communication delays between the hierarchical agents in the power grid.

References

[1] A. G. Azar, R. H. Jacobsen, and Q. Zhang, "Aggregated Load Scheduling for Residential Multi-Class Appliances: Peak Demand Reduction," in *IEEE International Conference on the European Energy Market (EEM)*, 2015, pp. 1–6.

[2] R. H. Jacobsen, A. G. Azar, Q. Zhang, and E. S. M. Ebeid, "Home Appliance Load Scheduling With SEMIAH," in *Fourth International Conference on Smart Systems, Devices, and Technologies (SMART)*, 2015, pp. 1–2.

[3] 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.

[4] A. G. Azar, "Demand Response Driven Load Scheduling in Formal Smart Grid Framework," *Technical Report Electronics and Computer Engineering, Aarhus University*, vol. 4, no. 24, 2016.

[5] A. G. Azar and R. H. Jacobsen, "Appliance Scheduling Optimization for Demand Response," *International Journal on Advances in Intelligent Systems*, vol. 9, no. 1 & 2, pp. 50–64, 2016.

[6] R. H. Jacobsen, A. G. Azar, and E. S. M. Ebeid, "Design of an Event-Driven Residential Demand Response Infrastructure," in *IEEE Euromicro Conference on Digital System Design (DSD)*, 2016, pp. 38–45.

[7] A. G. Azar, E. Ebeid, and R. H. Jacobsen, "A Formal Framework for Modeling Smart Grid Applications: Demand Response Case Study," in *IEEE Euromicro Conference on Digital System Design (DSD)*, 2016, pp. 46–54.

[8] A. G. Azar and R. H. Jacobsen, "Agent-Based Charging Scheduling of Electric Vehicles," in *IEEE Online Conference on Green Communications (OnlineGreenComm)*, 2016, pp. 64–69.

[9] A. G. Azar, H. Nazaripouya, B. Khaki, C.-C. Chu, R. Gadh, and R. H. Jacobsen, "A Non-Cooperative Framework for Coordinating a Neighborhood of Distributed Prosumers," *Submitted to IEEE Transactions on Smart Grid*, 2017.

[10] H. Nazaripouya, B. Wang, Y. Wang, P. Chu, H. Pota, and R. Gadh, "Univariate Time Series Prediction of Solar Power Using a Hybrid Wavelet-ARMA-NARX Prediction Method," in *IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*, 2016, pp. 1–5.

[11] A. Backers, F. Bliek, M. Broekmans, C. Groosman, H. De Heer, M. van der Laan, M. de Koning, J. Nijtmans, P. Nuygen, T. Snberg *et al.*, "An Introduction to the Universal Smart Energy Framework (USEF)," *USEF Foundation*, 2014.

[12] P. Capros, A. De Vita, N. Tasios, D. Papadopoulos, P. Siskos, E. Apostolaki, M. Zampara, L. Paroussos, K. Fragiadakis, N. Kouvaritakis *et al.*, *EU Energy, Transport and GHG Emissions: Trends to 2050, Reference Scenario 2013*, 2013.

[13] F. Li, W. Qiao, H. Sun, H. Wan, J. Wang, Y. Xia, Z. Xu, and P. Zhang, "Smart Transmission Grid: Vision and Framework," *IEEE Transactions on Smart Grid*, vol. 1, no. 2, pp. 168–177, 2010.

[14] P. Palensky and D. Dietrich, "Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 381–388, 2011.

[15] N. Framework, "Roadmap for Smart Grid Interoperability Standards," *NIST special publication*, vol. 1108, 2010.

[16] CEN-CENELEC-ETSI, "Smart Grid Coordination Group: Sustainable Processes," 2012, Tech. Rep. November.

[17] H. Farhangi, "The Path of the Smart Grid," *IEEE Power and Energy Magazine*, vol. 8, no. 1, 2010.

[18] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid-The New and Improved Power Grid: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944–980, 2012.

[19] T. Godfrey, S. Mullen, R. C. Dugan, C. Rodine, D. W. Griffith, and N. Golmie, "Modeling Smart Grid Applications With Co-Simulation," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2010, pp. 291–296.

[20] S. Schutte, S. Scherfke, and M. Troschel, "Mosaik: A Framework for Modular Simulation of Active Components in Smart Grids," in *IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS)*, Oct. 2011, pp. 50–60.

[21] D. Montenegro, M. Hernandez, and G. Ramos, "Real Time OpenDSS Framework for Distribution Systems Simulation and Analysis," in *Sixth IEEE/PES Transmission and Distribution: Latin America Conference and Exposition (T D-LA)*, Sept 2012, pp. 1–5.

[22] F. Andrén, M. Stifter, and T. Strasser, "Towards a Semantic Driven Framework for Smart Grid Applications: Model-Driven Development Using CIM, IEC 61850 and IEC 61499," *Informatik-Spektrum*, vol. 36, no. 1, pp. 58–68, 2013.

[23] Smart Grid Coordination Group, "Reference Architecture for the Smart Grid," Technical Report, 2012.

[24] M. de Miguel, T. Lambolais, M. Hannouz, S. Betgé-Brezetz, and S. Piekarec, "UML Extensions for the Specification and Evaluation of Latency Constraints in Architectural Models," in *2nd international workshop on Software and performance*. New York, NY, USA: ACM, 2000, pp. 83–88.

[25] A. Hennig, D. Revill, and M. Ponitsch, "From UML to Performance Measures–Simulative Performance Predictions of IT-Systems Using the Jboss Application Server With OMNET++," *AI-Dabass*, vol. 1, no. 1.6, pp. 1–11, 2003.

[26] E. Ebeid, F. Fummi, and D. Quaglia, "Model-Driven Design of Network Aspects of Distributed Embedded Systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 4, pp. 603–614, 2015.

[27] K. C. Sou, J. Weimer, H. Sandberg, and K. H. Johansson, "Scheduling Smart Home Appliances Using Mixed Integer Linear Programming," in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011, pp. 5144–5149.

[28] J.-W. Lee and D.-H. Lee, "Residential Electricity Load Scheduling for Multi-Class Appliances With Time-of-Use Pricing," in *IEEE GLOBECOM Workshops (GC Wkshps)*, 2011, pp. 1194–1198.

[29] N. Kumaraguruparan, H. Sivaramakrishnan, and S. S. Sapatnekar, "Residential Task Scheduling Under Dynamic Pricing Using the Multiple Knapsack Method," in *IEEE PES Innovative Smart Grid Technologies (ISGT)*, 2012, pp. 1–6.

[30] O. A. Sianaki, O. Hussain, and A. R. Tabesh, "A Knapsack Problem Approach for Achieving Efficient Energy Consumption in Smart Grid for Endusers' Life Style," in *IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES)*, 2010, pp. 159–164.

[31] A. Mohd, E. Ortjohann, A. Schmelter, N. Hamsic, and D. Morton, "Challenges in Integrating Distributed Energy Storage Systems Into Future Smart Grid," in *IEEE International Symposium on Industrial Electronics (ISIE)*, 2008, pp. 1627–1632.

[32] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, "A Survey on Demand Response Programs in Smart Grids: Pricing Methods and Optimization Algorithms," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 152–178, 2015.

[33] S. Rahnama, J. D. Bendtsen, J. Stoustrup, and H. Rasmussen, "Robust Aggregator Design for Industrial Thermal Energy Storages in Smart Grid," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 902–916, 2017.

[34] M. Boehm, L. Dannecker, A. Doms, E. Dovgan, B. Filipič, U. Fischer, W. Lehner, T. B. Pedersen, Y. Pitarch, L. Šikšnys *et al.*, "Data Management in the Mirabel Smart Grid System," in *Joint EDBT/ICDT Workshops*. ACM, 2012, pp. 95–102.

[35] J. Hu, S. You, M. Lind, and J. Ostergaard, "Coordinated Charging of Electric Vehicles for Congestion Prevention in the Distribution Grid," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 703–711, 2014.

[36] S. Rahnama, S. E. Shafiei, J. Stoustrup, H. Rasmussen, and J. Bendtsen, "Evaluation of Aggregators for Integration of Large-Scale Consumers in Smart Grid," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 1879–1885, 2014.

[37] P. Siano, "Demand Response and Smart grids—A Survey," *Renewable and Sustainable Energy Reviews*, vol. 30, pp. 461–478, 2014.

[38] A. Di Giorgio and L. Pimpinella, "An Event Driven Smart Home Controller Enabling Consumer Economic Saving and Automated Demand Side Management," *Applied Energy*, vol. 96, pp. 92–103, 2012.

[39] Y. Wang, I. R. Pordanjani, and W. Xu, "An Event-Driven Demand Response Scheme for Power System Security Enhancement," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 23–29, 2011.

[40] T. AlSkaif, A. C. Luna, M. G. Zapata, J. M. Guerrero, and B. Bellalta, "Reputation-Based Joint Scheduling of Households Appliances and Storage in a Microgrid With a Shared Battery," *Energy and Buildings*, vol. 138, pp. 228–239, 2017.

[41] T. AlSkaif, M. G. Zapata, and B. Bellalta, "A Reputation-Based Centralized Energy Allocation Mechanism for Microgrids," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2015, pp. 416–421.

[42] N. G. Paterakis, O. Erdinç, I. N. Pappi, A. G. Bakirtzis, and J. P. Catalão, "Coordinated Operation of a Neighborhood of Smart Households Comprising Electric Vehicles, Energy Storage and Distributed Generation," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2736–2747, 2016.

[43] J. C. Mukherjee and A. Gupta, "A Review of Charge Scheduling of Electric Vehicles in Smart Grid," *IEEE Systems Journal*, vol. 9, no. 4, pp. 1541–1553, 2015.

[44] F. Rassaei, W.-S. Soh, and K.-C. Chua, "Demand Response for Residential Electric Vehicles With Random Usage Patterns in Smart Grids," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1367–1376, 2015.

[45] H. Mohsenian-Rad and M. Ghamkhari, "Optimal Charging of Electric Vehicles With Uncertain Departure Times: A Closed-Form Solution," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 940–942, 2015.

[46] Z. Xu, W. Su, Z. Hu, Y. Song, and H. Zhang, "A Hierarchical Framework for Coordinated Charging of Plug-in Electric Vehicles in China," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 428–438, 2016.

[47] Y. He, B. Venkatesh, and L. Guan, "Optimal Scheduling for Charging and Discharging of Electric Vehicles," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1095–1105, 2012.

[48] S. Deilami, A. S. Masoum, P. S. Moses, and M. A. Masoum, "Real-Time Coordination of Plug-in Electric Vehicle Charging in Smart Grids to Minimize Power Losses and Improve Voltage Profile," *IEEE Transactions on Smart Grid*, vol. 2, no. 3, pp. 456–467, 2011.

[49] A. Veit and H.-A. Jacobsen, "Multi-Agent Device-Level Modeling Framework for Demand Scheduling," in *IEEE International Conference on Smart Grid Communications*, 2015, pp. 169–174.

[50] J. Hu, H. Morais, M. Lind, and H. W. Bindner, "Multi-Agent Based Modeling for Electric Vehicle Integration in a Distribution Network Operation," *Electric Power Systems Research*, vol. 136, pp. 341–351, 2016.

[51] I. G. Unda, P. Papadopoulos, S. Skarvelis-Kazakos, L. M. Cipcigan, N. Jenkins, and E. Zabala, "Management of Electric Vehicle Battery Charging in Distribution Networks With Multi-Agent Systems," *Electric Power Systems Research*, vol. 110, pp. 172–179, 2014.

[52] S. R. Griful, "System Design and Evaluation for Residential Demand Response," *PhD Dissertation, Aarhus University*, 2016.

[53] S. M. Nosratabadi, R.-A. Hooshmand, and E. Gholipour, "A Comprehensive Review on Microgrid and Virtual Power Plant Concepts Employed for Distributed Energy Resources Scheduling in Power Systems," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 341–363, 2017.

[54] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa, "Noncooperative and Cooperative Optimization of Distributed Energy Generation and Storage in the Demand-Side of the Smart Grid," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2454–2472, 2013.

[55] G. d. O. e Silva and P. Hendrick, "Lead–Acid Batteries Coupled With Photovoltaics for Increased Electricity Self-Sufficiency in Households," *Applied Energy*, vol. 178, pp. 856–867, 2016.

[56] E. Nyholm, J. Goop, M. Odenberger, and F. Johnsson, "Solar Photovoltaic-Battery Systems in Swedish Households–Self-Consumption and Self-Sufficiency," *Applied Energy*, vol. 183, pp. 148–159, 2016.

[57] K. Worthmann, C. M. Kellett, P. Braun, L. Grüne, and S. R. Weller, "Distributed and Decentralized Control of Residential Energy Systems Incorporating Battery Storage," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1914–1923, 2015.

[58] Q. D. La, Y. W. E. Chan, and B.-H. Soong, "Power Management of Intelligent Buildings Facilitated by Smart Grid: A Market Approach," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1389–1400, 2016.

[59] T. AlSkaif, M. G. Zapata, B. Bellalta, and A. Nilsson, "A Distributed Power Sharing Framework Among Households in Microgrids: A Repeated Game Approach," *Computing*, vol. 99, no. 1, pp. 23–37, 2017.

[60] T. Taniguchi, K. Kawasaki, Y. Fukui, T. Takata, and S. Yano, "Automated Linear Function Submission-Based Double Auction as Bottom-Up Real-Time Pricing in a Regional Prosumers' Electricity Network," *Energies*, vol. 8, no. 7, pp. 7381–7406, 2015.

[61] K. Rahbar, C. C. Chai, and R. Zhang, "Energy Cooperation Optimization in Microgrids with Renewable Energy Integration," *IEEE Transactions on Smart Grid*, 2016.

[62] B. Gao, X. Liu, W. Zhang, and Y. Tang, "Autonomous Household Energy Management Based on a Double Cooperative Game Approach in the Smart Grid," *Energies*, vol. 8, no. 7, pp. 7326–7343, 2015.

[63] A. Sha and M. Aiello, "A Novel Strategy for Optimising Decentralised Energy Exchange for Prosumers," *Energies*, vol. 9, no. 7, p. 554, 2016.

[64] N. Liu, X. Yu, C. Wang, C. Li, L. Ma, and J. Lei, "Energy Sharing Model With Price-Based Demand Response for Microgrids of Peer-to-Peer Prosumers," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3569–3583, 2017.

[65] Y. Zhou, S. Ci, H. Li, and Y. Yang, "A New Framework for Peer-to-Peer Energy Sharing and Coordination in the Energy Internet," in *IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[66] M. Vinyals, M. Velay, and M. Sisinni, "A Multi-Agent System for Energy Trading Between Prosumers," in *14th International Conference on Distributed Computing and Artificial Intelligence*, vol. 620. Springer, 2018, p. 79.

[67] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[68] R. Zheng, T. Dai, K. Sycara, and N. Chakraborty, "Automated Multilateral Negotiation on Multiple Issues with Private Information," *INFORMS Journal on Computing*, vol. 28, no. 4, pp. 612–628, 2016.

[69] E. Ernst, "Separation of Concerns," in *AOSD Workshop on Software-Engineering Properties of Languages for Aspect Technologies (SPLAT)*, 2003.

[70] International Electrotechnical Commission (IEC), "IEC 61970: Energy Management System Application Program Interface (EMS-API) - Part 301: Common Information Model (CIM) Base," Technical Report, 2009.

[71] E. Ebeid, S. Rotger-Griful, S. A. Mikkelsen, and R. H. Jacobsen, "A Methodology to Evaluate Demand Response Communication Protocols for the Smart Grid," in *IEEE International Conference on Communication Workshop (ICCW)*, 2015, pp. 2012–2017.

[72] 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.

[73] 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.

[74] S. Bera, S. Misra, and D. Chatterjee, "C2C: Community-Based Cooperative Energy Consumption in Smart Grid," *IEEE Transactions on Smart Grid*, 2017.

[75] M. Pipattanasomporn, M. Kuzlu, S. Rahman, and Y. Teklu, "Load Profiles of Selected Major Household Appliances and Their Demand Response Opportunities," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 742–750, 2014.

[76] M. Pipattanasomporn, M. Kuzlu, and S. Rahman, "An Algorithm for Intelligent Home Energy Management and Demand Response Analysis," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 2166–2173, 2012.

[77] 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.

[78] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa, "Demand-Side Management via Distributed Energy Generation and Storage Optimization," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 866–876, 2013.

[79] K. Deb, "Multi-Objective Optimization," in *Search Methodologies*. Springer, 2014, pp. 403–449.

[80] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack problems*. Springer Science & Business Media, 2013.

[81] G. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Springer Science & Business Media, 2011, vol. 24.

[82] G. T. Costanzo, G. Zhu, M. F. Anjos, and G. Savard, "A System Architecture for Autonomous Demand Side Load Management in Smart Buildings," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 2157–2165, 2012.

[83] J. Lee, "Time-Reversibility for Real-Time Scheduling on Multiprocessor Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 1, pp. 230–243, 2017.

[84] T. A. AlEnawy and H. Aydin, "Energy-Aware Task Allocation for Rate Monotonic Scheduling," in *11th IEEE Real Time and Embedded Technology and Applications Symposium (RTAS)*, 2005, pp. 213–223.

[85] S. Salinas, M. Li, P. Li, and Y. Fu, "Dynamic Energy Management for the Smart Grid With Distributed Energy Resources," *IEEE Transactions on Smart Grid*, vol. 4, no. 4, pp. 2139–2151, 2013.

[86] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001, vol. 16.

[87] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and Exploitation in Evolutionary Algorithms: A Survey," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, p. 35, 2013.

[88] K. Deb, "Multi-objective optimisation using evolutionary algorithms: an introduction," in *Multi-objective evolutionary optimisation for product design and manufacturing*. Springer, 2011, pp. 3–34.

[89] H. Jain and K. Deb, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach," *IEEE Transactions Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[90] A. J. Hanson, "Hyperquadrics: Smoothly Deformable Shapes With Convex Polyhedral Bounds," *Computer Vision, Graphics, and Image Processing*, vol. 44, no. 2, pp. 191–210, 1988.

[91] A. Rubinstein, "Perfect Equilibrium in a Bargaining Model," *Econometrica: Journal of the Econometric Society*, pp. 97–109, 1982.

[92] G. Lai and K. Sycara, "A Generic Framework for Automated Multi-Attribute Negotiation," *Springer Group Decision and Negotiation*, vol. 18, no. 2, pp. 169–187, 2009.

[93] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

[94] H. A. Simon, "Rational Choice and the Structure of the Environment," *Psychological Review*, vol. 63, no. 2, p. 129, 1956.

[95] J. F. Nash Jr, "The Bargaining Problem," *Econometrica: Journal of the Econometric Society*, pp. 155–162, 1950.

[96] Nord Pool Spot, "https://www.nordpoolspot.com," accessed: May 22, 2017.

[97] K. Qian, C. Zhou, M. Allan, and Y. Yuan, "Load Model for Prediction of Electric Vehicle Charging Demand," in *IEEE International Conference on Power System Technology*, 2010, pp. 1–6.

[98] B. Wang, Y. Wang, H. Nazaripouya, C. Qiu, C.-C. Chu, and R. Gadh, "Predictive Scheduling Framework for Electric Vehicles With Uncertainties of User Behaviors," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 52–63, 2017.

Part II

Parpers

Paper A

Aggregated Load Scheduling for Residential Multi-Class Appliances: Peak Demand Reduction

The paper presented in this chapter has been accepted as a conference publication [1].

[1] Armin Ghasem Azar, Rune Hylsberg Jacobsen, and Qi Zhang, "Aggregated Load Scheduling for Residential Multi-Class Appliances: Peak Demand Reduction," In *IEEE International Conference on the European Energy Market (EEM)*, 2015, pages 1-6, doi: 10.1109/EEM.2015.7216702

Authors contribution: Armin Ghasem Azar conceived the initial design of the demand response system; Armin Ghasem Azar carried out the literature survey and formulated the proposed scheduling system and algorithm; Rune Hylsberg Jacobsen and Qi Zhang helped in maturing the proposed system; Armin Ghasem Azar was the main editor of the manuscript; Rune Hylsberg Jacobsen and Qi Zhang contributed with text in selected areas and to the reviewing process.

Aggregated Load Scheduling for Residential Multi-Class Appliances: Peak Demand Reduction

Armin Ghasem Azar, Rune Hylsberg Jacobsen, Qi Zhang
Department of Engineering
Aarhus University
Aarhus, Denmark
{aga, rhj, qz}@eng.au.dk

***Abstract*—The Smart Grid represents an unprecedented opportunity to move the energy industry into a new era. In this context, Demand Response programs provide mechanisms to regulate the power demand through load control according to conditions of the supply side, where consumers can efficiently schedule the operation of their appliances. This paper has proposed an efficient local load scheduling optimization strategy for residential multi-class multi-constraint appliances by shifting and interrupting load requests to flatten the aggregated consumption. One scenario for each smart house, including the desirable usage schedule of its appliances in a 24-hour period, is considered. The proposed strategy has supposed a time-independent constant electricity consumption threshold, imposed by the grid stability management, in each time interval. The demand response system attempts to optimally schedule the received load requests over time, aiming at flattening the aggregated consumption meanwhile, maximizing satisfaction of consumers. The results have indicated that decreasing the electricity consumption threshold up to 60% of the maximum peak demand results in significant aggregated consumption flattening and also admissible delay in appliance reception.**

***Index Terms*—Smart Grid, Supply and Demand, Load Management, Scheduling.**

I. Introduction

The Smart Grid defines an electricity network that can intelligently integrate the behaviors and actions of generators and consumers, in order to efficiently deliver sustainable, economic, and secure electricity supply [1]. Demand Response (DR) has been thought as a key component in the smart grid which allows electricity consumers to adapt their electricity consumption according to fluctuations in the electricity generation over time. DR technologies facilitate the communications between energy utilities and smart appliances at the customer premises, which are indispensable for consumers' ability to reduce or to shift their power consumption during peak demand periods [2], [3].

Consumers become secondary actors in the electricity wholesale market dynamics through DR programs [4]. Load control actions are leveraged by market actors such as aggregators that offer specific load reductions in the market. Participants of DR programs have the opportunity to help those reduce the risk of power grid outages thus provide a value to the Distribution System Operator (DSO).

This paper critically discusses the *Peak Demand Reduction* problem based on the *Appliance Reception Minimization* method. First, a DR model for smart houses, where the DR System (DRS) receives and schedules a large number of consumers' partial load requests, is proposed. The advantage of this approach 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. Furthermore, an efficient local load scheduling optimization strategy has been proposed for smart houses to shift or to interrupt demands to flatten the aggregated consumption, where each smart house has a desired usage scenario of its appliances. When consumers provide their appliances to the DRS in the "DR Ready" mode, they give permission to the DRS to schedule their multi-class multi-constraint appliances in a 24-hour period restricting to a specific deadline flexibility for completion of each appliance in a worst case.

So far, however, there has been little discussion about coupling *Electricity Consumption Threshold (ECT)*, provided by the DSO grid stability management, with DR programs preventing appliances of smart houses to start at their desired time in order to flatten the aggregated electricity consumption. This paper follows a case-study design that appropriately utilizes the time-independent *ECT* constraint to schedule the received load requests of appliances of smart houses in each time interval. Although consumers provide flexibility to the DRS, they are not interested in waiting too long to receive their appliances in the completed status. Hence, the proposed aggregated local load scheduling technique for residential appliances aims at *maximizing satisfaction of consumers* while considering *ECT*. Here, *local* means receiving load requests in specific time intervals and scheduling them in these intervals.

The remainder of the paper is organized as follows: Section II reviews related work. Section III presents descriptions of the system model. Section IV clarifies the proposed local load scheduling algorithm with its relevant sub-procedures. Section V demonstrates the experimental setup and obtained results with their evaluations. Finally, conclusion and future work are provided in Section VI.

1
978-1-4673-6692-2/15/$31.00 ©2015 IEEE

II. Related Work

In recent years, there has been an increasing amount of literature on incentivizing consumers to shift their electricity consumption by varying the electricity prices [5], [6]. Sou *et al.* [5] investigated the minimization of electricity bills combined with enforcing uninterruptible and sequential operation model constraints. Nonetheless, the utilized mixed integer linear programming approach to solve the scheduling problem is not scalable and the appliance classification is limited to the interruptibility feature. In addition, [6] has proposed an electricity load scheduling algorithm that controls the operation time and energy consumption of appliances based on adapting time-of-use pricing to minimize the total electricity bill. A serious weakness with this argument, however, is that the authors have used solely one smart house as the test-bed, excluding any aggregated consumption threshold.

On the other hand, a solution to the problem of optimally scheduling a set of residential appliances under the day-ahead variable peak pricing scheme has been studied in [7]. Here, the objectives are minimizing the electricity bills and spreading the electricity usage out in each time interval simultaneously. On the contrary, they have considered a limited number of appliances. Finally, the focus in [8] is on applying the priority-based appliance methodology to quantify preferences of consumers for using appliances during peak times based on the Knapsack problem approach. Nonetheless, in the proposed mechanism, there is no consumption threshold constraint to prevent consumers from exceeding it.

III. System Model

Smart houses play a critical role in DR programs [9]. When the consumer of a smart house operates his appliances in the "DR ready" mode, he offers a flexibility to the grid and permits the DRS to take the control of his appliances. In this paper, it is assumed that there are $N \in \mathbb{N}$ smart houses, where each smart house $H_i, i \in (1,2,\dots,N)$, has $A \in \mathbb{N}$ appliances. Furthermore, $x_{i,j}^t \in \{0,1\}$ denotes the decision variable of the DRS which allows j^{th} appliance of i^{th} smart house to start or to continue its work in time interval t or not. In the following, the objective function and relevant constraints of the proposed system model will be clarified.

A. Objective function

Consumers may give priorities to their appliances based on their preferences [8]. More accurately, a time-independent constant pairwise priority exists between two distinct types of appliance based on some criteria, e.g., emergent usage, welfare, or electricity cost. Consumers provide their normalized priority vector to the DRS by using their own pairwise comparisons. The priority vector of i^{th} smart house, PV_i, includes A elements, in which its each element $0 < p_{i,j} \leq 1$ refers to the priority of j^{th} appliance of that smart house. Since PV_i is normalized, the sum of its elements should be equal to one. The DRS employs the provided priority vectors to qualify for permitting corresponding appliances of received load requests to start or to continue in each time interval. In this paper, the comparison criterion is the emergent usage. Even though consumers permit the DRS to schedule their appliances, however, they hope to receive their appliances in the completed status at when they desire. Equation (1) formulates this objective:

$$f(t) = max \sum_{i=1}^{N} \sum_{j=1}^{A} \left(x_{i,j}^t \times p_{i,j}\right). \quad (1)$$

B. Constraints

Appliance full operation: Appliances are drivers of electricity demands in each smart house. The electricity consumption of j^{th} appliance of i^{th} smart house in time interval t is $EC_{i,j}^t \in \mathbb{R}_0^+$ (watt). It should be noted that $TEC_{i,j} \in \mathbb{R}^+$ is its total electricity consumption in a 24-hour period. To guarantee the full operation of appliances, the DRS checks whether each appliance has completed its duty during the day defined by last time interval of the day, i.e. T. Hence, (2) is imposed to satisfy this hard constraint:

$$\sum_{t=1}^{T} \left(EC_{i,j}^t \times x_{i,j}^t\right) = TEC_{i,j}. \quad (2)$$

Smart features: Appliances are divisible based on some smart features. Fig. 1 pictures classification of appliances coupling with correspondent examples. Firstly, appliances have been classified according to the *shiftability* feature [10]. Shiftability is to give permission to the DRS to shift load requests of *shiftable appliances* to another time interval. On the other hand, load requests of some appliances, such as the refrigerator, cannot be shifted. Thus, those appliances become members of *non-shiftable appliances*. Secondly, the shiftable appliances can be divided into two groups based on the *interruptibility* feature. The electric vehicle is an example of this feature, where the DRS can both shift and interrupt the duty cycle of charging the electric vehicle. Nevertheless, those appliances which are shiftable but uninterruptible are called *uninterruptible appliances* (e.g., the dish washer). Consequently, the DRS, while receiving an uninterruptible load request in t^{th} time interval, should check whether the relevant appliance has been allowed to start or to continue its work in the $(t-1)^{th}$ time interval. If so, the DRS cannot interrupt and shift it to another time interval. Equation (3) is a hard constraint and belongs to only uninterruptible appliances:

$$\begin{cases} x_{i,j}^t = 1, & x_{i,j}^{t-1} = 1, \\ x_{i,j}^t \in \{0,1\}, & otherwise. \end{cases} \quad (3)$$

Appliance dependency: In practice, there are some *dependency* relationships between consumption activities of some appliances of each smart house [2]. These relationships impose a hard constraint on the DRS that must be satisfied entirely. Dependency is denoted by the relationship between two different appliances. For instance, it is infeasible to put clothes into the laundry dryer before washing them. More accurately, these dependencies can be divided into two independent groups named *consecutive* or *concurrent* dependencies. The former group relates to those appliances which cannot be utilized at the same time, as exemplified before. Alternatively, concurrent dependency refers to operations which should be performed simultaneously (e.g.,

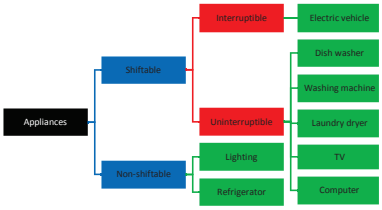


Figure 1. Classification of appliances

lighting and watching TV). $D_{i,j,k} \in \{0,1,2\}$, in which $\{j,k\} \in (1,2,\dots,A)$, corresponds to the dependency value between appliances j and k of smart house i. If it is zero, there is no dependency between the relevant appliances. If $D_{i,j,k}$ is equivalent to one, there is a consecutive dependency relationship. Here, if appliance k finishes its duty until time interval $t-1$, then, appliance j can be allowed to start from time interval t. Finally, if the dependency value is equal to two, there is a concurrent dependency. In this situation, appliance j can be allowed to start or to continue if appliance k is allowed to start or to continue at the same time interval. Equation (4) demonstrates the dependency constraint:

$$\begin{cases} x_{i,j}^t \in \{0,1\}, & \exists k \rightarrow \left(\left(D_{i,j,k}=1\right) \wedge \left(\sum_{h=1}^{t-1} \left(EC_{i,k}^h \times x_{i,k}^h\right) = TEC_{i,k} \right) \right), \\ x_{i,j}^t \in \{0,1\}, & \exists k \rightarrow \left(\left(D_{i,j,k}=2\right) \wedge \left(x_{i,k}^t = 1\right) \right), \\ x_{i,j}^t = 0, & otherwise. \end{cases} \quad (4)$$

Deadline flexibility: As mentioned previously, consumers put their appliances in the "DR Ready" mode to be scheduled by the DRS. In addition, they provide a particular time-oriented hard *deadline flexibility* constraint of each appliance to the DRS. With respect to the load profile of each appliance, the required completion time period of each appliance is known to consumers. For instance, one consumer desires to charge the electric vehicle from 18:00 to 20:00. Nevertheless, one may provide two hours flexibility of the electric vehicle to the DRS. The provided deadline flexibility means that one can wait for at most two additional hours to receive the charged electric vehicle. This flexibility is applicable to both start and finish times of appliance operation, since the DRS can shift the starting time with postponing the finishing time. However, it should finish the appliance operation at latest at the provided flexibility. This kind of flexibility facilitates the DRS by shifting or even by interrupting appliances. The DRS considers the time difference between the provided deadline flexibility of each appliance and current time interval before shifting it to another one. Equation (5) indicates this hard constraint:

$$RL_{i,j}^t \leq \left(DF_{i,j} - t\right). \quad (5)$$

$RL_{i,j}^t \in \mathbb{Z}^*$ relates to the total remaining load requests of j^{th} appliance of i^{th} smart house from t^{th} interval until the end of its cycle. $DF_{i,j}$ (time instant) denotes the provided deadline flexibility of that specific appliance. The DRS examines each of the received load requests for satisfying this constraint. If the total remaining load requests of an appliance is still less than the time difference between the provided deadline flexibility and current time interval t, then, the DRS can decide to whether allow it to start or to continue in current time interval or shift it to another time interval.

Aggregated consumption threshold: Many electricity producers are experiencing a deficit of electricity generation capacity in consequence of load requests by consumers. More accurately, the generated amount of electricity is often unable to satisfy the requested loads in a specific time interval. Hence, the DSO currently applies an $ECT \in \mathbb{R}^+$, to ease grid stability [11]. ECT (watt) is a soft constraint over the time intervals, in which it sometimes cannot be satisfied due to the provided deadline flexibilities and uninterruptibility feature of some appliances. Therefore, the DRS can only apply this constraint on the remaining shiftable load requests including those which: I) have not started yet, and II) have started earlier but corresponding appliances are interruptible. This threshold, as formulated in (6), attempts to keep the aggregated consumption of allowed load requests in each time interval under the provided ECT:

$$\sum_{i=1}^{N} \sum_{j=1}^{A} \left(EC_{i,j}^t \times x_{i,j}^t\right) \leq ECT. \quad (6)$$

C. Scenario

Each smart house H_i has a specific scenario including usage pattern of its appliances. The DRS is unaware of the scenario details of all smart houses before scheduling and it continuously receives load requests over time. Table I exhibits a sample scenario including the desired schedule of appliances of a smart house.

TABLE I. A SAMPLE SCENARIO OF A SMART HOUSE

Start	End	Activity description	Deadline	Dependency	Priority
00:00	24:00	Using the refrigerator.	24:00	--	Infinite
08:00	24:00	Turning the lights on.	24:00	--	Infinite
08:05	09:50	Using the dish washer.	10:30	--	0.2158
08:40	10:00	Using the washing machine.	10:30	--	0.1063
11:00	11:50	Using the laundry dryer.	12:30	Washing machine	0.1499
11:30	22:40	Using the computer.	23:30	Lighting	0.2649
19:50	22:00	Watching the TV.	24:00	Lighting	0.1293
20:00	22:00	Charging the electric vehicle.	24:00	--	0.1338

For instance, the consumer of the smart house provides two hours of deadline flexibility to the DRS from charging the electric vehicle. The DRS receives the first load request of the electric vehicle at 20:00. Therefore, the DRS has an opportunity to deliver the charged electric vehicle until 24:00 by shifting and interrupting the charging procedure, since the electric vehicle is a member of the interruptible appliances. Furthermore, there is a consecutive dependency between the laundry dryer and the washing machine. Moreover, a concurrent dependency exists between the personal computer (or the television) and the lighting system. It is worthwhile to note that only shiftable appliances have priority among each other. Hence, the refrigerator and lighting will not undergo any scheduling procedure, since they are members of the non-shiftable appliances. Therefore, they receive infinite priority.

IV. Local Load Scheduling Optimization Algorithm

The DRS continuously applies the load scheduling optimization algorithm on the received load requests to produce a specific schedule for appliances of each smart house based on the aforementioned objective and constraints in each time interval. Apart from *ECT*, the DRS permits the non-shiftable loads to do their duty over time. Furthermore, if there are uninterruptible appliances which have been allowed to start or to continue in the previous time interval, they should be granted to continue. Finally, if there would be a load, where shifting it to the next time interval will exceed its provided deadline flexibility, again, it should be permitted to start or to continue. After completing these procedures, the scheduling algorithm will check whether total consumption of the remaining loads is less than the remaining *ECT*. If so, all will be allowed to start or to continue their procedure. Otherwise, the algorithm calls the Knapsack procedure to permit a subset of loads from the remaining ones to start or to continue, and to shift the unpermitted loads to the next time interval.

A. The Knapsack Problem

The Knapsack problem is a traditional problem of Computer Science in combinatorial optimization literature [12]. Given $M \in \mathbb{N}$ items, the Knapsack packs the items to get the maximum total value, where each indivisible item has a weight and a value. The Knapsack has a fixed capacity named $C \in \mathbb{R}^+$. The Knapsack problem is a weakly NP-Complete problem since the time complexity of solving it in a brute-force manner is $O(2^M)$. This method calculates all feasible subsets in order to find the optimal one. In the load scheduling problem, the priority of corresponding appliances of the remaining loads matches with the items in the Knapsack problem. In addition, weights correspond to the electricity consumption of the remaining loads. The objective in the Knapsack problem is to maximize the total value, whereas in the load scheduling problem, the objective is to maximize the total number of allowed loads in each time interval. Finally, the Knapsack capacity corresponds to the remaining *ECT*. In the load scheduling problem, the DRS should select and allow those loads which optimize the objective and satisfy the constraints thoroughly. In summary, the Knapsack problem is reducible to the load scheduling problem. As a result, the load scheduling problem is also a weakly NP-Complete problem.

The Knapsack procedure receives the remaining loads and calculates the fitness of produced feasible subsets, where each subset comprises some loads. In conclusion, the outcome of this approach is a subset of remaining loads which should be allowed to start or to continue in this time interval. Obviously, there will probably be some loads which are not permitted to start or to continue. These loads should be shifted to the next time interval. In addition, in order to decrease the computation time, Dynamic Programming approach with $O(MC)$ time complexity has been applied to solve the Knapsack problem, when required [7], [8]. Algorithm 1 describes the procedure of the proposed local load scheduling optimization algorithm.

Inputs : Scenarios, flexibilities, classification of appliances, *ECT*.
Output: Schedule of appliances of all smart houses.

Preprocessing the input data;
While *receiving load requests in specific time intervals* **do**
  Start or continue the non-shiftable loads;
  Continue uninterruptible loads, which have started previously;
  Start or continue loads which cannot be shifted or interrupted anymore due to their provided deadline flexibility;
  If *there are some remaining load requests* **then**
    If *their total consumption is less than the remaining ECT* **then**
      Allow them to start or to continue;
    Else
      Call the Knapsack procedure, allow the output loads to start or to continue, and shift unpermitted loads to the next interval;
    End
  End
End

Algorithm 1. Pseudo code of the local load scheduling optimization algorithm

V. Experimental Setup and Analysis

This section first describes the experimental setup including analysis criterion and data types. Subsequently, the experimental results will be clarified precisely.

A. Experimental Setup

The proposed algorithm has been implemented with MATLAB® R2014b on a computer with an Intel Core i7 2.0 GHz CPU, and 6 GB memory. Load profiles of appliances, shown in Fig. 1, have been captured from the TraceBase open repository which comprises power traces of electrical appliance [13]. To simplify the experiments, consumers will operate their appliances once a day. Furthermore, operation time of appliances (based on the load profiles of TraceBase database), their deadline flexibility, and priority have been randomly selected. The DRS continuously receives load requests of $N = 100$ smart houses in 5-minutes time intervals over a 24-hour period. Each load request only includes its required electricity consumption for the next five minutes except the first load request, which additionally comprises its shiftability and interruptibility memberships, priority, deadline flexibility, and dependency status.

The results will be analyzed based on variations of *ECT*. *ECT* is constant over time and will be 20%, 40%, 60%, or be 80% of the maximum aggregated consumption of all smart houses at the peak time interval. It is beneficial to note that although the DRS is unaware of all scenarios prior to starting the schedule, however, the DSO grid stability management informs the DRS about *ECT* based on the forecasted or learnt scenarios of previous days. If *ECT* equals to or is greater than the maximum peak demand, no scheduling is needed. Hereinafter, each *ECT* percentage is based on the informed maximum peak demand.

B. Experimental Results

Fig. 2 depicts the scheduled aggregated consumption of 100 scenarios with respect to changes in *ECT*. The period from 10:45 to 00:45, in which the aggregated consumption approaches *ECT*, is selected. The peak electricity consumption equals to almost 289 kW at 20:30, when *ECT* is 100%. In addition, the average aggregated consumption is almost 56 kW in a 24-hour day. The DRS desires to reach a point, in which there are as few peak times as possible. More in details, as Fig. 2 demonstrated that, when *ECT* is 20%, the scheduler

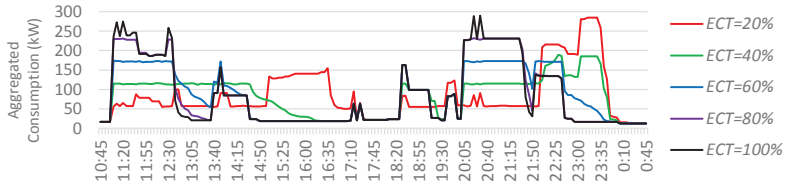


Figure 2. Aggregated consumption of 100 scenarios with respect to changes in *ECT*

has been successfully flattened the aggregated consumption between almost 10:45 to 14:30. However, this flattening causes another peak period from almost 14:45 to 16:45. The reasons are the uninterruptibility feature and deadline flexibility of some appliances. This is also true for the future time intervals. Furthermore, when *ECT* is 20%, the Knapsack cannot find any solution for the remaining loads in some time intervals due to the low remaining *ECT*. Consequently, the DRS has to shift all the remaining loads to the next time interval. According to the deadline flexibility constraint, the DRS must allow some loads to start or to continue their operation apart from the remaining *ECT* in the next time intervals which produces another peak time. By applying this threshold (20%), aggregated electricity consumption at the peak time (almost 284 kW) decreases only 1.73% comparing with the maximum peak demand (almost 289 kW).

Nonetheless, when *ECT* is 40%, the Knapsack procedure can permit most of loads to start or to continue their work in corresponding time interval, and accordingly, the DRS should shift only a few remaining loads to the next interval. It will decrease the aggregated consumption at peak times (almost 189.4 kW) and flatten the aggregated consumption by 34.46%, as represented in Fig. 2. In addition, if *ECT* increases up to 60%, the peak reduction ratio will be 40%. Ultimately, no significant achievement was found from increasing *ECT* to 80%. The reason is that most of load requests are permitted to start or to continue their operation at the time they request. In this situation, the peak reduction ratio is only 20.17%.

Table II analysis the number of referrals to the Knapsack procedure (*K-Referrals*), average deviation between appliance deliverance and reception times (*T-Deviation*), maximum required *ECT* (*M-ECT*), and peak-to-average ratio (*PAR*).

TABLE I. ALGORITHM ANALYSIS BASED ON *ECT* VIOLATION

	ECT violation				
	57.8 kW (20%)	115.6 kW (40%)	173.4 kW (60%)	231.2 kW (80%)	289 kW (100%)
K-Referrals	56	68	48	12	0
T-Deviation	220	115	30	5	0
M-ECT	284 kW	189.4 kW	173.4 kW	231.1 kW	289 kW
PAR	5.07	3.38	3.09	4.12	5.16

According to the *K-Referrals*, if *ECT* equals to 20%, the Knapsack procedure will be called for 56 times during the whole schedule. This number will raise if *ECT* increases up to 40%. The reason is the existence of some intervals that the DRS allows some shiftable and interruptible loads to start or to continue their operation. Furthermore, there will be some new load requests in the next time intervals. Hence, the aggregated consumption of new and shifted load requests will be more than the remaining *ECT* and accordingly, the number of referrals to the Knapsack procedure increases. Nevertheless, this number will decrease when *ECT* is equal to 60%. The reason is the large number of load requests that can be allowed to start or to continue in each time interval with respect to the assigned *ECT*. In addition, some of these loads, permitted to start or to continue, are members of the uninterruptible appliances. Hence, in the next time intervals, apart from the remaining *ECT*, they must be allowed to continue their duty. Therefore, the number of referrals to the Knapsack procedure will decrease. This is also applicable when *ECT* is 80%. Obviously, there is no need to call the Knapsack procedure when the assigned *ECT* is 100%.

Considering the *T-Deviation* values in Table II, the DRS has the opportunity to shift and interrupt some loads to satisfy the constraints based on the provided deadline flexibility of appliances. Hence, some consumers will confront reception delay of their appliances. For instance, although one consumer desires to receive his charged electric vehicle at 18:00, however, he has provided two hours flexibility to the system. It means that he can receive it in the worst case at 20:00. Now, after scheduling the charging process of the electric vehicle, the consumer observes that he has received the charged electric vehicle at 18:45. Therefore, there is 45 minutes reception delay. As a result, according to Table II, consumers will averagely receive their appliances in the completed status with 220 minutes delay. This average delay decreases when *ECT* increases. The fact is that appliances will be permitted to start their operation at the time they request. Most delays are related to the electric vehicle since it is a member of the interruptible appliances and it consumes more than other appliances according to its load profile.

With respect to Table II, *M-ECT* denotes the aggregated consumption at the peak time, when *ECT* changes. What is interesting is that if assigned *ECT* is 60%, the corresponding *M-ECT* is less than the assigned *ECT* (173.4 kW). It means that the DRS is entirely successful in flattening the aggregated consumption, even below the assigned *ECT*. Nevertheless, this fact is not feasible, where *ECT* is equal to 20% or to 40%. In these two circumstances, the DRS has to exceed the assigned threshold since it should allow the non-shiftable and previously-started uninterruptible loads to continue apart from the remaining *ECT*. Furthermore, since the *T-Deviation* is not averagely too high when *ECT* is 60%, it is not mandatory to

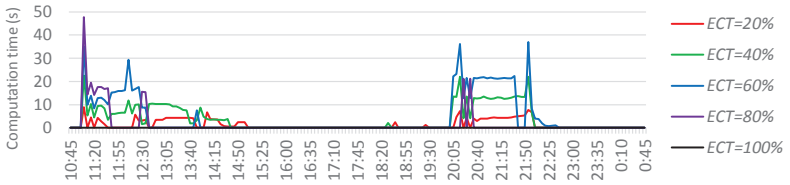


Figure 3. Computation time of running the algorithm with respect to changes in *ECT*

assign a higher *ECT*, e.g., 80%. Accordingly, the value of *PAR* decreases, when *ECT* is up to 60%.

As the final analysis, Fig. 3 demonstrates more time complexity when the assigned *ECT* increases. Nonetheless, number of time intervals, in which the Knapsack procedure should run, decreases. Having some uninterruptible appliances and the deadline flexibility constraint are reasons of this decrease. If the DRS allows uninterruptible loads to start in a time interval, it has to shift more loads in the next time intervals since the remaining *ECT* in that time intervals will be low. These shifted loads will be accumulated and eventually the Knapsack procedure confronts a large number of remaining loads in a time interval. As a final note for this analysis, running the local load scheduling algorithm employs only 5% of CPU bandwidth with almost 350 megabytes of memory, when *ECT* is 20% to 80%. Obviously, there is no influential computation time when *ECT* is 100%.

VI. Conclusion and Future Work

In this paper, load requests of appliances of smart house are scheduled by a Demand Response System (DRS) that runs as a centralized control by e.g., an aggregator actor in the electricity market. Based on this control, this paper has investigated a DR strategy incorporating a local load scheduling optimization algorithm. Each consumer provides a daily scenario describing the usage pattern of his appliance. Appliances of each smart house are classified based on the shiftability feature. Those appliances which can be shifted to other time intervals have further been divided based on the interruptibility feature. Each shiftable appliance has a hard deadline flexibility constraint provided by the consumer. Furthermore, some appliances have concurrent or consecutive dependency relationship among themselves imposing a hard constraint on the DRS in scheduling. Consequently, a local load scheduling optimization algorithm based on the Knapsack concept for solving the scheduling problem has been proposed. The Knapsack procedure in some time intervals attempts to select a subset of remaining loads to permit them to start or to continue, and shift the unpermitted ones to the next time interval. The main objective applied in this algorithm is maximizing satisfaction of consumers while aiming at peak demand reduction. Results considering an aggregation of 100 smart houses, indicate that the application of an electricity consumption threshold results in flattening of the aggregated consumption and leads to a decreased peak demand. As future work, we are planning to apply learning and forecasting approaches to enhance the load scheduling strategy while improving the system model by utilizing multi-objective optimization techniques in order to incentivize consumers more to actively participate in DR programs.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework program (FP7/2007-2013) under grant agreement n° 619560 (SEMIAH).

References

[1] J. Gao, Y. Xiao, J. Liu, W. Liang, and C. Chen, "A survey of communication/networking in smart grids," Future Generation Computer Systems, vol. 28, no. 2, pp. 391-404, 2012.

[2] A. Soares, A. 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.

[3] R. Belhomme, R. Cerero, G. Valtorta, and P. Eyrolles, "The ADDRESS project: Developing Active Demand in smart power systems integrating renewables." *Power and Energy Society General Meeting, 2011 IEEE*.

[4] M.H. Albadi and E.F. El-Saadany, "A summary of demand response in electricity markets", Electric Power Systems Research, vol. 78, no. 11, pp. 1989–1996, 2008.

[5] K. C. Sou, J. Weimer, H. Sandberg, and K. H. Johansson, "Scheduling smart home appliances using mixed integer linear programming," in Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on., pp. 5144-5149.

[6] J. W. Lee and D. H. Lee, "Residential electricity load scheduling for multi-class appliances with time-of-use pricing," in GLOBECOM Workshops (GC Wkshps), 2011 IEEE, pp. 1194-1198.

[7] N. Kumaraguruparan, H. Sivaramakrishnan, and S. S. Sapatnekar, "Residential task scheduling under dynamic pricing using the multiple knapsack method," in Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES. pp. 1-6.

[8] O. A. Sianaki, O. Hussain, and A. R. Tabesh, "A knapsack problem approach for achieving efficient energy consumption in smart grid for endusers' life style," in Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES), 2010 IEEE Conference on, pp. 159-164.

[9] K. Kok, S. Karnouskos, D. Nestle, A. Dimeas, A.Weidlich, C.Warmer, P. Strauss, B. Buchholz, S. Drenkard, N. Hatziargyriou et al., "Smart houses for a smart grid," in Electricity Distribution-Part 1, 2009. CIRED 2009. 20th International Conference and Exhibition on. IET, 2009, pp. 1-4.

[10] X. He, L. Hancher, I. Azevedo, N. Keyaerts, L. Meeus, and J.-M. GLACHANT, "Shift, not drift: towards active demand response and beyond, THINK project, EU FP7," 2013.

[11] M. T. Beyerle, J. A. Broniak, J. M. Brian, and D. C. Bingham, "Manage whole home appliances/loads to a peak energy consumption," Mar. 8 2011, US Patent App. 13/042,550.

[12] H. Kellerer, U. Pferschy, and D. Pisinger, "Knapsack problems," Springer Science & Business Media, 2004.

[13] [Online]. Available: http://www.tracebase.org/

Paper B

Home Appliance Load Scheduling with SEMIAH

The paper presented in this chapter has been accepted as a conference publication [2].

[2] Rune Hylsberg Jacobsen, Armin Ghasem Azar, Qi Zhang, and Emad Samuel Malki Ebeid, "Home Appliance Load Scheduling With SEMIAH," In *Fourth International Conference on Smart Systems, Devices, and Technologies (SMART)*, 2015, pages 1-2, Link to paper

Authors contribution: Rune Hylsberg Jacobsen and Armin Ghasem Azar conceived the initial design of the demand response system; Armin Ghasem Azar formulated the load scheduling system; ; Rune Hylsberg Jacobsen performed the simulations and Armin Ghasem Azar contributed to the analysis of simulations; Rune Hylsberg Jacobsen was the main editor of the manuscript; Armin Ghasem Azar, Qi Zhang, and Emad Samuel Malki Ebeid contributed to the reviewing process.

Home Appliance Load Scheduling with SEMIAH

Rune Hylsberg Jacobsen, Armin Ghasem Azar, Qi Zhang, and Emad Samuel Malki Ebeid
Department of Engineering, Aarhus University, Denmark
Email: {rhj, aga, qz, esme}@eng.au.dk

***Abstract*—The European research project SEMIAH aims at designing a scalable infrastructure for residential demand response. This paper presents the progress towards a centralized load scheduling algorithm for controlling home appliances taking power grid constraints and satisfaction of consumers into account.**

Keywords–Smart grids, demand response, load scheduling

I. INTRODUCTION

Demand Response (DR) is in its nascent stage in Europe. DR programs allow Distribution System Operators (DSOs) to reduce electricity peak demand by incentivizing consumers to adapt their usage to variations in the electricity generation [1]. Existing DR programs aim at large industrial consumers, who can be managed as one large client, representing an aggregated demand of hundreds of residential households. Despite the fact that households constitute 27% of the total energy consumption in Europe and are responsible for 10% of the CO_2 emissions, no automated DR programs have been implemented for European households. The European FP7 research project *SEMIAH* (Scalable Energy Management Infrastructure for Aggregation of Households) strives for developing an Information Communication Technology (ICT) infrastructure for DR [2]. SEMIAH enables shifting of energy consumption to periods with high electricity generation from Renewable Energy Sources (RESs) which helps DSOs to flatten the peak electricity demand.

SEMIAH undertakes three different approaches to address the home appliance load scheduling optimization problem as follows: 1) scheduling of non-critical power-intensive loads using a residential Home Energy Controlling Hub (HECH) system, 2) two-stage linear stochastic programming for scheduling of domestic loads, and 3) load scheduling with multi-objective optimization techniques. This paper introduces a single-objective load scheduling optimization as a precursor for the latter multi-objective optimization approach.

II. THE SEMIAH SYSTEM

The SEMIAH system employs a centralized approach for aggregation and scheduling of load demands of appliances. It relies on the flexibilities provided by households who decide to join a DR program. The *flexibility* concept of SEMIAH aligns with the European mandate M/490 [3]: "The flexibility [offering] concept assumes that parties connected to the grid produce offerings of flexibility in load and (distributed) generation. Thereby, so-called flex-offers are issued indicating these power profile flexibilities, e.g., shifting in time or changing the energy amount. In the flex-offer approach, consumers and producers directly specify their demand and supply power profile flexibility in a fine-grained manner (household and SME level)." In SEMIAH, flexibility from home appliances

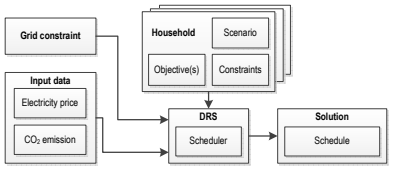


Figure 1. Conceptual diagram of the demand response serving subsystem.

are aggregated in a coherent way to produce flex-offers that can be traded in the electricity markets.

Load demands of appliances can be categorized based on the *shiftability* feature [4]. Shiftability means to authorize a DR System (DRS) to shift load requests of *shiftable* appliances to a future time interval. Some appliances cannot be shifted, for instance the refrigerator. Hence, these become members of the category of *non-shiftable* appliances. Shiftable appliances can be further divided into groups based on the *interruptibility* feature. As an example, the DRS can both shift and interrupt the charging cycle of an electric vehicle. However, it should continue operation of the *uninterruptible* appliances until completion when these are started, e.g., a washing machine. Each household presents a scenario including the usage schedule of appliances. The household applies a *deadline flexibility* constraint, which sets a contract when a given appliance must complete its operation at latest. Subsequently, the DRS produces a schedule for the aggregated set of appliances, i.e., a solution. The deadline constraint imposes a non-trivial optimization problem for the scheduling of electricity loads.

Fig. 1 illustrates a conceptual diagram of the load scheduling subsystem. The DRS applies input data from the electricity market and the bulk generation side to establish an objective function used by the scheduling algorithm. In the household, a HECH is installed to manage loads of appliances. The HECH connects to sensors and actuators of the household by using ZigBee communication. It receives control information from the DRS and runs the scheduled appliances accordingly.

III. LOAD SCHEDULING

The DRS schedules and manages appliances based on desired scenarios of households. When consumers provide their appliances in the "DR Ready" mode to the DRS, they authorize the DRS to schedule appliances in a 24-hour period. The DRS receives load requests from all presented scenarios in each time interval of 5 minutes. Consecutively, it runs the scheduling algorithm on load requests taking the shiftability and interruptibility features of appliances into account. Three constraints are assumed by the scheduler: 1) keeping the total

 ISBN: 978-1-61208-414-5

power consumption below a specific *Electricity Consumption Threshold (ECT)*, 2) satisfying the deadline flexibility of appliances, and 3) satisfying the *dependencies* between appliances, e.g., the laundry washing is completed before drying can start. The first constraint relates to the grid stability. The second and third constraints impact on satisfaction of consumers.

The scheduling algorithm allows non-shiftable loads to start or to continue their operation. If there are uninterruptible loads running in the previous time interval, they are permitted to continue. When there are loads which cannot be shifted without violating the deadline constraint, they must start or continue. Afterwards, the algorithm utilizes a Knapsack approach [5] on the remaining load requests to calculate the fitness of subsets. It returns a subset of remaining load requests to start or to continue in the current time interval. Loads, which cannot be started, are shifted to the next time interval.

IV. Preliminary Results

Table I offers an example of a scenario from a household with a consumer returning home at 18:00 and commencing to operate his appliances. The corresponding scheduled load demands of the household is demonstrated in Fig. 2 using two different *ECT*s. The maximum demand occurs at 18:25 and equals to 8,940 W. It comprises the electric vehicle, lighting, washing machine, oven, and stove. The day-ahead market is utilized for electricity price data (www.nordpoolspot.com). The CO_2 emission rate is derived from the electricity generation mix (www.energinet.dk) using the Danish power grid as the case study. To arrive at a *cost metric*, combining electricity price and CO_2 emission cost, an average cost of CO_2 emission of 171.78 DKK/1,000 kg is used. No shifting occurs when *ECT* is 9 kW which is higher that the peak demand of the household. When the threshold is lowered to 3 kW, load shifting takes place. The DRS decides to shift the charging of the electric

TABLE I. An example of a household scenario.

Start	End	Activity description	DF	P_p [W]
18:00	23:00	Turning the **lights** on.	23:00	100
18:00	20:00	Plugging the **electric vehicle** in its station.	23:00	3,600
18:05	19:50	Running the **washing machine**.	23:00	2,000
18:10	18:50	Preparing food and turning the **oven** on.	22:15	2,350
18:20	18:50	Starting and using the **stove**.	22:15	840
19:00	19:45	Eating the food while watching **TV**.	23:00	55
21:30	23:00	Preparing the **laundry dryer**.	23:00	2,000

DF and P_p are the the deadline flexibility and the peak power consumption of appliances (marked with **bold** type face) , respectively.

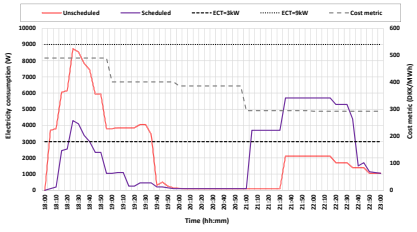


Figure 2. Peak demand shifting of home appliances due to *ECT* constraint. The cost metric (from 4 Nov. 2014 data) indicates a decreasing trend.

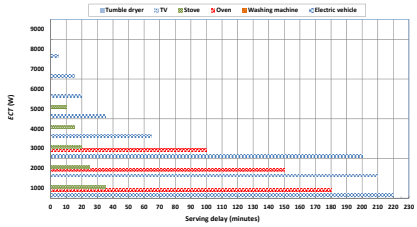


Figure 3. As *ECT* decreases, the serving delay of appliances increases.

vehicle by 200 minutes, and operations of the stove and oven by 100 and 20 minutes, respectively. It is beneficial to note that the threshold cannot be fully satisfied due to non-shiftable appliances that must run. This implies a "softness" of *ECT*.

To study consumer satisfaction, Fig. 3 examines the deviation between the starting and the serving times of appliances. Obviously, consumers prefer minimal deviation between the provided scenario and the offered schedule. As *ECT* increases, the consumer gets closer to the desired scenario. In the example, the electric vehicle is the best candidate to be shifted to later time intervals due to its higher peak power consumption.

V. Conclusions and Future Work

The SEMIAH project aims at developing an infrastructure for DR enabling aggregation and scheduling of electricity loads of home appliances. A scheduling algorithm based on a single-objective optimization approach has been developed. It allows the shifting of loads according to flexibilities provided by consumers. As future work, the scheduling algorithm will support multi-objective optimization techniques coupling with the divergent priorities of consumers and the DSO. SEMIAH targets a solution that scales to 200,000 households to produce aggregated flex-offers tradable in the electricity markets.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework program (FP7/2007-2013) under grant agreement n° 619560 (SEMIAH).

References

[1] M.H. Albadi and E.F. El-Saadany, A summary of demand response in electricity markets, Electric Power Systems Research, vol. 78, no. 11, pp. 1989-1996, 2008.

[2] R. H. Jacobsen and E.S.M. Ebeid, SEMIAH: Scalable Energy Management Infrastructure for Aggregation of Households, 40th EUROMICRO conference on Software Engineering and Advanced Applications and 17th EUROMICRO conference on Digital System Design, 2014.

[3] CEN-CENELEC-ETSI Smart Grid Coordination Group – Sustainable Processes. Technical report, CEN-CENELEC-ETSI Smart Grid Coordination Group, November 2012.

[4] 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.

[5] H. Kellerer, U. Pferschy, and D. Pisinger, Knapsack problems, Springer Science & Business Media, 2004.

 ISBN: 978-1-61208-414-5

Paper C

SEMIAH: An Aggregator Framework for European Demand Response Programs

The paper presented in this chapter has been accepted as a conference publication [3].

[3] Rune Hylsberg Jacobsen, Dominique Gabioud, Gillian Basso, Pierre-Jean Alet, Armin Ghasem Azar, and Emad Samuel Malki Ebeid, "SEMIAH: An Aggregator Framework for European Demand Response Programs," In *IEEE Euromicro Conference on Digital System Design (DSD)*, 2015, pages 470-477, doi: 10.1109/DSD.2015.96

Authors contribution: Rune Hylsberg Jacobsen conceived the initial design of the SEMIAH system; Rune Hylsberg Jacobsen was the main editor of the manuscript; Armin Ghasem Azar, Dominique Gabioud, Gillian Basso, Pierre-Jean Alet, and Emad Samuel Malki Ebeid contributed with the text in selected areas and to the reviewing process.

2015 Euromicro Conference on Digital System Design

SEMIAH: An Aggregator Framework for European Demand Response Programs

Rune Hylsberg Jacobsen*, Dominique Gabioud†, Gillian Basso†, Pierre-Jean Alet‡,
Armin Ghasem Azar*, and Emad Samuel Malki Ebeid*
*Department of Engineering, Aarhus University, Denmark. Email: {rhj, aga, esme}@eng.au.dk
†Institute of Systems Engineering, University of Applied Sciences and Arts Western Switzerland, Sion, Switzerland.
Email: {dominique.gabioud, gillian.basso}@hevs.ch
‡CSEM SA, Neuchâtel, Switzerland. Email: pierre-jean.alet@csem.ch

***Abstract*—Smart grids offer an indisputable business opportunity for system operators and energy traders to engage in demand response programs. Hereby these actors may profit from trading flexibility provided at the prosumer side on the energy markets. This paper discusses the system design challenges for an information and communication technology system infrastructure facilitating DR for residential prosumers. It presents the SEMIAH framework for providing a scalable infrastructure for residential DR built on a component-based architecture with the virtual power plant at the hearth of the system. The paper examines the possible impact of deploying an automated residential DR program on the quality and stability of a low voltage grid.**

***Index Terms*—Smart grids, demand response, aggregator, infrastructure framework.**

I. Introduction

European countries are progressing towards the development of smart grid concepts for the establishment of an efficient market for trading flexibility in electricity consumption and production [1]. The concept is based on a wholesale model, which determines the future roles of actors in the electricity market [2]. Today, energy markets, such as the Nord Pool Spot[1], offer intraday trading across different regions in Europe to act as a balancing market to the day-ahead markets. The intraday market offers opportunities for risk reduction as well as increased profit by giving access to a wide selection of counterparts with different consumption and production mix, marginal costs, etc. subjected to general market conditions.

Presently, most trading of flexibility takes place bilaterally between companies that can interrupt their power consumption for some periods, and power grid system operators, possibly through an electricity trading company. Rather than investing in grid expansions, the system operators in some countries can pay, through market agreements, large electricity users to reduce the consumption in concerned hours so that the congestion in the grid is avoided. To realize the potential of providing flexibility, *aggregators* are needed to pool offers of reduced and shifted electricity demands into aggregated offers to the electricity market or to system operators [3]. Upon market acceptance, the aggregators actuate the flexible consumption according to a defined schedule by shifting electricity demand to meet the contractural obligations of the offer.

Demand Response (DR) provides an opportunity for prosumer to play an active role in the operation of the electricity grid by reducing or shifting their electricity usage during peak periods in response to an external trigger signal. Prosumers, that engage in DR, offer flexibility for certain types of appliances and processes by trading off convenience in daily practices and comfort e.g., the indoor temperature range of a building. The realization of DR requires an Information and Communication Technology (ICT) platform that provides management, aggregation, and scheduling of a large number of appliances with flexible prosumption [4]. Furthermore, the aggregator service platform has the potential of providing DR to be traded both on the existing wholesale markets and on new retail markets for system operators. Hereby electricity trading companies and other service providers have a central role in promoting concrete flexibility products and services that are connected to a flexible management of prosumption towards savings on energy bills.

DR programs can incentivize Distribution System Operators (DSOs) by taking into account the Low Voltage (LV) grid constraints such as power quality and power demand limits. First, it supports the grid operators and balance responsible providers in balancing the grid. Second, it provides an assest for flatterning peak demand and hence has the potential of postponing grid infrastructure investments. In this context, the European research project SEMIAH (Scalable Energy Management Infrastructure for Aggregation of Households) aims to develop a novel and open smart grid infrastructure for the implementation of automated DR in households [5]. To validate this new infrastructure and for assessing the potential impacts, a large-scale simulation of up to 200,000 households will be performed. In addition, the SEMIAH concepts will be validated with a total of 200 households running as a pilot in Norway and Switzerland.

Fig. 1 shows an overview of the SEMIAH aggregator infrastructure. The infrastructure uses a centralized approach where a *back-end* system, hosted by the aggregators, runs the aggregation, scheduling and actuation of electricity loads from households. The back-end connects to a number of *front-end* Home Energy Management Systems (HEMSs) to act as gate-

[1]Accessible at: www.nordpoolspot.com

978-1-4673-8035-5/15 $31.00 © 2015 IEEE
DOI 10.1109/DSD.2015.96

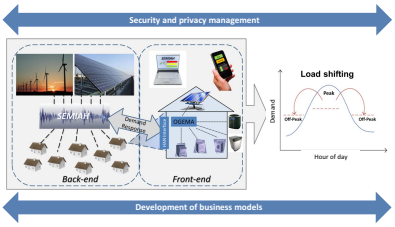


Fig. 1: SEMIAH system overview. The SEMIAH back-end coordinates a demand response from a large set of front-end systems to provide peak flattening by load shifting.

ways and proxies for actuation of home appliances. Security and privacy protection are managed end-to-end between the back-end and the front-end. New business models are subject of further studies as these are essential for a commercialization of an aggregator service provisioning.

This paper addresses the challenges of residential DR. It presents an open framework for a scalable aggregator infrastructure that allows the aggregation of flexible consumption from a large set of households. The paper is organized as follows. Section II gives a brief background of DR. Section III presents the SEMIAH system model. Section IV presents the aggregator service framework. The impacts on the LV grid stability is discussed in Section V. Finally, the paper is concluded in Section VI.

II. Demand Response Background

DR is defined as "changes in electric usage by demand-side resources from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized." [6]. A survey of DR architectures and load management algorithms including ICT architectures for enabling DR programs is provided in [7]. A study of ongoing technology trends, opportunities, products, communications, DR efforts in residential homes are presented in [8]. Potentials, benefits, enabling technologies and DR systems are described and discussed in [9]. An overview of various types of demand side management analysis including DR and an outlook with the latest demonstrations are presented in [10]. Several smart grid projects in Europe are introduced in [11].

DR technology offers several benefits all over the power grid. Globally, DR programs can support a more efficient integration of Renewable Energy Sources (RES) in the grid an hereby contribute to a future low carbon economy. Locally, DR can improve the stability of the distribution grid, managing the local production as well as controlling consumption. Moreover, peak flattening reduces the need for investments in grid expansions.

The ability to control flexible electricity loads in households with DR implies great impact for the grid by improved grid control and for the energy markets through market optimizations [12]. The various DR programs govern the controlling of electricity load between the utility and the prosumer when they are applied. Load management is on the prosumer side, and there can be either load reduction or load shifting, while there is only load shifting on system operator side. These service providers install switches to control the loads and communicate directly with the switch without engaging the asset or facility owner. In some cases, the controlling entity can also send control signals to a home automation system that can affect the control action.

A. Control Strategies

The operation of a household or a building follows the principles of optimal control. The objective is typically to run the household or building with high energy-efficiency taking into account comfort and economic constraints of the residents. DR is generally triggered by varying peak demand capacity or high prices at the wholesale level [3].

Incentive-based or event-driven DR can be invoked in response to a variety of trigger conditions, including environmental parameters (e.g., temperature); local or regional grid congestion; economics; or operational reliability requirements [6]. For DR programs that foster an improved integration of renewable energy, aggregators can shift consumption to periods with lower CO_2 intensity. The Load Research Committee of Association of Edison Illuminating Companies (AEIC) suggests a simple percentage-based calculation for average demand shifting [13]. It "estimates the total energy shifted from peak hours to off-peak hours by calculating the difference in on-peak energy usage between the consumption baseline and the participants' load shape." This definition provides a mean to quantify the effect of a DR action. Triggers from economics can for example be spikes in the wholesale electricity price. Other triggers can derive from reliability requirements such as the risk of a blackout caused by a major power plant tripping offline.

B. Renewable Energy Utilization

The Renewable Energy Utilization (REU) identifies the energy generation mix of the provided power by defining the Effective Substitution Ratio ESR for substituting renewable energy Q_{res} for conventional energy Q_{ces} [14]. This can be done by using the REU, which is expressed by:

$$ESR = \frac{Q_{res}}{Q_{res} + Q_{ces}} (1 - \eta) \quad (1)$$

The coefficient η takes into account any differences in the efficiency of the conversion of energy [14]. Fig. 2 shows the energy production data for 5 days of the Danish power grid system[2]. The top figure displays the RES generation, the

[2]Sum of DK1 and DK2 areas of the Nord Pool Spot market.

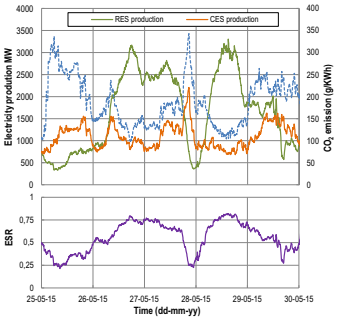


Fig. 2: RES generation, CO_2 emission intensity, and the Effective Substitution Ratio (ESR) over a 5-day period from 25 May to 29 May 2015. Data has been provided by Energinet.dk.

conventional, fossile fuel based generation (CES generation), and the CO_2 emission intensity. The bottom figure shows the ESR metric. The correlation between the CO_2 emission intensity and ESR has been calculated to -0.80 for the 5-day period. By using CO_2 as a trigger, DR programs can improve the integration of RES in the power grid.

C. Appliance Load Forecasting and Scheduling

Load forecasting is an essential step in the planning of a reliable electrical grid operation. Forecasting refers to a prediction of the future electricity load based on short-, medium-, or long-term periods, to prevent the electricity distribution infrastructure from unforeseen electricity outages. Although accurate models for electric power load forecasting are indispensable, load forecasting facilitates the electricity provides and DSOs by adopting important decisions on both electricity distribution and scheduling [15]. DR programs attempt to alleviate the power system stresses through encouraging the prosumers to voluntarily modify their daily electricity prosumption behavior in order to decrease the peak demand while maintaining the prosumers' comfort level [16].

Encouraging participation in DR programs requires some incentives for the prosumers. The main idea behind DR programs is to follow the fluctuations of the electricity generation over time by reducing the consumption during periods of RES generation and/or shifting consumption to off-peak periods [17]. As a consequence of the discrepancy between the predictions and prosumer demands, appliances also require a mechanism to respond to prosumers' load requests. High-potential and scalable load scheduling approaches play a key role in reducing the peak demand. Load scheduling scheme tries to effectively schedule the electricity load requests of various domestic *smart appliances* over time. The load scheduling should take into account various system objectives and constraints imposed. Therefore, the load scheduling faces a complex constrained stochastic problem that should be solved continously in time [12].

D. Home Automation as an Enabler for Demand Response

Home automation technologies are designed to enhance the quality of life of occupants. These technologies use wired or wireless solutions to interconnect the home's smart devices (e.g., sensors and actuators). Conservative solutions make use of power lines for establishing the communication link between such devices. Such wired installations easily become expensive and complex [18]. Other solutions, established by wireless communication standards, provide a degree of flexibility, interoperability and cost-effectiveness. However, some wireless technologies require high-power consumption which make them infeasible to be used with low-power smart devices.

Assessment of emerging technologies is needed not only to evaluate the overall behavior of home automation applications, but also to check if they meet the required low-power, cost-effective, reliable, and scalable constraints. Rathnayaka *et al.* [19] compared the features of emerging wireless solutions. They concluded that different wireless solutions offer comparative benefits and limitations in different perspectives. On the top of that, HEMSs are taking place and benefiting from home automation technologies [20]. They aim to monitor, control, and optimize the performance of the automated homes. HEMSs provide solutions for the connected homes to let them consume energy in an efficient way by following a DR schema.

E. The Virtual Power Plant

The Virtual Power Plant (VPP) is a cluster of dispersed generator units, controllable loads, and storage systems, aggregated to operate as a unique power plant [3], [21], [22]. VPPs can bring flexibility into the smart grid by controlling Distributed Energy Resources (DERs) and smart appliances. VPPs help balancing the fluctuating electricity feeds and the loads locally. It supports the integration of decentralized power generation which will bring more reliability in the grid. In practice, most VPPs merely act as virtual storages by providing load shifting. If the grid operator requests a certain amount of regulation power, it must be delivered by the VPP. They might not be interruptible just at a particular time or their virtual storage might be empty at other times [10]. Despite these difficulties, VPPs can provide a statistical guarantee for load shifting on a small scale, and hence contribute to maximizing the flexibility of a grid.

III. System Model

To understand the role of an ICT platform supporting DR, a system model has been developed. The model provides an abstraction of the key elements of the infrastructure. This section introduces the SEMIAH system model.

The system model divides into three distinct layers to provide means for generalizations of device, information objects, communications, etc. The SEMIAH system model is designed as a layered model illustrated in Fig. 3. The Internet of Things

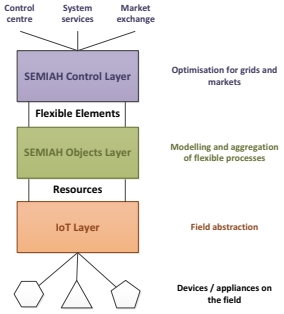


Fig. 3: SEMIAH layered system model.

(IoT) layer has to role of providing an abstraction for field devices deployed in households. The SEMIAH Objects layer manages customized objects such as information objects. The SEMIAH Control layer has the role to manage SEMIAH objects according to actors requirements. For instance, this could be the actuation of an appliance that is operated to provide flexibility for a household.

A. The Internet of Things Layer

The IoT layer addresses the problem of providing a uniform access to heterogeneous devices and appliances. The SEMIAH infrastructure has to collect measured values and to control output values on appliances through sensors and actuators.

Appliances belong to different categories such as heat pumps, washing machines, refrigerators, etc. and they can share some characteristics based on *shiftability* and/or *interruptibility* [12]. Different appliance models of the same category feature different ways to access the input and output parameters and often also a different set of parameters. The fundamental role of the IoT layer is to provide an abstract view of that diverse world to the SEMIAH Objects layer. Hence, data acquisition methods, local communication protocols, etc. are abstracted by the SEMIAH IoT layer.

B. The SEMIAH Objects layer

Appliances are addressed through a coherent semantics provided by the SEMIAH Objects layer. Ideally, semantics are defined at the appliance category level, i.e., there is a resource type per appliance category. The SEMIAH Objects layer disposes of a uniform method to access input and output parameters. This method must be independent of any appliance models, but also independent of appliance categories. In SEMIAH, the abstract representation of an appliance is called a Resource. A Resource represents a device, for example an appliance (appliance interface) prosuming electrical energy or a meter of any type. The most relevant object in the SEMIAH Objects layer is the "flexible element". Such an element knows at each instant the flexibility it can provide in the close future, and it enables to reserve and activate that flexibility.

TABLE I. List of possible collections.

Collection type	Description
Household or Building	Set of all processes in the household / building
Feeder	All collections / processes connected on a given feeder;
Market group	Set of all collections / processes belonging to owners having a contract with a given market actor (typically an electricity supplier);
Electrical cars collection	Set of all electrical cars.

In the UML class diagram, shown in Fig. 4, the essential parts of the system model are presented. The main class is FlexibleElement. It models a flexible element that can be linked to one or several IoT layers resources. Collection types are not defined in the system model since they can be dynamically defined to match specific environments. The FlexibleElement provides a bidirectional communication with Controllers representing the SEMIAH actors. A Controller can be informed about the current Flexibility (i.e., the forecast of the flexibility for the near future) provided by a FlexibleElement, and it can request the activation of available flexibility.

A FlexibleElement can be either a *process* or a *collection*. A *process* abstracts an atomic flexible process like for example: space heating. A Process is linked to one or more Resources. The Process state is essentially stored in its Flexibility object. The Forecaster upgrades the ProcessFlexibility each time a Resource is updated. When the Process is requested to activate flexibility, it asks its ProcessController to set (in real-time) appropriate Resource parameters and to check that the request is performed as expected. A Collection features also a Flexibility. When the Flexibility of one of its members (either a Process or a Collection) is significantly modified, the collection's aggregator updates the proper Flexibility of the Collection. Flexibility calculation for a Process or a Collection has to consider constraints (formalised using the Constraint interface). When a Collection is requested to activate Flexibility, its Scheduler dispatches the request among all member elements.

Individual processes are basic flexible elements. Collections, which are sets of processes and/or sub-collections, are also flexible elements. Hence, a collection must express its flexibility and provide some controllability. The collections flexibility is an aggregation of the flexibility of its members. When requested to activate its flexibility, a collection must dispatch the request between its members. Collections have two roles: *i*) they allow strongly reducing the number of elements presented to the SEMIAH Control layer, and *ii*) they allow enforcement of constraints for flexibility aggregation and dispatching of flexibility requests. Table I and Table II give lists of possible meaningful *collections* and *processes*, respectively.

Constraints can be specified at the process level and at the

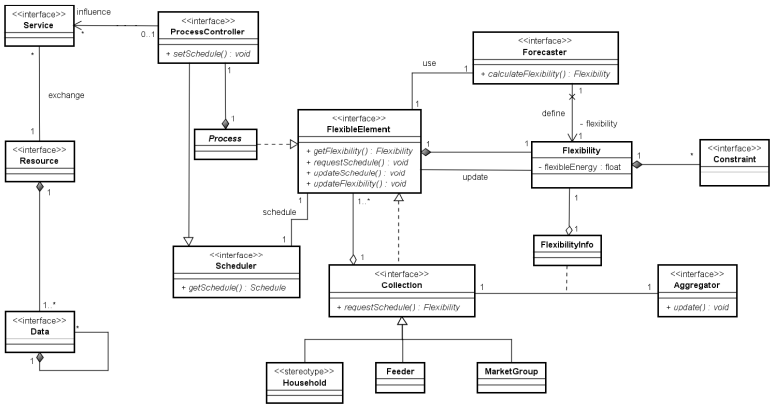


Fig. 4: UML class diagram for the system model.

TABLE II. List of possible processes.

Process type	Description
Space heating	Uses energy to provide electricity to heat conversion
Appliance work	Uses an appliance to make mechanical work
Lighting	Provides illumination in rooms, streets etc.
Charging	Loading an appliance with capacity to store electrical energy

collection level. An example of a constraint for a thermal process is the need for a temperature to remain in a given band to provide comfort to the occupants. Moreover, a *deadline flexibility* can be used to describe the required ending-time of a running appliance [12]. For a building collection, a constraint could be for example the limitation of the consumed power. The system model should put as few constraints as possible on semantics as well as on aggregation and scheduling algorithm. Thenceforth, new semantics or new algorithms can be supported without changes in the system model.

C. The SEMIAH Control Layer

SEMIAH allows system operators and electricity suppliers to activate flexibility to fulfil their specific objectives. The goal of the SEMIAH Control layer is to capture and serve user-specific requests. Typically these relate to grid control through access to control centers, ancillary services or bidding on the energy market exchange. The SEMIAH Control layer processes requests so that they can be forwarded to the SEMIAH Objects layer. The SEMIAH Control layer addresses SEMIAH Objects flexible elements. Hence, the SEMIAH Control layer can manage collections and/or individual processes.

FlexibleElements require that an external entity interact with them to concretely activate available flexibility in whole or in part. This is precisely the role of a Controller instance such as a VPP. A VPP is a class, typically instantiated by a larger software component that orchestrates FlexibleElement instances to address market objectives and/or grid control objectives.

IV. SEMIAH Integration Framework

This section presents an architectural approach that addresses the challenges of aggregators offering a DR service. Fig. 5 shows an abstract view of the SEMIAH architecture of the SEMIAH component-based integration framework. The heart of the architecture is the Generic Virtual Power Plant (GVPP). The GVPP has a number of consumer and provider interfaces to be listed below.

SEMIAH is a component-based framework that lets developers build, statically or dynamically, a scalable infrastructure based on a variety of IoT technologies. It has been the goal to develop a framework that integrates a set of well-defined concepts in DR and programming abstractions with the types of infrastructural support. To this end, the conceptual foundations of the SEMIAH framework has been developed; starting with system modeling as the primary interest. Furthermore, the frameworks rests on the state of the art in distributed computing such as service-oriented architectures for loosely-coupled systems.

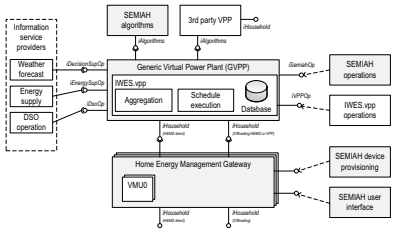


Fig. 5: SEMIAH technical architecture.

A. Functional Components

SEMIAH proposes a two-layer approach: a back-end service that resides in a scalable Infrastructure as a Service (IaaS) and a front-end that controls household appliances via a residential gateway. The two-layer approach applies to service-oriented platforms that encompasses a "virtualization" layer, including a Virtual Machine Units (VMUs) such as a Java Virtual Machine (JVM). Within these VMUs the components related with the application workflow execution and the DR instance of the presented framework are deployed.

The SEMIAH aggregator subsystem "wraps" the VPP component and becomes the integration layer between a centrally controlled VPP and the distributed set of residential households (a.k.a. "the horse shoe model"). This configuration allows or IoT protocols and middleware to be used in the communication between the VPP component and the aggregation layer.

B. Interfaces

Table III provides a list of open interfaces in the SEMIAH framework architecture. Interfaces are composed based on the principle of separation of concerns in software engineering. The framework uses Representational State Transfer (REST) for the communication between entities by using HTTPS over TCP/IP.

C. Data Models

The interworking between SEMIAH components are enabled by a common set of data models. These are defined based on a subset of the Common Information Model (CIM) [23] taylored to the DR concept of SEMIAH.

D. Deployments and Configurations

The system model does not address the distribution of objects on concrete computing devices. The underlying assumption is that any object can be implemented on any device and that some distributed systems technology will implement the interactions between objects. The following computing devices can be considered: *i*) home energy management gateway in each household/building; *ii*) feeder management device typically located in the secondary substation; and *iii*) full-featured server or (private or public) cloud service.

TABLE III. SEMIAH infrastructure interfaces.

Interface name	Description
iDecisionSupOp	Used to deliver forecast information e.g., RES production, CO_2 emission rate etc. to support decisions of the Generic VPP (GVPP)
iEnergySupOp	Used to deliver information related to energy markets such as time varying price information.
iDsoOp	Use by DSO to disseminate grid related constraints, provide direct control and for emergency control.
iAlgorithms	The GVPP has an open interface for 3rd party components to interface to VPP operations. This may e.g., be new algorithms for aggregation, forecasting, and scheduling. The interface is also a pivot point for integration of components developed and used in SEMIAH.
iSemiahOp	Used for operation and maintenance of the SEMIAH system.
iHousehold	Used for connecting and controlling the household appliances through the Home Energy Management Gateway (HEMG).
iHouseholdCollection	The interface is similar to iHousehold except that is operates of collections of e.g., households and/or appliances.
iVppOp	Used for operation and maintenance of the GVPP components planned for use in the SEMIAH pilot.

The distribution of objects on devices is influenced by several factors including: hardware costs and energy consumption of computing devices, telecommunication bandwidth, prosumers expectation and legal requirements for privacy, prosumers need for confidentiality protection, and the management of complexity (to supervise the system and for software updates). For SEMIAH, the following generic rules apply:

- The VPP component is an existing software module integrated into SEMIAH through a SEMIAH defined Application Programming Interface (API);
- All SEMIAH models and algorithms are independent of deployment topologies and of of distributed systems technologies.

The development and implementation of the proposed integration framework demands a concerted effort to apply and extend the existing technologies through near-future initiatives, while promoting forward-looking research and development to solve underlying critical issues in the long term to ensure economic, environmental, and societal impacts.

V. Low Voltage Grid Impact Assessment

A first conceivable risk, resulting from the possible impact of a DR on the power grid, is instability due to the feedback loop introduced between demand and DR triggers such as wholesale market prices. Theoretical work has shown that application of real-time pricing to consumers leads to increased volatility [24]. DR programs based on aggregation have been

found better in terms of stability than ones based on prices to end users [25]. A second possible risk, loss of diversity, arises when the volume of participation in a given DR program increases. Load diversity is an important dimensioning factor for power grids. It is based on the observation that consumption patterns vary across electricity users. As a result, the total peak demand on a grid portion is lower than the sum of individual peak demand levels. The ratio between this sum and the total demand is called *coincidence factor*. It generally decreases with an increasing number of users. For electric water heaters, it is typically around 1/6 in Europe [26]. This observation enables great savings in infrastructure costs by dimensioning cables and transformers for a much lower load than the sum of nominal loads of the households/buildings attached to the grid.

Ripple control as used for example to control electric water heaters in peak/off-peak tariff structures is widely implemented in countries like France, Italy and Switzerland. As this approach is based on broadcast signals, it leads to a strong reduction in diversity and to spikes in power demand. In a transformer station in Switzerland whose maximum load is 6 MW, overloading by more than 1 MW has been observed due to an almost instant rise in demand by about 6 MW in response to a ripple control signal. Probabilistic analysis has shown that the loss of diversity depends on the duration of the controlled interruption of demand: from a starting point of 0.32, the coincidence factor increases to 0.39 after a ten-minute interruption, 0.8 after a two-hour interruption, and 0.95 after a four-hour interruption. In the latter case, more than five hours are necessary to reach back the initial coincidence of 0.32. This rebound effect has two detrimental effects on the power grid. It leads to rapid voltage fluctuations potentially beyond acceptable values, and increases cable and transformer loading potentially beyond their maximum values. It is particularly strong in the case of thermostatically-controllable appliances such as Heating Ventillation and Air Conditionings (HVACs), heat pumps, or water heaters [27]. The rebound effect can be integrated in the load forecasting so that sufficient generation capacity is available to serve the additional load after a sudden decrease of DR incentive [28]. This solution, however, does not solve the local network issues. The workaround put in place in the case of ripple control is to spread signals over time to avoid simultaneously turning on all controlled devices. More generally, several control strategies of thermal electric loads have been investigated to mitigate the rebound effect. The strategy which leads to the lowest peak load after interruption is the one which maintains the user's comfort during the interruption as the SEMIAH concept intends to do. However, this approach also delivers the lowest reduction in power and energy demand [29]. Spreading complete switch off of appliances over time seems to give the best compromise between effectiveness of the response to a load-reduction signal and minimization of the rebound load.

SEMIAH will take into account the grid constraints in the scheduling of electricity loads. However, this procedure will be limited by the lack of detailed estimates of voltage variations due to simultaneous switching of controlled appliances and a pragmatic approach seems to be the only feasible way forward. Measurements available in SEMIAH are expected to be only energy consumption in participating households/buildings, at a maximum rate of 0.1 Hz. This is an approximate of average active power consumption. This information can only provide a crude estimate of the voltage, even if the topology of the grid is known, because the voltage in LV grids also depends on reactive power. The information can, however, be used to manage the increase in component loading due to the loss of diversity. For efficient operations, the power rating of the transformers and of each connected participant should be known. Otherwise, it can only be estimated based on design rules used in the location under consideration, but that bears large uncertainties. Managing diversity is relatively easy from an information point of view but it is a complex constraint on the optimization algorithm. To minimize risk, this means that geographic spread should be maximized when deploying DR schemes rather than rolling them out to district by district.

Table IV summarizes the conclusions of the LV grid stability analysis. The two locations, that have been analysized, are Skarpnes in Norway and Visp in Switzerland.

VI. Conclusion and Future Work

The European research project SEMIAH addresses the DR challenge of smart grids. It offers an open component-based framework for an aggregator service platform that can be deployed to pool flexibility of prosumption for a large number of residential households. The key system design challenges such as aggregation, forecasting, and load scheduling have been outlined. Furthermore, the SEMIAH system model and the architectural framework have been presented. The impact of the LV grid stability was analyzed. By combining the SEMIAH framework with the introduction of novel and innovative business models, prosumers are provided with incentives to offer flexibility of prosumption that will lead to a more secure and sustainable energy supply.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework program (FP7/2007-2013) under grant agreement n° 619560 (SEMIAH).

References

[1] F. Li, W. Qiao, H. Sun, H. Wan, J. Wang, Y. Xia, Z. Xu, and P. Zhang, "Smart Transmission Grid: Vision and Framework," *IEEE Transactions on Smart Grid*, vol. 1, no. 2, pp. 168–177, 2010.

[2] M. Ruska and L. Similä, *Electricity Markets in Europe: Business Environment for Smart Grids*, ser. VTT Tiedotteita - Valtion Teknillinen Tutkimuskeskus, 2011, no. 2590.

[3] S. Rotger-Griful and R. Jacobsen, "Control of Smart Grid Residential Buildings with Demand Response," vol. 581, pp. 133–161, 2015.

[4] S. Karnouskos, "Demand Side Management via Prosumer Interactions in a Smart City Energy Marketplace," in *Innovative Smart Grid Technologies (ISGT Europe), 2011 2nd IEEE PES International Conference and Exhibition on*, 2011, pp. 1–7.

TABLE IV. Analysis of the grid impact.

Issue	Expected	Present	Rood cause	SEMIAH potential impact
Overvoltage	Yes	No	Net power injection from building Cable sizing	Positive: alignment between generation and consumption in building
Harmonic distortion	Yes	Only in Skarpnes, Norway	Power converters	None
Reverse power flows	Yes	Only in Visp, Switzerland, to the LV feeders; expected to MV side in summer time	Instant generation in excess of instant consumption in feeder	Positive: alignment between generation and consumption in feeder
Power factor fluctuations	At night	At times of Photovoltaics (PV) generation in Visp, Switzerland	Power converters Cable sizing	None

[5] R. H. Jacobsen and E. Ebeid, "SEMIAH: Scalable Energy Management Infrastructure for Aggregation of Households," in *40th EUROMICRO conference on Software Engineering and Advanced Applications and 17th EUROMICRO conference on Digital System Design*, work in Progress Session. Friday 29 August 2015. Verona, Italy.

[6] F. E. R. Commission, "2011 Assessment of Demand Response and Advanced Metering: Staff Report," Tech. Rep., 7 Novemeber 2011 2011. [Online]. Available: http://www.ferc.gov/legal/staff-reports/11-07-11-demand-response.pdf

[7] Y. W. Law, T. Alpcan, V. C. S. Lee, A. Lo, S. Marusic, and M. Palaniswami, "Demand Response Architectures and Load Management Algorithms for Energy-Efficient Power Grids: A Survey," in *2012 Seventh International Conference on Knowledge, Information and Creativity Support Systems (KICSS)*, , 2012, pp. 134–141.

[8] J. LaMarche, K. Cheney, S. Christian, and K. Roth, "Home Energy Management Products & Trends," *Fraunhofer Center for Sustainable Energy Systems.Cambridge, Massachusetts*, 2011.

[9] P. Siano, "Demand Response and Smart Grids – A survey," *Renewable and Sustainable Energy Reviews*, vol. 30, no. 0, p. 461, 2014.

[10] P. Palensky and D. Dietrich, "Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 3, pp. 381–388, 2011.

[11] C. Goldman, "Coordination of Energy Efficiency and Demand Response," *Lawrence Berkeley National Laboratory*, 2010.

[12] A. G. Azar, R. H. Jacobsen, and Q. Zhang, "Aggregated Load Scheduling for Residential Multi-Class Appliances: Peak Demand Reduction," in *Proceedings of the* 12^{th} *International IEEE Conference on the European Energy Market (EEM)*, 2015, pp. 1–6, 2015.

[13] P. Bartholomew, W. Callender, C. Hindes, C. Grimm, K. Johnson, M. Straub, D. Williams, and M. Williamson, "Demand Response Measurement & Verification," Assoc. Edison Illum. Co. Load Res. Comm., Tech. Rep., March 2009.

[14] C. Xia, Y. Zhu, and B. Lin, "Renewable Energy Utilization Evaluation Method in Green Buildings," *Renewable Energy*, vol. 33, no. 5, p. 883, 2008.

[15] E. A. Feinberg and D. Genethliou, *Load Forecasting*, ser. Applied Mathematics for Restructured Electric Power Systems. Springer, 2005, pp. 269–285.

[16] T. Mancini, F. Mari, I. Melatti, I. Salvo, E. Tronci, J. K. Gruber, B. Hayes, M. Prodanovic, and L. Elmegaard, "Demand-Aware Price Policy Synthesis and Verification Services for Smart Grids," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2014, pp. 794–799.

[17] A. Soares, A. 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.

[18] M. B. Propp and D. L. Propp, "Power Line Communication Apparatus," mar 21 1989, uS Patent 4,815,106.

[19] A. J. D. Rathnayaka, V. M. Potdar, and S. J. Kuruppu, "Evaluation of Wireless Home Automation Technologies," in *2011 Proceedings of the 5th IEEE International Conference on Digital Ecosystems and Technologies Conference (DEST)*, 2011, pp. 76–81.

[20] F. Bouhafs, M. Mackay, and M. Merabti, *Home Energy Management Systems*, ser. Communication Challenges and Solutions in the Smart Grid. Springer New York, 2014, pp. 53–67.

[21] H. Saboori, M. Mohammadi, and R. Taghe, "Virtual Power Plant (VPP), Definition, Concept, Components and Types," in *2011 Asia-PacificPower and Energy Engineering Conference (APPEEC)*, , 2011, pp. 1–4.

[22] P. Lombardi, M. Powalko, and K. Rudion, "Optimal Operation of a Virtual Power Plant," in *IEEEPower Energy Society General Meeting, 2009. PES '09.*, July 2009, pp. 1–6.

[23] "Energy management system application program interface (ems-api) - part 301: Common information model (cim) base," International Electrotechnical Commission (IEC), Standard IEC61970-301:2013, 2013.

[24] M. Roozbehani, M. A. Dahleh, and S. K. Mitter, "Volatility of Power Grids Under Real-Time Pricing," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 1926–1940, 2012.

[25] J. L. Mathieu, T. Haring, J. O. Ledyard, and G. Andersson, "Residential Demand Response Program Design: Engineering and Economic Perspectives," in *2013 10th International Conference on the European Energy Market (EEM)*, 2013, pp. 1–8.

[26] M. Orphelin and J. Adnot, "Improvement of Methods for Reconstructing Water Heating Aggregated Load Curves and Evaluating Demand-Side Control Benefits," *Power Systems, IEEE Transactions on*, vol. 14, no. 4, pp. 1549–1555, 1999.

[27] W. Zhang, K. Kalsi, J. Fuller, M. Elizondo, and D. Chassin, "Aggregate Model for Heterogeneous Thermostatically Controlled Loads with Demand Response," in *2012 IEEE Power and Energy Society General Meeting*, 2012, pp. 1–8.

[28] J. W. Black and R. Tyagi, "Potential Problems with Large Scale Differential Pricing Programs," in *2010 IEEE PES Transmission and Distribution Conference and Exposition*, 2010, pp. 1–5.

[29] D. D. Silva, "Analyse de la flexibilité des usages électriques résidentiels: application aux usages thermiques," 2011, PhD dissertation, École Nationale Supérieure des Mines de Paris.

Paper D

Appliance Scheduling Optimization for Demand Response

The paper presented in this chapter has been accepted as a journal publication [5].

[5] Armin Ghasem Azar and Rune Hylsberg Jacobsen, "Appliance Scheduling Optimization for Demand Response," *International Journal on Advances in Intelligent Systems*, vol. 2, no 1&2, 2016, pages 50-64, Link to paper

Authors contribution: Armin Ghasem Azar conceived the main design of the scheduling system; Armin Ghasem Azar was the main editor of the manuscript; Rune Hylsberg Jacobsen provided support on the validation of simulations; Rune Hylsberg Jacobsen contributed with the text in selected areas and to the reviewing process.

Appliance Scheduling Optimization for Demand Response

Armin Ghasem Azar and Rune Hylsberg Jacobsen
Department of Engineering, Aarhus University, Denmark
Email: {aga, rhj}@eng.au.dk

***Abstract*—The paper studies the challenge of the electricity consumption management in smart grids. It focuses on different impacts of demand response running in the smart grid engaging consumers to participate. The main responsibility of the demand response system is scheduling the operation of appliances of consumers in order to achieve a network-wide optimized performance. Each participating electricity consumer, who owns a set of home appliances, provides the desired expectation of his/her power consumption scenario to the demand response system. It is accompanied with time limits on the flexibility of controllable appliances for shifting their operational time from peak to off-peak periods. The appliance scheduling optimization for demand response is modeled as an optimization problem. It concentrates on reducing the total electricity bills and CO_2 emissions as well as flattening the aggregated peak demand at the same time. This paper categorizes the appliances based on shiftability and interruptibility characteristics. It uses information of dwellings to determine an effective appliance scheduling strategy. This strategy gets influenced by grid constraints imposed by distribution system operators. The simulations confirm that scheduling appliances of 100 consumers yields a significant achievement in the peak demand reduction while averagely satisfying the comfort level of consumers.**

***Keywords*–Smart grid, demand response, appliance scheduling, knapsack problem, dynamic programming, multi-objective optimization.**

NOMENCLATURE

Constants

PDT	Peak Demand Threshold
PPD	Peak Power Demand
A_i	Number of appliances in D_i
$a_{i,j}$	Appliance j in dwelling i
D_i	Dwelling i
G	Number of generations
N	Number of dwellings
p_c	Crossover propability
p_m	Mutation propability
$p_{i,j}$	Priority of appliance $a_{i,j}$
Q	Population size
T	Number of time intervals
$DF_{i,j}$	Deadline flexibility of appliance $a_{i,j}$
$TPD_{i,j}$	Total power demand of appliance $a_{i,j}$

Indices

i	Index of dwellings
j	Index of appliances
t	Index of time intervals

Variables

$x^t_{i,j}$	Decision variable of selecting $PD^t_{i,j}$
CO_2E^t	Amount of CO_2 emission at time interval t
EP^t	Electricity price at time interval t
$PD^t_{i,j}$	Power demand request of appliance $a_{i,j}$ at time interval t
$RP^t_{i,j}$	Number of remaining power requests of appliance $a_{i,j}$ at time interval t

I. INTRODUCTION

The smart grid has emerged as a novel infrastructure aiming to transform the existing power system into a reliable and consumer-centric one. It forms a distributed energy delivery network using the electricity and information streams simultaneously. This network possesses a self-healing characteristic toward facing unforeseen electricity outage circumstances. Its reliability and stability are based on intelligent controllers, in which they try to establish bilateral communication channels between consumers and Distribution System Operators (DSOs). The demand side management service provides an opportunity to energy actors for an active participation in counterbalancing the *demand response*. It helps to find the most reliable and effective energy solutions in real-time. This paper extends the work presented in [1]. Here, the key contributions include the extended mathematical formulation and description of the demand response system along with a presentation of an extensive simulation performance analysis.

Demand response is one of the most challenging issues in demand side management, which is responsible for providing effective and comprehensive energy solutions [2]. From the consumers' point of view, demand response attempts to motivate them to modify their electricity usage patterns, in response to potential grid incentives. In contrast to this point of view, DSOs intend to equilibrate demands with responses to reduce peak power demands as much as possible [3]. These purposes can be achieved through both *curtailing the power demand* and *controlling the activation time of electricity usages*. However, a mutual challenge behind these procedures is how to motivate consumers to modify their power demand profiles [4][5].

One of the most pragmatic incentives for consumers to modify their consumption behavior is electricity prices. Although demand response includes efforts to change the electricity usage of consumers with respect to the alterations in the electricity prices, however, reducing the peak demand and CO_2 emission also help to decrease the greenhouse gas emissions [6]. This reduction results in a co-optimization approach of power demand cost and CO_2 emission. In some peak hours, the demand response system has to shift some *power demand requests* from diverse dwellings to another time interval. This shifting can occur several consecutive/separate times over a day. Obviously, this leads to some changes in the daily power consumption of consumers. This causes a problem named *dissatisfaction of consumers*. As a result, maximizing the satisfaction of consumers is an essential objective as well.

Consumers are also interested to reduce their electricity cost while contributing to CO_2 emission reduction program. From the DSOs' point of view, they aim to shave the peak period, which results in flattening the aggregated power demands over time.

Figure 1 shows a conceptual view of various communications in the grid. Each dwelling has a specific scenario for its own appliances. This scenario includes the desired timetable of using appliances in a day. First, appliances are classified based on the *shiftability* feature [7]. Second, shiftable appliances are categorized by the *interruptibility* feature. These classifications permit consumers to give a priority to appliances, which is important for their starting time. Once the consumer chooses to operate an appliance in demand response ready mode, the consumer offers flexibility to the grid and provides an opportunity to the demand response system for reducing the peak demand.

This paper proposes a local power scheduling algorithm attempting to schedule power demand requests of appliances. Here, local means receiving the power demand requests with a specific time resolution and scheduling them accordingly. As its principal novelty, the algorithm runs concurrently and need not know the whole operating period of appliances. The scheduler intends to schedule power demand requests optimally once they arrive. At each time interval, its main responsibility is to allow some appliances to operate and shift the operating cycle of the remaining appliances to the future. This shifting is enabled by utilizing Peak Demand Thresholds (PDTs) imposed by DSOs. The scheduling algorithm attempts to keep the aggregated power consumptions below PDTs continuously.

This rest is organized as follows: Section II overviews the related work. Section III presents the system model. Section IV proposes the power scheduling algorithm. Section V discusses the simulation setup and analysis. Finally, Section VI concludes the paper and provides the possible future extensions.

II. Related Work

A considerable amount of literature is published on smart grids due to concerns on the inefficient structure of the current electrical grid in responding to the growing demand for electricity [8][9]. Farhangi [8] investigated the differential impacts of transforming the current electrical grid to a complex system of systems, named the smart grid while Fang *et al.* [9] surveyed the enabling technologies for data communications in the smart grid. With the advent of smart grids, new solutions are becoming available. To support these, demand response programs endeavor to change the electricity usage patterns of consumers in response to electricity prices or other signals. These programs are considered as reliable solutions to improve the energy efficiency and reduce the peak demand [10]. To reach these goals necessitates demand response service providers investing on proposing functional and potential power scheduling services to the smart grid.

Most of the current research on the power scheduling problem focuses on scheduling power demand requests of appliances of consumers wrapping as a single-objective framework while relying on historical data and forecasting services [11][12]. Agnetis *et al.* [11] defined the problem of optimally scheduling a limited number of manageable appliances of only one dwelling solving with a high computational

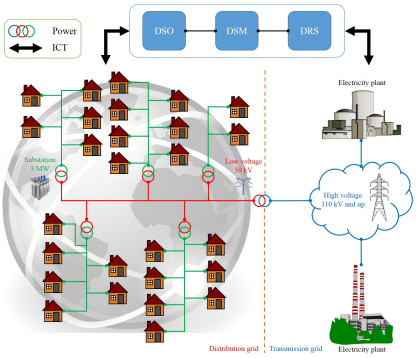


Figure 1. Conceptual view of various communications in the grid

algorithm based on the mixed integer linear programming. O'Brien [12] proposed a greedy algorithm for automatically scheduling the shiftable appliances with completely predetermined power profiles while missing to take any grid stability constraint into account.

Nevertheless, far too little attention has been paid by smart grid researchers to design a system model where power scheduling is done near real-time. Jacobsen *et al.* [1] found this gap and developed a simple but efficient smart appliance power scheduling mechanism based on the peak demand reduction strategy. Consecutively, Azar *et al.* [13] followed a design methodology that efficiently utilized a time-independent PDT policy for decreasing the aggregated peak demand considering the appliance reception minimization method. It successfully flattened the aggregated power consumption based on a centralized demand response system.

This paper advances the state of the art in formulating a demand response service where appliances send their power demand requests with a specific time resolution accompanying the consumer's time-limit flexibilities. The DSO schedules the incoming power demand requests according to the customers' and its objectives. It attempts to keep the aggregated power demands below PDTs over time.

III. System Model

This section clarifies the proposed system model, as Figure 2 illustrates its conceptual view. Consumers play a major role in this system model since they provide their desired electricity consumption scenarios and corresponding flexibilities to the demand response system. In addition, DSOs impose some grid stability constraints to maintain the electrical grid, such as PDT. Electricity prices of a typical day with the corresponding CO_2 emission data are another system input. The demand response system will receive these input data and then, executes the scheduling algorithm attempting to schedule appliances of dwellings with respect to the objectives and constraints settled in the demand response system.

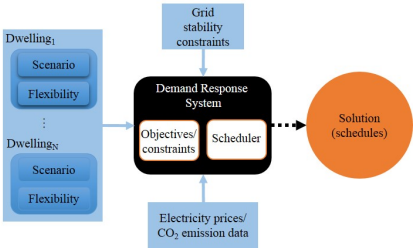


Figure 2. System model of the appliance power scheduling

A. Consumers: Appliance Point of View

This paper assumes there are $N \in \mathbb{N}$ dwellings connected to a feeder in the electrical grid. Each dwelling D_i, where $i \in \{1, 2, \ldots, N\}$, possesses $A_i \in \mathbb{N}$ appliances. Each appliance $a_{i,j}$, where $j \in \{1, 2, \ldots, A_i\}$, is a driver of residential power demands. To guarantee the full operation of appliances, the demand response system should check whether appliances have completed their responsibilities during the day or not. Therefore, Equation (1) shows this hard constraint.

$$\sum_{t=1}^{T} \left(PD_{i,j}^{t} \times x_{i,j}^{t} \right) = TPD_{i,j}, \tag{1}$$

where $PD_{i,j}^{t} \in \mathbb{R}^{*}$ (watts) is the power demand of appliance $a_{i,j}$ at time interval t. Notation $x_{i,j}^{t} \in \{0, 1\}$ is the decision variable of the optimization problem. $x_{i,j}^{t} = 1$ allows appliance $a_{i,j}$ to operate at time interval t while $x_{i,j}^{t} = 0$ shifts its operation to the future. Furthermore, $TPD_{i,j} \in \mathbb{R}^{+}$ (watts) is the total power demands of the appliance.

Appliances are classified according to some smart features named shiftability and interruptibility [7][13]. Shiftability means giving permission to the demand response system to shift the power demand requests of shiftable appliances to later time intervals. However, some appliances cannot be shifted, for instance the refrigerator. These appliances are members of non-shiftable appliances. Afterwards, shiftable appliances are divided into two groups based on the interruptibility feature. The electric vehicle is a typical example of an appliance exhibiting this feature. The demand response system can both shift and interrupt the duty cycle of charging the electric vehicle. Nevertheless, those appliances, which can be shifted, but are infeasible to be interrupted are called uninterruptible appliances (e.g., dishwasher). Their whole operating duty cycle can be shifted to another time interval. However, they should not be interrupted because of the continuity in their cycle. Equation (2) formulates this hard constraint, which is valid at each time interval:

$$\begin{aligned} &\text{Non-shiftable appliances} \rightarrow x_{i,j}^{t} = 1, \\ &\text{Uninterruptible appliances} \rightarrow \begin{cases} x_{i,j}^{t} = 1 & \text{if } x_{i,j}^{t-1} = 1, \\ x_{i,j}^{t} \in \{0, 1\} & \text{otherwise}, \end{cases} \\ &\text{Interruptible appliances} \rightarrow x_{i,j}^{t} \in \{0, 1\}. \end{aligned} \tag{2}$$

At each time interval t, the demand response system is signaled with power demand requests of appliances. Once it receives a power demand request from a non-shiftable appliance, it is allowed to operate. If the request belongs to an uninterruptible appliance first it should check whether the relevant appliance has been allowed to start its work at the previous time interval. If so, the system cannot interrupt and shift it to another time interval. Otherwise, it is possible to shift it, if needed. Finally, if an interruptible appliance sends a power demand request at any interval, it is possible to either allow or shift it.

In real world, consumers sometimes give priorities to use their appliances based on their preferences. For instance, the stove has higher priority compared to the laundry machine. There are two kinds of priority preference named *static* and *dynamic*. The former denotes time-independent priorities of appliances, where the pairwise comparison between each two appliances is constant with respect to some criteria such as emergent usage, welfare, or electricity cost. Each consumer can set $0 < p_{i,j} \leq 1$ as the priority of using appliance $a_{i,j}$ over the day. As a result, if the demand response system confronts a circumstance, when it should decide to select one appliance among two or more, then, the appliance, which has the highest priority will be selected [14]. Finally, as a brief description of the dynamic priority, sometimes consumers change the priorities of their appliances as time moves on. For instance, one consumer gives a priority to his/her dishwasher in the morning. In the afternoon, he/she changes its priority since the washing machine is needed to operate at the same time. Therefore, dishwasher's priority is decreased. Nevertheless, for simplifying the model, the dynamic priority constraint is not considered in this paper.

Consumers participating in demand response programs provide some flexibilities to the demand response system for operating their appliances. Let us assume one consumer is interested to plug in his/her Nissan Altra electric vehicle at 18:00. The charging cycle will typically take five hours [15]. Nonetheless, he/she is flexible to receive the electric vehicle in the finished state at most at 08:00 the next day. Therefore, the flexibility that the consumer offers to operate his/her electric vehicle is 14 hours. We name this concept as a *deadline* flexibility, which is a time-oriented constraint. This kind of flexibility helps the demand response system to shift some appliances, which relatively consume more than others, to the future. The demand response system should consider the remaining power demand flexibility (with given time limits) before shifting them. Equation (3) describes this constraint:

$$RP_{i,j}^{t} \leq (DF_{i,j} - t), \tag{3}$$

where $RP_{i,j}^{t} \in \mathbb{Z}^{*}$ relates to the number of remaining power demand requests of appliance $a_{i,j}$ from time interval t until the end of its duty cycle. Moreover, $DF_{i,j}$ (e.g., UTC) denotes the deadline flexibility of this appliance. The demand response system satisfies this constraint while it receives the power demand requests continuously. If the remaining power demand of an appliance is still less than its provided time limit flexibility, the demand response system can decide to allow it to start/continue in this time interval or to shift it to another time interval. To shift a power demand request, it is essential to ensure the satisfaction of all constraints.

Considering the aforementioned descriptions, each dwelling D_i has a specific scenario showing how the consumer intends to operate the appliances. Table I lists a sample scenario of operating the appliances in a typical dwelling. As described previously, deadline flexibility in using appliances means a firm deadline for finishing the related activity. For example, the consumer provides two hours of flexibility to the demand response system for charging the electric vehicle. More in details, it receives the first power demand request for charging the electric vehicle at the defined time. The demand response system has an opportunity to deliver the charged electric vehicle later in time by utilizing the provided deadline flexibility. It is possible to both shift and interrupt the charging process during the defined time period since the electric vehicle is a member of the interruptible appliances. Here, the priorities are time-independent (static). It is worthwhile emphasizing that the priority is applied to only shiftable appliances. Hence, the refrigerator and lighting will not undergo any scheduling procedure. They will receive an infinite priority since they are members of non-shiftable appliances.

B. Distribution System Operator: Grid Constraint Point of View

Currently, electricity producers generate more electricity since they are experiencing an insufficiency of electricity generation capacity because of the power demands by consumers. However, it can be avoided using demand shaping schemes. DSOs currently apply a threshold policy, in order to shave the peak, which results in shaping the demand profiles over time [1]. From an electricity grid point of view, the upper limit of the PDT may be enforced by the DSO by the installation of fuses and other safety-related measures such as protective relays. These devices may be dimensioned differently and the subscription fee for a dwelling often depends on the installed capacity. As a complement, adaptive schemes can be deployed as a control loop between a DSO-controlled generator side and individual dwellings [16]. Let

$$\sum_{i=1}^{N}\sum_{j=1}^{A_i}\left(PD_{i,j}^{t} \times x_{i,j}^{t}\right) \leq PDT, \tag{4}$$

where $PDT \in \mathbb{R}^{+}$ (watts) is a constant and time-independent power demand threshold, in which the demand response system attempts to keep the amount of allowed power demand requests below it. Nevertheless, Equation (4) sometimes cannot be satisfied owing to the provided deadline flexibilities and uninterruptibility feature of some appliances. Therefore, the demand response system will consider this constraint for power demand requests, in which the corresponding appliances: 1) still have time to start operating or 2) have not started yet. For the former the demand response system can still use the provided flexibility while for the latter it can shift the starting time of the appliance to the later time intervals. It is worth noting that priorities of appliances could be also considered in Equation (4).

C. Demand Response System: Objective Point of View

While the demand response system receives power demand requests of appliances, it cannot globally optimize the objectives since they are received at specific time intervals continuously. As a result, all objectives are based on a local controlling strategy, as follows.

TABLE I. A SIMPLIFIED EXAMPLE OF A DWELLING' SCENARIO

Start	End	Activity description	Deadline flexibility	Priority
00:00	24:00	Using the refrigerator	24:00	Infinite
08:00	24:00	Turning the lights on	00:30	Infinite
08:05	09:50	Putting the dishes into the dishwasher	10:30	0.2158
13:00	14:15	Putting the laundry into the washing machine	17:00	0.1063
17:25	18:15	Putting the washed laundry into the laundry dryer	22:00	0.1499
11:30	22:40	Using the computer	23:30	0.2649
19:50	22:00	Watching the TV	24:00	0.1293
20:00	22:00	Charging the electric vehicle	24:00	0.1338

1) Minimizing the Electricity Cost: Equation (5) formulates the willingness of the demand response system to minimize the electricity cost of consumers at each time interval. Here, $EP^t \in \mathbb{Z}^*$ (DKK per watts per hour) is the electricity price at each time interval.

$$f(x) = \min \sum_{i=1}^{N}\sum_{j=1}^{A_i}\left(PD_{i,j}^{t} \times x_{i,j}^{t} \times EP^{t}\right). \tag{5}$$

2) Minimizing the CO_2 Emission: Equation (6) shows the interest for reducing the CO_2 emission of dwellings at each time interval by applying the decision variable $x_{i,j}^{t}$ for all power demand requests. Here, $CO_2E^t \in \mathbb{R}^*$ (grams per watts per hour) is the amount of CO_2 emission at each time interval.

$$g(x) = \min \sum_{i=1}^{N}\sum_{j=1}^{A_i}\left(PD_{i,j}^{t} \times x_{i,j}^{t} \times CO_2E^{t}\right). \tag{6}$$

3) Maximizing the Comfort Level of Consumers: Equation (7) formulates how the demand response system is interested to maximize the comfort level of consumers over time. Comfort level indicates the consumers' desire to have their activities being done as they exactly expect from their scenarios. In fact, appliances aim to get permission to run their operations at each time interval as much as possible.

$$h(x) = \max \sum_{i=1}^{N}\sum_{j=1}^{A_i}\left(x_{i,j}^{t} \times p_{i,j}\right). \tag{7}$$

In conclusion, the demand response system considers the appliance power scheduling optimization as a mixed-integer linear programming problem including Equations (5) to (7) as its objective functions subject to Equations (1) to (4) as the relevant constraints. Next section will describe how the proposed scheduling algorithm attempts to solve this optimization problem applying diverse approaches.

IV. SCHEDULING ALGORITHM

Algorithm 1 presents the pseudo-code of the power scheduling algorithm. Considering the system model shown in Figure 2, the demand response system executes the scheduling algorithm to produce a specific schedule for appliances of dwellings based on the objectives and constraints, described in Section III. It receives power demand requests at specific time intervals. Apart from the PDT, the scheduler allows the non-shiftable power demand requests to start or to continue their

Algorithm 1: Power scheduling

```
Input  : The scenarios, power profiles, classification of appliances, PDT.
Output: Schedule of appliances of all dwellings.
Preprocessing the input data;
while receiving the power demand requests over time do
    Allow the non-shiftable appliances to start or to continue;
    Update PDT;
    if there are uninterruptible appliances, which have started previously then
        Allow them to continue;
        Update PDT;
    end
    if there are appliances, which cannot be shifted due to their deadline flexibility constraint then
        Allow them to start or to continue;
        Update PDT;
    end
    if there are some remaining power demand request then
        if their total consumption is less than the remaining PDT then
            Allow them to start or to continue;
            Update PDT;
        else
            Refer to the single/multi-objective Knapsack procedure;
            Allow the output power demand requests of the Knapsack procedure to start or to continue;
            Shift the remaining power demand requests to the next time interval;
        end
    end
end
```

duties. Furthermore, if there is an uninterruptible appliance, which has started at the previous time interval, it should be allowed to continue. Finally, if there is a power demand request, where shifting it to the next time interval violates its provided deadline flexibility, then, the same action of allowing it to start takes place. After finishing these procedures, the algorithm will check whether the total power demand of the remaining requests is below the remaining PDT (capacity) or not. If so, all will be permitted to start or to continue their procedure. Otherwise, the algorithm refers to the Knapsack procedure to select some requests from the remaining power demand requests to enable them to start or to continue, and shift the unselected requests to the next time interval.

Two challenging circumstances can occur during the scheduling, and handling them confirms the robustness of the scheduling algorithm. If there is a sudden drop in the electric power, indeed no appliance can send any power demand request. Therefore, the scheduling algorithm waits until the appliance sends its new power demand request. Furthermore, if all appliances in all dwellings are configured as non-shiftable with high priorities, the scheduling algorithm will allow all of them to operate, when they send their power demand requests. This is based on respecting the consumers who do not provide any flexibility to non-shiftable appliances. However, this is considered to be an infeasible and greedy setup.

A. The Knapsack Problem

The Knapsack problem is one of the traditional problems of computer science in combinatorial optimization literature [17]. Given F items, the Knapsack tries to pack the items to obtain the maximum total value. Each item gets a weight and value. The maximum weight that the Knapsack can tolerate is limited by a fixed capacity W. This problem has two versions: "0-1" and "fractional". In the former, items are indivisible meaning it is possible to either take an item or not. In contrast, in the fractional version, items are divisible and, therefore, the Knapsack can take any fraction of an item.

This paper gets the benefit from the first version since the remaining power demand requests are similar to the indivisible items in "0-1" Knapsack problem. The "0-1" Knapsack problem is NP-Complete since the time complexity of solving it in a brute-force approach is $O(2^F)$. Time complexity measures the time that an algorithm takes as a function of the size of its input. Applying brute-force approach means calculating the fitness of 2^M solutions to locate the optimal one. The power scheduling problem is reducible to this version since the demand response system should decide to allow those indivisible power demand requests, which optimize the objective(s) and satisfy the constraints simultaneously. Therefore, the discussing problem is also NP-Complete. Hereinafter, we the scheduler needs to refer to the Knapsack problem, we name it the Knapsack procedure.

Indeed, the Knapsack procedure requires not only to decide, which power demand requests have to be processed now and delay the others afterwards, but should also consider the starting (ending) times of the latter. The latter is reflected in the flexibility that consumers provide.

Table II defines the equivalent parameters of the Knapsack and power scheduling optimization problems according to various objectives. As described previously, the Knapsack procedure receives the remaining power demand requests, which their total power demand is indeed more than the remaining capacity. It calculates the fitness of produced feasible solutions, in which each solution includes some power demand requests.

TABLE II. EQUIVALENT PARAMETERS OF THE KNAPSACK PROCEDURE AND POWER SCHEDULING OPTIMIZATION PROBLEM

	Values (items)	Objective(s)	Weights	Capacity
Single-objective	Electricity cost of power demand requests	Minimizing the total electricity costs	Power demand requests	PDT
	CO_2 emission of power demand requests	Minimizing the total CO_2 emissions	Power demand requests	PDT
	Priority of power demand requests	Maximizing the total allowed power demand requests	Power demand requests	PDT
Multi-objective	Electricity cost and priority of power demand requests	Minimizing the total electricity costs and maximizing the total number of allowed power demand requests	Power demand requests	PDT
	CO_2 emission and priority of power demand requests	Minimizing the total CO_2 emission and maximizing the total number of allowed power demand requests	Power demand requests	PDT

As a result, the solution to this problem is a subset of received power demand requests, which should be allowed to start or to continue in this time interval. Then, there will most likely be some remaining power demand requests, which cannot successfully start or continue. These power demand requests should be shifted to the future.

Depending on the number of objectives chosen by the demand response system, different approaches can be used to run the Knapsack procedure. On the one hand, if the demand response system decides to run the scheduling with one objective, the scheduling problem turns into a single-objective optimization problem. This is equal to run the single-objective "0-1" Knapsack procedure with dynamic programming at each time interval (if needed) [14]. On the contrary, if at least two objectives are chosen, the scheduling algorithm corresponds to a multi-objective optimization problem, which has to be solved with relevant techniques [18]. It is worth noting that these approaches are used at each time interval, if needed. The following describes them.

1) Dynamic Programming: We utilize a dynamic programming approach to solve single-objective power scheduling problem. As Figure 3 demonstrates its principles, this approach first characterizes the structure of an optimal solution. Then, it decomposes the problem into smaller problems. Meanwhile, it finds a relationship between the structure of the optimal solution of the original problem and solutions of the smaller problems. It recursively expresses the solution of the original problem in terms of optimal solutions to smaller problems, which supports the optimality.

To this end, it follows a bottom-up computation approach. The value of an optimal solution is computed in a bottom-up manner using a table structure. This table is repeatedly filled to use in each iteration [19]. The structure of an optimal solution to the power scheduling problem is a subset of the remaining power demand requests, which optimizes the relevant objective. Algorithm 2 declares the dynamic programming method for running the single-objective Knapsack procedure. The time complexity of approaching the Knapsack procedure using dynamic programming is *O(M×PDT)*.

2) Multi-Objective Optimization: Multi-Objective Optimization (MOO) is an area of multiple criteria decision-making, where mathematical optimization problems involving more than one objective function should be optimized simultaneously [20]. Optimal decisions are taken in the presence of trade-offs between two or more conflicting objectives. Solving a MOO problem necessitates computing all or a representative set of Pareto-optimal solutions. In this paper, a Pareto solution comprises a subset of remaining power demand requests. When decision-making is emphasized, the objective of solving a MOO problem is to support a decision-maker in finding the most preferred Pareto-optimal solution. Here, the decision-maker is the demand response system, which should decide to allow only a subset of the remaining power demand requests to optimize the objectives and satisfy the constraints at each time interval accordingly. The objective functions are in

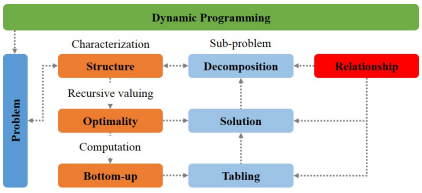


Figure 3. Principles of the dynamic programming approach

Algorithm 2: Approaching the Knapsack procedure: Dynamic programming

Input : power demand requests, PDT.
Output: The optimal solution at the current time interval.

Set F as the number of input power demand requests;
Create a $(F+1) \times (PDT+1)$ table named V;
if *the objective is minimization* **then**
  Set $V[0,0:PDT+1]$=Inf;
else
  Set $V[0,0:PDT+1]$=0;
end
for $i = 1$ *to* F **do**
  for $j = 1$ *to* PDT **do**
    if $PD[i] \leq j$ **then**
      if *the objective is minimization* **then**
        $V[i,j] = \min(V[i-1,j], PD[i] + V[i-1, j-PD[i]])$;
      else
        $V[i,j] = \max(V[i-1,j], PD[i] + V[i-1, j-PD[i]])$;
      end
    else
      $V[i,j] = V[i-1,j]$;
    end
  end
end
Return the $V[F, PDT]$ as the final solution;

Algorithm 3: Approaching the Knapsack procedure; Multi-objective evolutionary algorithm

Input : Remaining power demand requests, PDT, population size (Q), number of generations (G), crossover (p_c) and mutation (p_m) probabilities.
Output: A near-optimal solution at the current time interval.
Randomly produce initial solutions and combine them as the parent population;
Evaluate the parent population based on the objective functions;
Calculate the Pareto-fronts and the crowding distance of solutions inside the parent population;
$c = 1$;
while $c \leq G$ **do**
  Apply the selection operator on the parent population and forward to the crossover operator;
  Apply the crossover operator on the received solutions with a probability of p_c and forward to the mutation operator;
  Apply the mutation operator on the received solutions with a probability of p_m and put them into the offspring population;
  Evaluate the offspring population based on the objectives;
  Combine the parent and offspring populations into a temporary population;
  Calculate the Pareto-fronts and crowding distances of solutions inside the temporary population;
  Select solutions from the Pareto-fronts orderly while replacing them with solutions in the parent population until reaching Q;
end
Return a Pareto-solution from the first Pareto-front as a near-optimal solution;

conflict, when there exist an infinite number of Pareto-optimal solutions. A Pareto-optimal solution does not improve for one objective unless it satisfies others. The main goal in MOO problems is to find a finite Pareto-front in the objective space including a finite number of diverse Pareto-solutions.

Evolutionary Algorithms (EAs) are one of the most well-known meta-heuristic search mechanisms utilized for the MOO problems since their structure is free of search space and objective capacities [21]. EAs form a subset of evolutionary computation, in which they generally involve techniques and implementing mechanisms inspired by biological evolutions such as reproduction, mutation, recombination, natural selection, and survival of the fittest. The main advantage of EAs, when applied to solve MOO problems, is the fact that they typically generate sets of solutions, allowing computation of the entire Pareto-front. Currently, most Multi-Objective Evolutionary Algorithms (MOEAs) apply Pareto-based ranking schemes such as the Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) [22]. Algorithm 3 describes the procedure of running the multi-objective Knapsack procedure using the NSGA-II. The time complexity of approaching the Knapsack procedure using the NSGA-II is $O(G \times M \times Q^2)$, where G is the number of generations, M is the number of objectives, and Q is the population size.

The NSGA-II randomly generates an initial Pareto-population, and then, applies some evolutionary procedures such as tournament selection with crossover and mutation operators. Next, it generates an offspring population from parents in each generation. It classifies the temporary population, as the combination of parent and offspring populations, based on the dominance principle to some fronts f_1, f_2, f_3 and so on. A solution Sol_1 dominates a solution Sol_2, if Sol_1 is better than Sol_2 in some objectives and perhaps equal to others. All the solutions, which lie in one specific front are non-dominant. In addition, for each solution Sol_a in f_k, there exists a solution Sol_b in $f_{k'}$ such that Sol_b dominates Sol_a, where $k' < k$. In the last step, the NSGA-II fills the next generation's population starting from the first front and continuing with solutions in the next fronts. Since the size of the combined population is twice the new one, all fronts, which could be unable to accommodate are removed. However, it needs to handle the last allowed fronts, in which some of its solutions are possibly considered in the new population. In this situation, the NSGA-II uses a niching strategy to choose solutions of the last allowed fronts, which lie in the least crowded regions of the solution space. To this end, it finds the distance between each solution and its nearest left and right neighbors in the last allowed fronts for each dimension in the objective hyperspace. Finally, it sums up such distances for each solution as the largest hypercube around it, which is empty from other solutions. The largest hypercube shows a solution with the least crowd. Figure 4 elaborates a conceptual view of Pareto-fronts and Pareto-solutions with corresponding crowding distances.

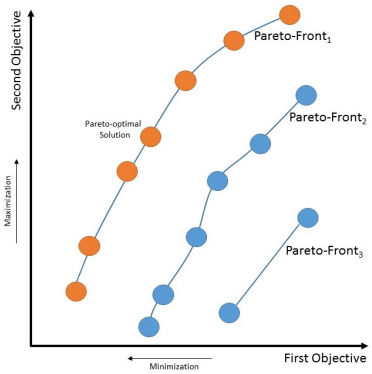
V. Simulation Setup and Analysis

This section first describes the simulation setup and subsequently, analyzes the results.

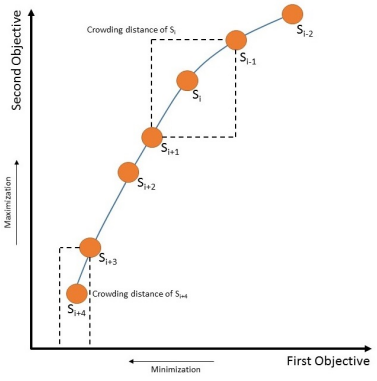
A. Simulation Setup

This work has been implemented with Matlab R2014b on a personal computer with an Intel Core i7-2.0 GHz CPU and 6 GB memory. Power profiles of all appliances are captured from the TraceBase open repository, which comprises a collection of real power traces of electrical appliances [23]. The electricity prices in the Danish day-ahead market, known as Elspot market, are provided by Nord Pool Spot with an hourly resolution on the day before the power delivery [24]. CO_2 emission intensity prognosis data are also provided in an hourly resolution by the Danish transmission system operator [25]. It is significant to note that the demand response system is set to receive the power demand requests at five-minute time intervals until finishing all activities. At each hour, it receives the power demand requests 12 times. As a result, T has been set to 24×12. $N = 100$ dwellings are assumed to provide their power demand requests over time.

A precise scenario for each dwelling is created randomly based on power profiles of appliances. Corresponding power



(a). Pareto-fronts and solutions



(b). Crowding distance of the Pareto-solutions

Figure 4. Conceptual view of Pareto-fronts and Pareto-solutions with corresponding crowding distances

demand requests are established in each scenario. To streamline the model, each appliance is operated only one time. Regarding flexibilities, we generate a random flexibility value for each appliance. A lower bound for each flexibility value is the following time interval from the moment, at which the operating cycles should finish without scheduling. An upper bound for each flexibility value is the end of the day.

It is considered that priorities are generated randomly. Figure 5 shows the aggregated power demand of the appliances of one dwelling in a typical day. Figure 6 shows the aggregated power demands of 100 dwellings. Peak power demand occurs at 20:30, which is 293 kW. Therefore, in order to allow all requested power demands at each time interval without shifting

TABLE III. SIMULATION CASE STUDIES INSPIRED FROM TABLE II

	Objective(s)
Case study 1	1) Minimizing the electricity cost
Case study 2	1) Maximizing the comfort level of consumers
Case study 3	1) Minimizing the electricity cost 2) Maximizing the comfort level of consumers

or interrupting any of them, the PDT should be at least 293 kW since it has been indicated that the PDT is constant and time-independent. However, the demand response system desires to flatten the aggregated demand by shifting power demand requests from on-peak periods to off-peak times. Therefore, it modifies the PDT to enable the shifting and interruption.

As described earlier, the MOEA includes some evolutionary parameters. As a selection operator, this paper utilizes the tournament selection. Linear crossover and exchange mutation are also utilized as the exploitation parameters. Their probabilities are set to p_c=80% and p_m=20%, respectively. Finally, the population size (Q) and the number of generations (G) are both adjusted to 100.

B. Simulation Analysis

This section analyzes the results obtained based on three simulation case studies, as Table III lists. The first case study is single-objective and aims to minimize the electricity cost as its objective function (see Equation (5)). The second case study is also single-objective and attempts to only maximize the comfort level of consumers (see Equation (7)). Finally, the third case study is multi-objective and intends to both minimize the total electricity cost and maximize the comfort level of consumers. We omit to show a case study including minimization of the CO_2 emission as an objective function since it would be similar to minimizing the electricity cost. The results will be analyzed based on variations of the PDT as follows:

$$PDT = \{(10\% \sim 100\%) \times PPD\}\,, \qquad (8)$$

where $PPD \in \mathbb{R}^+$ (watts) denotes the peak power demand. It is equal to 293 kW (see Figure 6). We change the PDT from 10% to 100% to analyze the obtained results. Hereinafter, when PDT is equal to R%, where $10\% \leq R \leq 100\%$, it means PDT=R$\times$PPD. We examine the effects of these variations on:

- Computation time of running the algorithm over time;
- Number of referrals to the Knapsack procedure;
- Computation time of the total number of referrals to the Knapsack procedure;
- Total electricity costs of the dwellings in a day;
- Deviation between the reception and delivery times of appliances;
- Aggregated power demands of the scheduled scenarios in a day.

Figure 7 analyzes the computation of running the scheduling algorithms based on different case studies. In Figure 7(a), according to Algorithm 1, non-shiftable power demand requests will be allowed to start or to continue apart from the assigned PDT. Considering computation time, when PDT is equal to 10%, the remaining capacity for allowing the remaining power demand requests is very low or even below zero.

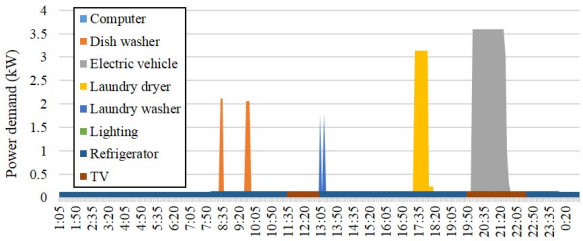


Figure 5. Aggregated power demand of appliances used in Table I

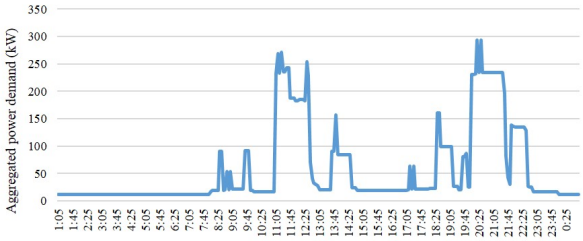
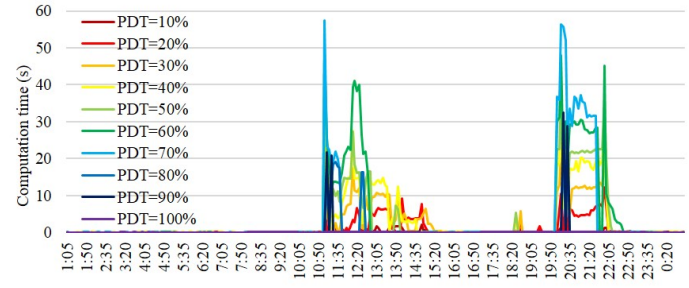


Figure 6. Aggregated power demands of 100 dwellings based on randomly generated scenarios in a typical day

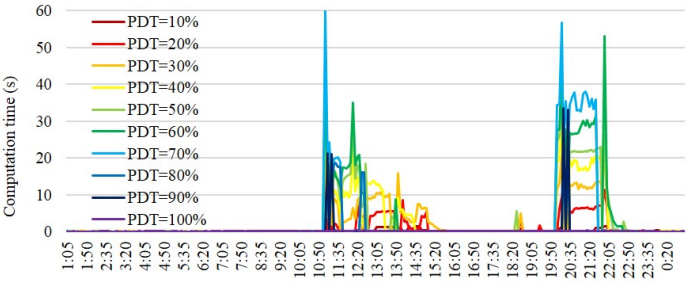
The reason is that the algorithm should satisfy Equations (1) to (4). Therefore, it is not possible to run the Knapsack procedure since the minimum consumption of the remaining power demand requests is greater than the remaining capacity. In the next intervals, the system, apart from the remaining capacity, should allow some power demand requests to start or to continue, for which shifting or interrupting them is not possible due to their deadline flexibility constraints. As a result, the number of remaining power demand requests as inputs to the Knapsack procedure will be few and, therefore, computation time will be lowered accordingly. Nevertheless, when PDT increases, the Knapsack procedure will allow more power demand requests to start or to continue at each interval. Some of these allowed power demand requests are members of the uninterruptible set. Therefore, at the next intervals, the system has to allow the corresponding appliances to continue their operation apart from the PDT. The demand response system will confront more remaining power demand requests compared to lower assigned PDT in later time intervals. This will increase the complexity and computation time of running the Knapsack procedure.

We experience more complexity and higher computation time, when assigned PDT increases. Nevertheless, the number of intervals, in which the Knapsack procedure should run decreases. Having some uninterruptible appliances and time limit flexibility constraints make this decreasing. If the system allows an uninterruptible power demand request to start at a certain time interval, it will be unable to interrupt it in the following intervals. Therefore, it will have to shift more power demand requests since the remaining capacity has decreased. These shifted power demand requests will be accumulated and, finally, the Knapsack procedure will face several remaining requests. When PDT is 90%, we observe a noticeable decrease in computation time compared to previous figures. The reason is the reduced amount of the Knapsack procedure's inputs. Since the aggregated power demands of the remaining power demand requests are less than the remaining capacity at most of the time intervals, it is not necessary to run the Knapsack procedure. Obviously, there is no need to run the Knapsack procedure at any of the time intervals, when the threshold is equal to 100%.

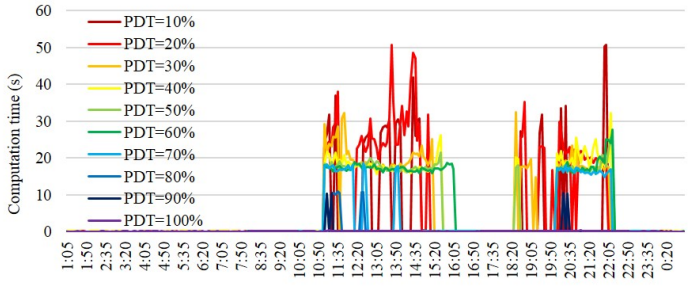
Figure 7(b) demonstrates the same analysis based on the second case study. The description of this figure is almost the same as Figure 7(a). However, there are some minor differences, which are linked to the differences in the nature of the objectives. The main reason is underlining the intention of consumers to pay for the highest comfort as little as possible. The computation time of running the third case study is illustrated in Figure 7(c). In contrast to Figures 7(a) and 7(b), here, the computation time is completely different. The main reason is the repetitive manner of the MOEA in finding the non-dominated near-optimal solution at each time interval. As described previously, there is no exact solution for multi-objective problems. Therefore, the near-optimal solutions ob-



(a). Computation time of running the scheduling algorithm based on the first case study



(b). Computation time of running the scheduling algorithm based on the second case study



(c). Computation time of running the scheduling algorithm based on the third case study

Figure 7. Computation time of running the scheduling algorithms based on different case studies

tained from running the algorithm at each time interval, affect the computation time of subsequent intervals. Computation times for the next intervals may change due to the randomized nature of finding near-optimal solutions. If all scenarios and relevant information are known before scheduling, it will be possible to limit the computation time. However, in this situation, when the system receives the power demand requests with a specific time resolution, it is not possible to do it since there is no future prediction or even forecasted data to learn before scheduling.

According to Figure 7(c), the computation time decreases, when PDT is 50% or more. The total power demand of remaining requests at 22:00 is a bit more than the remaining capacity. Also, most of the corresponding appliances are members of the uninterruptible appliances. Therefore, the Knapsack procedure's output comprises most of them. The demand response system should allow them to continue their duties at the next time intervals apart from the remaining capacity. This decreases the computation time at the next time intervals since the number of inputs to the Knapsack procedure decreases. As the final note, in this analysis, only 35% of the CPU speed and 400 MB of memory have been employed by the local power algorithm in all three case studies in the worst case.

Figure 8 analyzes the number of referrals to the Knapsack procedure in Algorithm 1. Figure 9 studies the corresponding computation time, when PDT changes. According to Figure 8, the number of referrals to the Knapsack procedure in the first two case studies is different, when PDT is equal to 10%. The reasons are first the reductive nature of Equation (5) and second the remarkable difference between the assigned PDT and the power demand of the remaining requests. When the threshold changes to at least 20%, uninterruptible power demand requests will roughly be allowed to start or to continue their work at the time they desire. Therefore, the number of inputs to the Knapsack procedure will decrease and the total number of referrals to the Knapsack procedure in the first two case studies will be almost the same. Now, due to the multi-objective nature of the third case study, the total number of referrals will also be more than previous case studies since the outcome solutions of the Knapsack procedure at each time interval are near-optimal.

According to Figure 9, the computation time of the total referrals to the Knapsack procedure increases when the number of referrals rises. However, this fact is applicable to only the first two case studies. The computation time of running the multi-objective algorithm is decreased when the number of referrals to the Knapsack procedure increases. Similar to the provided descriptions to Figure 7(c), this algorithm does not seek to obtain the optimal solution of the problem. As a result, the near-optimal solutions contain a mix of interruptible and uninterruptible power demand requests. Intuitively, the uninterruptible power demand requests will not be shifted to the next intervals and, therefore, the number of Knapsack procedure's inputs will decrease.

As the next analysis, Figure 10 displays the differences between the total electricity costs in the three case studies based on the variations in PDT. With respect to Figure 7, computation time increases nearly linearly when PDT changes. The total electricity cost is the same since the total number of interruptions decrease when the threshold increases. Thus, appliances start operating roughly at their desired time. This causes the peak times to remain over the time (see Figure 6). Nevertheless, with decreasing the PDT, some of the power demand requests should be shifted to the low price intervals, which result in decreasing the total electricity costs. As can be easily seen, the electricity cost is reduced for 1%, when PDTs are equivalent to 10% and 100%. Having almost low fluctuating Danish electricity prices make this very low reduction.

The third case study performs better in terms of electricity cost reduction. This is due to having a multi-objective problem. For instance, in the second case study, the algorithms tries to find an optimal solution at each time interval. An optimal solution should include the maximum number of possible power demand requests. However, this is different in the third case study since objectives are conflicting. Therefore, the solution's size is smaller, which causes the requests being shifted to lower electricity price periods.

Table IV analyzes the actual required PDT and the differences at peak time intervals when the assigned PDT changes based on Equation (8). The variation rates of PDT required for scheduling the power demand requests in first two case studies are almost the same. If we compare the maximum needed PDT in the first case study with the second one when assigned PDT rises, we observe that the gradients of maximum needed PDTs are almost similar to one another. Nevertheless, the decreasing gradient of PDT, when the system applies the third case study, is lower than the other case studies. The time interval, at which the peak demand happens, is equivalent in the first two case studies. This time interval is different in the third case study due to its multi-objective nature.

According to Equation (7), consumers desire to receive their appliances in the completed status at the time they expect. This expected time for each appliance is the sum of the time periods provided in the scenarios and the corresponding additional deadline flexibility period. However, it is not possible to satisfy all consumers due to some restrictions such as PDT. The average deviation between reception and delivery times of each appliance of each dwelling for all case studies is pictured in Figure 11. These waiting times do not result in a violation of the deadline flexibility constraint. Assigning 60% is beneficial to minimize the deviation between delivery and reception times of each appliance in the first two case studies. Consumers have to wait to receive their charged electric vehicle almost 30 minutes when PDT is 60%. For the multi-objective case study, if PDT is 80%, consumers should wait averagely almost 20 minutes for receiving their charged electric vehicle. It is worth emphasizing that these waiting times are in addition to the time it takes to actually charge the EV.

As the final analysis, Figures 12 demonstrates the aggregated consumption of the scenarios after applying the scheduling algorithm. The demand response system endeavors to flatten the aggregated power consumption over the day. According to Figure 12(a), it shows the best condition of aggregated power demand when PDT is equal to 60% (17 kW). If the system does not apply any scheduling algorithm on the received power demand requests, i.e., PDT is 100%, the total maximum consumption will be approximately 293 kW. It proves that the demand response system can reduce the peak demand by 40%. This fact is also applicable to the second case study shown in Figure 12(b). Finally, it is worthy to note that since the complexity of the multi-objective case study is high, it needs a high PDT. Figure 12(c) pictures the

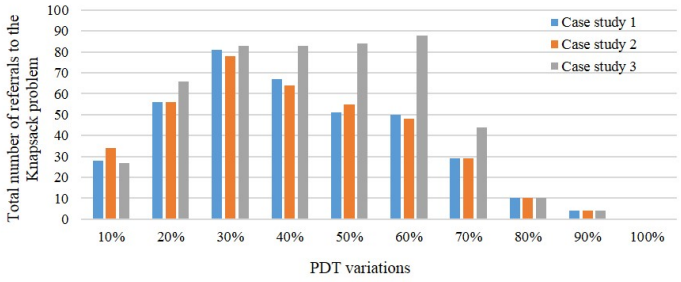


Figure 8. Total number of referrals to the Knapsack procedure in Algorithm 1

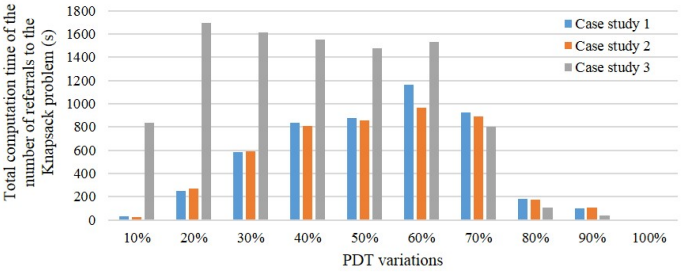


Figure 9. The computation time of referrals to the Knapsack procedure in Algorithm 1

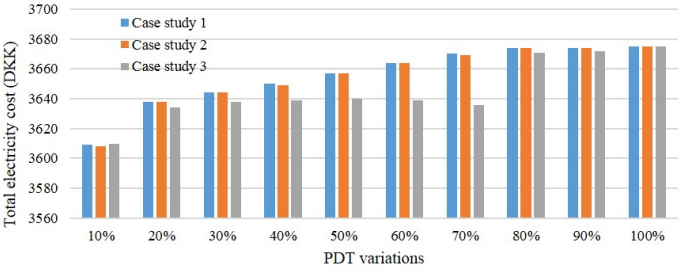
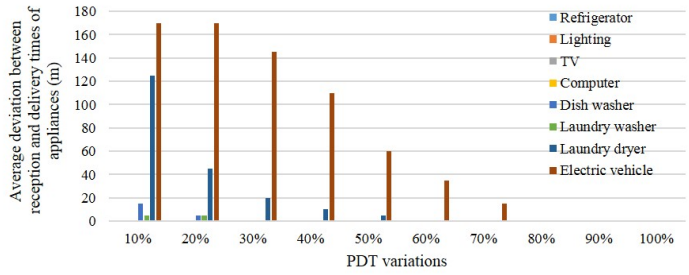
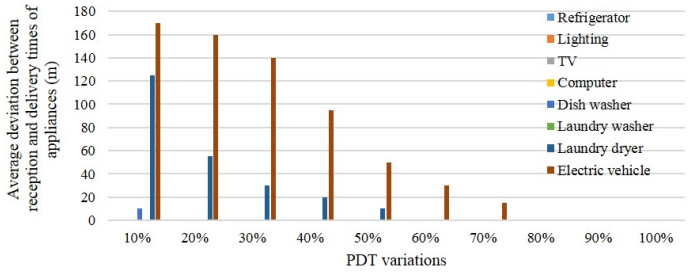


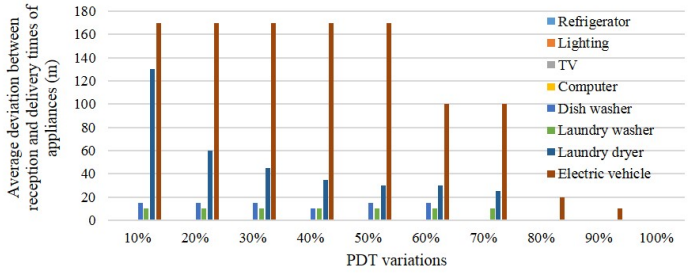
Figure 10. Total electricity costs of dwellings in a day based on three case studies



(a). Deviation time between appliance delivery and reception times based on the first case study



(b). Deviation time between appliance delivery and reception times based on the second case study

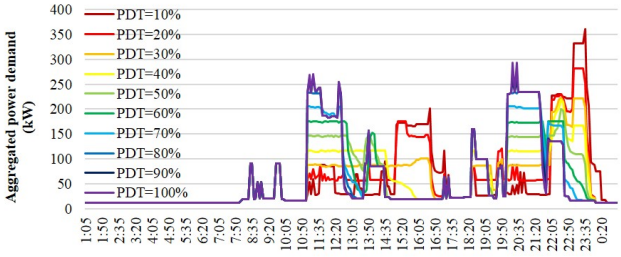


(c). Deviation time between appliance delivery and reception times based on the third case study

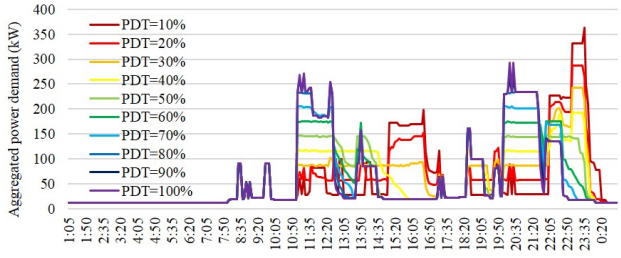
Figure 11. Deviation time between appliance delivery and reception times based on different case studies

TABLE IV. MAXIMUM NEEDED PDT AND CORRESPONDING PEAK TIME INTERVAL WHEN THE ASSIGNED PDT CHANGES

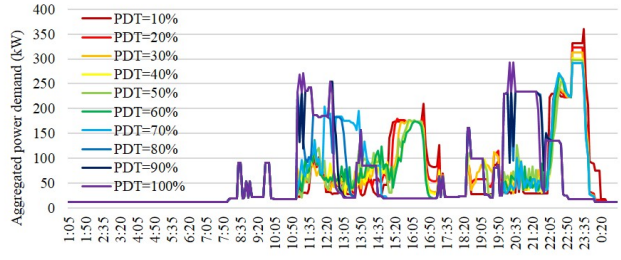
		Assigned PDT									
		10% 293 kW	20% 586 kW	30% 879 kW	40% 117 kW	50% 146 kW	60% 175 kW	70% 205 kW	80% 234 kW	90% 263 kW	100% 293 kW
Maximum needed PDT	Case study 1	360 kW	282 kW	222 kW	218 kW	200 kW	175 kW	205 kW	234 kW	263 kW	293 kW
	Case study 2	363 kW	287 kW	242 kW	192 kW	155 kW	175 kW	205 kW	234 kW	263 kW	293210
	Case study 3	360 kW	322 kW	213 kW	300 kW	298 kW	291 kW	291 kW	234 kW	254 kW	293 kW
Peak time interval	Case study 1	23:35	23:05	22:30	22:30	22:30	11:05	20:20	20:35	11:20	20:30
	Case study 2	23:35	23:05	23:05	23:05	22:35	11:20	11:05	20:35	11:20	20:30
	Case study 3	23:35	23:05	23:05	23:05	23:05	23'05	23:05	20:35	12:30	20:30



(a). Aggregated power demand of 100 dwellings based on the first case study



(b). Aggregated power demand of 100 dwellings based on the second case study



(c). Aggregated power demand of 100 dwellings based on the third case study

Figure 12. Aggregated power demand of 100 dwellings based on different case studies

aggregated consumption of 100 dwellings when the system applies the third case study. In this figure, the demand response system will receive the minimum aggregated power demand, when PDT is equal to 80%. In this status, the maximum power demand is 234 kW and the achievement is 20%.

VI. Conclusion and Future Work

This paper developed a demand response system. It received power demand requests of appliances continuously and scheduled them accordingly. Appliances are classified based on the shiftability and interruptibility features. The well-known "0-1" Knapsack procedure has been applied to the scheduling problem, when it is necessary to choose some requests to allow them to start or to continue their duties at the current time interval and shift the remaining to the future time intervals. The objectives of the proposed scheduling algorithm are minimizing the total electricity costs and CO_2 emission intensities coupling with maximizing the satisfaction of consumers. In addition, as constraints, the system attempts to keep the total power demands under a constant and time-independent power demand threshold provided by distribution system operators at each time interval. Consumers may provide time limits of flexibilities of electricity powers to the demand response system. These time limit flexibilities of power demand requests vary among appliances. It helps the system to find an optimal or near-optimal solution (based on the approach used) to decide when to shift or to interrupt power demand requests. The results were analyzed based on changing the thresholds. It was confirmed that applying this kind of threshold led to a reduction in the total electricity costs, a change in the daily behavior of consumers in a beneficial way, and additionally, a flattened aggregated power demand.

An investigation of reformulating the current power scheduling algorithm to a hierarchical scheduling algorithm to run in each dwelling is a promising future work. It would be also interesting to investigate the sensitivity of the scheduling algorithm to the stochasticity of power profiles. In practice, the adaptation of power demand thresholds can be accomplished by implementing a control loop between the demand response system and a gateway deployed in each dwelling.

Acknowledgment

The work in this document has been funded by European Union Seventh Framework program (FP7/2007-2013) under grant agreement n° 619560 (SEMIAH).

References

[1] R. H. Jacobsen, A. G. Azar, Q. Zhang, and E. S. M. Ebeid, "Home Appliance Load Scheduling with SEMIAH," in the Fourth International IARIA Conference on Smart Systems, Devices, and Technologies (SMART), 2015, pp. 1–2.

[2] J. Gao, Y. Xiao, J. Liu, W. Liang, and C. P. Chen, "A Survey of Communication/networking in Smart Grids," Future Generation Computer Systems, vol. 28, no. 2, 2012, pp. 391–404.

[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, 2014, pp. 490–503.

[4] M. Rastegar, M. Fotuhi-Firuzabad, and F. Aminifar, "Load Commitment in a Smart Home," Applied Energy, vol. 96, 2012, pp. 45–54.

[5] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, "Scheduling Policies for Two-State Smart-Home Appliances in Dynamic Electricity Pricing Environments," Energy, vol. 69, 2014, pp. 455–469.

[6] P. Stoll, N. Brandt, and L. Nordström, "Including Dynamic CO_2 Intensity With Demand Response," Energy Policy, vol. 65, 2014, pp. 490–500.

[7] X. He, L. Hancher, I. Azevedo, N. Keyaerts, L. Meeus, and J.-M. GLACHANT, "Shift, Not Drift: Towards Active Demand Response and Beyond," 2013.

[8] H. Farhangi, "The Path of the Smart Grid," IEEE Power and Energy Magazine, vol. 8, no. 1, 2010, pp. 18–28.

[9] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid-The New and Improved Power Grid: A Survey," IEEE Communications Surveys & Tutorials, vol. 14, no. 4, 2012, pp. 944–980.

[10] P. Palensky and D. Dietrich, "Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads," IEEE Transactions on Industrial Informatics, vol. 7, no. 3, 2011, pp. 381–388.

[11] A. Agnetis, G. Dellino, P. Detti, G. Innocenti, G. de Pascale, and A. Vicino, "Appliance Operation Scheduling for Electricity Consumption Optimization," in Proceedings of 50^{th} IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), 2011, pp. 5899–5904.

[12] G. O'Brien and R. Rajagopal, "A Method for Automatically Scheduling Notified Deferrable Loads," in Proceedings of IEEE American Control Conference (ACC), 2013, pp. 5080–5085.

[13] A. G. Azar, R. H. Jacobsen, and Q. Zhang, "Aggregated Load Scheduling for Residential Multi-Class Appliances: Peak Demand Reduction," in Proceedings of the 12^{th} International IEEE Conference on the European Energy Market-EEM, 2015.

[14] O. A. Sianaki, O. Hussain, and A. R. Tabesh, "A Knapsack Problem Approach for Achieving Efficient Energy Consumption in Smart Grid for End Users' Life Style," in IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES), 2010, pp. 159–164.

[15] F. Rassaei, W.-S. Soh, and K.-C. Chua, "Demand Response for Residential Electric Vehicles With Random Usage Patterns in Smart Grids," IEEE Transactions on Sustainable Energy, vol. 6, no. 4, 2015, pp. 1367–1376.

[16] M. T. Beyerle, J. A. Broniak, J. M. Brian, and D. C. Bingham, "Manage Whole Home Appliances/loads to a Peak Energy Consumption," 2011, US Patent App. 13/042,550.

[17] H. Kellerer, U. Pferschy, and D. Pisinger, Knapsack Problems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, ch. Introduction to NP-Completeness of Knapsack Problems, pp. 483–493. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24777-7_16

[18] K. Florios, G. Mavrotas, and D. Diakoulaki, "Solving Multi-objective, Multi-constraint Knapsack Problems Using Mathematical Programming and Evolutionary Algorithms," European Journal of Operational Research, vol. 203, no. 1, 2010, pp. 14–21.

[19] R. E. Bellman and S. E. Dreyfus, Applied Dynamic Programming. Princeton university press, 2015.

[20] K. Deb, Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques. Boston, MA: Springer US, 2014, ch. Multi-objective Optimization, pp. 403–449. [Online]. Available: http://dx.doi.org/10.1007/978-1-4614-6940-7_15

[21] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of Multi-objective Evolutionary Algorithms on Many-Objective Knapsack Problems," IEEE Transactions on Evolutionary Computation, vol. 19, no. 2, 2015, pp. 264–283.

[22] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, 2002, pp. 182–197.

[23] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, and R. Steinmetz, "On the Accuracy of Appliance Identification Based on Distributed Load Metering Data," in Proceedings of the 2nd IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT), 2012, pp. 1–9.

[24] Nord Pool Spot. Last access on May 4, 2016. [Online]. Available: http://www.nordpoolspot.com

[25] Energinet. Last access on May 4, 2016. [Online]. Available: http://www.energinet.dk/

Paper E

Design of an Event-Driven Residential Demand Response Infrastructure

The paper presented in this chapter has been accepted as a conference publication [6].

[6] Rune Hylsberg Jacobsen, Armin Ghasem Azar, and Emad Samuel Malki Ebeid, "Design of an Event-Driven Residential Demand Response Infrastructure," In *IEEE Euromicro Conference on Digital System Design (DSD)*, 2016, pages 38-45, doi: 10.1109/DSD.2016.105

Authors contribution: Rune Hylsberg Jacobsen conceived the initial design of the event-driven demand response infrastructure; Armin Ghasem Azar carried out the literature survey in selected areas and formulated the buffering sub-system, SALSA algorithm, prototype implementation, and the case study; Rune Hylsberg Jacobsen and Armin Ghasem Azar were the main editors of the manuscript; Armin Ghasem Azar and Emad Samuel Malki Ebeid contributed with the text in selected areas and to the reviewing process.

2016 Euromicro Conference on Digital System Design

Design of an Event-Driven Residential Demand Response Infrastructure

(Invited Paper)

Rune Hylsberg Jacobsen, Armin Ghasem Azar, and Emad Samuel Malki Ebeid
Department of Engineering, Aarhus University, Denmark,
Emails: {rhj, aga, esme}@eng.au.dk

***Abstract*—Event-driven demand response programs reward consumers for shifting their electricity loads upon requests from an aggregator, a balance responsible party, or a distribution system operator. In order to have any significant impact on the smart grid, the flexibility exhibited on the demand-side needs to be aggregated from a large number of consumers. This poses significant scalability challenges for the ICT infrastructure that controls flexible electricity loads. This paper reports from the European research project SEMIAH. The project aims to design a scalable infrastructure for residential demand response. The study presents progress towards the system design of a centralized load scheduling algorithm for controlling home appliances over time. The demand response system takes the power grid constraints and satisfaction of consumers into account. Simulation results from a case study quantify the potential impact of a residential demand response program.**

***Keywords*—Smart grids, residential demand response, scalability, load scheduling.**

I. Introduction

In the last two decades, the demand for electricity has risen exponentially, and it will continue to grow remarkably. In Europe, the electricity demand in the residential sector is expected to rise on average 56% from its 2000 level until 2050 with an annual growth rate of 1.1% [1]. Although many appliances are becoming more efficient, their number is increasing and they are used more often and for longer periods of time. Moreover, many appliances have more functions or special features that require more energy. By 2050, it is estimated that 23% of total electricity demand will be consumed by households compared to today's level of 12-14% [1].

The Energy Roadmap 2050 strategies aim to reduce "greenhouse gas emissions to 80-95% below 1990 levels by 2050" [2]. Government policies are furthermore setting costs on CO_2 emissions to increase the share of Renewable Energy Sources (RES) for power generation significantly in order to reach a secure, competitive and decarbonized energy system in 2050. Energy utilities will therefore face significant challenges to meet the increasing energy demand while at the same time comply with the EU energy policies with respect to greenhouse gas emissions.

Advancement in technologies and new services for the smart grid enable novel solutions for energy system integration while respecting the stability and security needed in the context of an increasing share of RES in the electricity grid. The European research project SEMIAH (Scalable Energy Management Infrastructure for Aggregation of Households) strives to develop an open Information and Communication Technology (ICT) infrastructure for the implementation of residential demand response [3]. The paper extends the SEMIAH aggregator framework presented in [3].

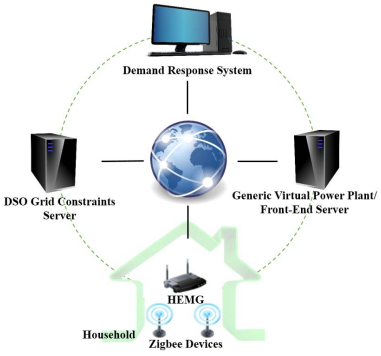


Fig. 1: Overview of the SEMIAH demand response distributed infrastructure.

The infrastructure exploits the flexibility provided by consumers in doing work with their home appliances and through the flexibility provided by thermal appliances such as heating/cooling systems. An expected impact of the implemented Demand Response System (DRS) will result from the matching of supply with demand for load management and energy efficiency. Hereby, SEMIAH will facilitate a reduction of peak demand and increase the use of RESs through flexible demand.

Fig. 1 shows an overview of the distributed nature of a SEMIAH demand response infrastructure. Load demands of electrical appliances can be categorized based on the *shiftability* feature [4]. Shiftability promotes a DRS to shift load requests of *shiftable* appliances to a future time interval. Typical flexible loads are wet appliances, heating, and pumping devices [5]. Some appliances can hardly be shifted, for instance, the refrigerator, while others represent appliances where residents do not offer any flexibility such as cooking.

978-1-5090-2817-7/16 $31.00 © 2016 IEEE
DOI 10.1109/DSD.2016.105

However, such systems consume high amounts of electricity and shifting the operation even for small amounts of time has therefore, a high impact. Shiftable appliances can be further divided into groups based on the *interruptibility* feature. As an example, the DRS can both shift and interrupt the charging cycle of an electric vehicle. However, it should run *uninterruptible* appliances to completion when these are started, e.g., a laundry machine. When an appliance is plugged in, the household applies a deadline flexibility constraint, which sets a contract between the DRS and the consumer when the given appliance must complete its operation at latest.

Communication between appliances and the DRS is based on an event-driven approach. Each appliance starts sending load requests to the DRS which runs with a specific time resolution. At each time interval, the DRS produces a schedule solution for the aggregated set of load requests with respect to flexibilities offered, grid constraints, and grid incentives such as real-time electricity prices. To make such solutions, the DRS employs an algorithm named SALSA (Scalable Aggregation of Load Schedulable Appliances). SALSA runs an event selection procedure to investigate which load requests should be decided to operate.

The remaining parts of the paper is organized as follows: Section II presents the related work. Section III describes the design of the DRS and Section IV introduces the implementation. In Section V, we evaluate the DRS based on a case study. Finally, the paper is concluded in Section VI.

II. Related Work

Electrical grids increasingly depend on RES production that is weather-dependent, often fluctuating and difficult (or impossible) to plan and control. To smoothen RES production, the use of physical storage (e.g., batteries and electrical vehicles) and virtual storage (e.g., demand response systems) are currently being considered [6], [7].

As significant energy amounts are involved and substantial flexibility (elasticity) is available to operate the devices within the allowed bounds, and due to their broad availability, storage loads are of great value [8] for different smart-grid applications, e.g., demand and supply balancing [9], grid congestion problem solving [10], and electricity market trading [11]. However, large storage systems are still considered to be too costly and demand response has been considered to a feasible way to provide a cost-effect virtual storage.

Today, most commercial demand response services in Europe are leveraged by large consumers that e.g., find flexibility potentials in their production processes. More recently, attention has shifted towards smaller consumers such as residential households. Consumers may be engaged in event-driven (also known as incentive-based) demand response programs [12]. In these programs, home appliances can be invoked in response to a variety of trigger conditions, including environmental parameters (e.g., temperature); local or regional grid congestion; economics; or operational reliability requirements. For demand response programs that foster an improved integration of renewable energy, aggregators can shift consumption to periods with lower CO_2 intensity or electricity prices.

Vardakas *et al.* [7] presented a survey on different demand response schemes and programs according to their control mechanism, offered motivations, and decision variable. Moreover, the authors reviewed various optimization algorithms for optimal operation of the smart grid.

Di Giogio and Pimpinella [13] presented the design of a smart home controller strategy providing efficient management of electric energy in a domestic environment. Their work provided an integration between ICT and automation to implement a pervasive control platform, which allowed consumers to automatically fulfil the terms of previously subscribed contracts, while assuring cost-effective use of energy. The deployed controller is event-driven which means that it reacts to events from the environment such as requests from the consumer for the execution of loads and signals from a demand-side control.

Early forms of event-driven demand response include the old-fashioned ripple control acting as an emergency reserve [14]. Ripple control aims to protect power systems in the emergency conditions caused by critical contingencies. Wang *et al.* [15] proposed an event-driven emergency demand response scheme to improve the stability of the power system from experiencing voltage collapse. The proposed scheme was able to provide key setting parameters such as the amount of demand reductions at various locations to prepare the demand response infrastructure and hereby act as a balancing asset for the grid in case of emergency.

A more novel approach used ICT in a service-oriented approach to offer smart grid applications that enabled services to be deployed in a distributed way over the Internet. A recent European research project SmartHG has addressed the challenges of deploying residential energy management as an integrated set of intelligent automation services deployed in a cloud infrastructure (the SmartHG Platform) [16]. The set of interrelated service aims to steer energy demand of residential users in order to: keep operating conditions of the power grid within given healthy bounds, minimize energy costs, and minimize CO_2 emissions from electricity generation. This is achieved by creating demand-awareness based on consumption habits (as for electricity production/consumption) of residential users as gained from sensing and communication infrastructure. The control policies forming the backbone of such software service-based methodology has proven to be feasible [17]. At the top tier, the Distribution System Operator (DSO) sets operational constraints for the low-voltage/medium-voltage distribution grid and retrieves a power profile (i.e., power grid constraints) for each residential user from the service platform. At the bottom tier, the SmartHG platform monitors and controls smart devices in the residential households in order to keep the household power demand within its power profile at all times.

III. System Design

Fig. 2 gives an overview of the SEMIAH system design. The infrastructure complies with the system architecture previously published in [3]. The system is agnostic to the choice of Virtual Power Plant (VPP) platform and the aggregator operating the demand response program has the possibility of choosing between different VPP vendors. The system implements a Front-End Server (FES) to provide provisioning and monitoring down to household level. In addition, a *back-end* system is implemented as an open loop control to coordinate and provide schedules to the *front-end* system.

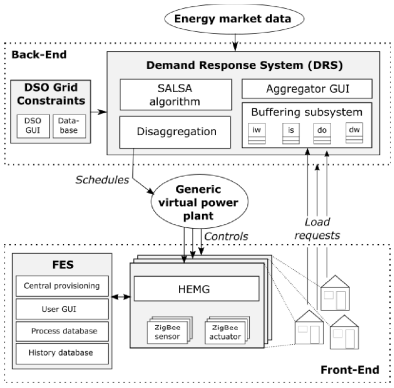


Fig. 2: Overall SEMIAH system design.

In the following, the front-end system is briefly introduced. Hereafter, a more thorough presentation of the back-end is given. Communications, which take place for both the front and back-end systems are shown with arrows. The mechanism on which load requests are delivered to the back-end is beyond the scope of this paper.

A. Front-End Design

The front-end system consists of home appliances, and a Home Energy Management Gateway (HEMG) that connects them with the back-end system. The HEMG is a flexible solution for connecting networks based on different Home Area Network (HAN) technologies. It is a programmable Linux hub with Java and Open Services Gateway initiative (OSGi) support. ZigBee has been chosen as the HAN technology preferred for communication with sensors, actuators, and HEMGs in the households.

A FES is deployed as a Software as a Service (SaaS) infrastructure to serve a neighborhood of households. The FES is the basis for the central provisioning, the system administration, and the database & analytics functions. It stores the current and historical state of front-end systems including recording of time series of measurement results, history of set points' updates, connection and disconnection of HEMGs. The central provisioning service offers a set of rules to identify any faulty behavior of the DRS. The FES stores data needed for system operation monitoring and analysis. Appropriate user data can be presented to system administrators and consumers by using the web-based Graphical User Interface (GUI). A database & analytics function is responsible for management of set points as they change. It relies on a history database that stores the past sensors and actuators values as well as a process database that keeps track on the current status of sensors and actuators.

The FES does not take direct part in the demand response operations but rather serves as a monitoring service for the program. The central provisioning provides basic functions for remote HEMG software maintenance and updating.

B. Back-End Design

The back-end of the DRS infrastructure consists of a set of distributed services interconnected over the Internet using open standards communication protocols in particular secure HyperText Transfer Protocol (HTTP) over Transmission Control Protocol (TCP). Data exchange between back-end services is primarily accomplished by exchange of eXtendible Markup Language (XML) documents. Communication is secured by using the Transport Layer Security (TLS) protocol.

The distributed nature of the back-end allows system parts to have independent product life cycles. The back-end encompasses the DRS, which essentially provides a third party implementation of a scheduling algorithm for actuation of electricity loads that can work with the Generic VPP (GVPP). Furthermore, it serves DSO grid constraints including its configuration GUI and corresponding database. Finally, the back-end communicates with the GVPP by sending schedule solutions produced by the DRS.

Our starting point is the home appliances providing flexibility to the demand response aggregator. Each appliance can generate different types of load profiles with a specific time resolution. Load profiles are generated randomly with a normal distribution. Taking these load profiles, a specific electricity usage pattern is created for each consumer for a whole day. The back-end receives the load request from appliances in real-time based on events created in the households. The following describes how these components behave in the back-end system.

1) Buffering Sub-System: The buffering system is composed of four various buffers named *immediately wait* (*iw*), *immediately start* (*is*), *decided to operate* (*do*), and *decided to wait* (*dw*). It stores the load requests, coming from the households (cf. Fig. 2), into the *iw* buffer. These buffered load requests are forwarded to SALSA to be scheduled. The *is* buffer is specialized for load requests of non-shiftable appliances while the last two buffers are designed for those load requests which are sending from shiftable appliances. A load request is moved to the *do* buffer if the SALSA algorithm decides to operate the corresponding appliance at that time interval. Otherwise, the load request will be moved to the *dw* buffer. At the next time interval, the buffering system removes the load requests from the *dw* buffer and appends them to the *iw* buffer. Indeed, the SALSA will decide about appended load requests and newly arrived load requests simultaneously. At each time interval, the DRS sends the schedules to the GVPP based on load requests located in the *do* and the *dw* buffers.

2) SALSA Algorithm: SALSA is a key component of the DRS. It follows an event-based approach, where it communicates with the buffering subsystem to decide about the buffered load requests at each time interval. SALSA fetches load requests from *iw* buffer at each time interval. All load requests whose corresponding appliances are non-shiftable are decided to operate immediately without undergoing any scheduling. Then, it removes these load requests from *iw* buffer and places

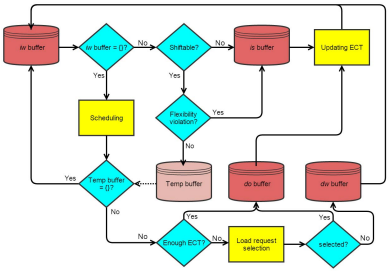


Fig. 3: Flowchart of the SALSA algorithm.

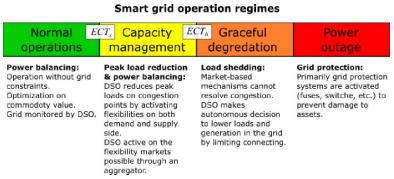


Fig. 4: Adaptation of the ECT concept to the USEF operating regimes. Adapted from [18].

them in *is* buffer. Afterwards, SALSA investigates whether deciding to postpone an appliance at the current time interval violates the relevant flexibility or not. If so, the load request is moved to *do* buffer. Otherwise, it is added to a temp buffer. When the *iw* buffer is emptied at the current time interval, if temp buffer is empty, it moves to the next time interval. Otherwise, it determines whether the remaining Electricity Consumption Threshold (ECT) can handle the load requests stored in the temp buffer or not. If there is any possibility, all load requests will be appended to *do* buffer. On the contrary, SALSA executes a load request selection procedure to allow some load requests to operate (will be stored in *do* buffer) and shift others to the future time interval (will be stored in *dw* buffer).

3) DSO Grid Constraints: Many electricity producers are experiencing an insufficiency of electricity generation capacity because of the load requests by consumers. More accurately, the generated amount of electricity is often unable to satisfy the requested loads at a specific time interval. Therefore, the DSO currently applies an ECT policy in order to overcome the mentioned outages and shortcomings [19].

The DSO Grid Constraints is a service for configuring grid constraints for a collection of households belonging to an aggregator. The interface is used by DSOs to define boundaries for the operation of the scheduling of power consumption. This interface complies with the smart grid operating regimes as defined by Universal Smart Energy Framework (USEF) [18]. These operation regimes are shown in Fig. 4. As the electricity demand increases the grid operation moves gradually from the Normal Operations scheme to the Capacity Management regime. This demarcation point is marked by the *soft* ECT (ECT_s) and triggers the scheduler to start scheduling electricity loads for the future. At the same time, the regulating electricity markets start planning a significant role by providing peak load reduction and power balancing between electricity supply and demand through market mechanisms. Should the DSO not be able to successfully manage capacity, there is a risk of hitting the second demarcation point in which the regime of Graceful Degradation starts. This is marked by the *hard* ECT (ECT_h). When crossing this boundary the DSO may take action to curtail load regarding any Service Level Agreement (SLA) promises just to secure the grid supply and to regain control of the network.

C. Data Models

The designed aggregator infrastructure is built on pre-existing components and should interact with external systems. These facts influence the definition of data models. Concretely, the front-end is based on the OGEMA framework [20]. OGEMA has already its own data model (OGEMA data model elements are called resources). OGEMA resources can be extended to cover SEMIAH needs but existing resources should be used whenever possible. Furthermore, grid transport, distribution systems, and energy management systems such as a VPP have their own data model. When relevant, DRS data models have been based on the Common Information Model (CIM) semantics [21], [22], to ease integration into grid operation and energy markets. CIM is an open standard that defines how managed elements in an information technology environment are represented as a common set of objects and relationships between them. Within the energy domain, CIM can be used for representing power system components and networks which has been primarily developed by the Electric Power Research Institute (EPRI) [21], [22].

Data modeling is a key aspect of SEMIAH to ensure interoperability between stakeholders systems. In accordance with the recommendation for European smart grids, the data models in the SEMIAH project will be based on the CIM. Unified Modeling Language (UML) has been selected as a language for the expression of the data models, because UML class diagrams allow the definition of data models and also because CIM is expressed in UML. The SEMIAH back-end uses the data model as a standard that was developed by the electric power industry and that has been adopted by the International Electrotechnical Commission (IEC). The core package of CIM is defined by the standard IEC 61970-301 [21].

Fig. 5 shows the data model of SEMIAH based on IEC CIM. The VPP has different sub elements like feeder or clusters of households, which are modeled as Facility. Each Facility has a connection to a GeneratingUnit. The GeneratingUnit has one or more Measurements to represent the characteristics of an element.

The core object for modeling measurements and schedules is the Analog object show in Fig. 6. Every measurement or

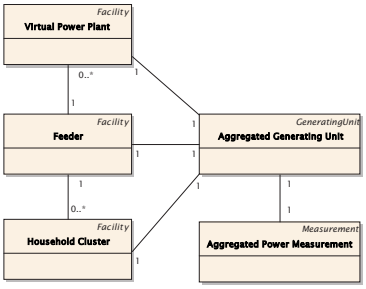


Fig. 5: SEMIAH CIM compliant data model.

schedule represent its core data in a time series of values. With an additional type of data together with UnitMultiplier and UnitSymbol, it is distinguished between these different types of an Analog. The Analog has a createdDateTime which is treated differently. In case of measurements and schedules this field can be neglected, because every AnalogValue has the complete information of the timestamp where the given value is valid for. The Analog-Class is derived from Measurements. The UML diagram of the Analog can be found in Fig. 6.

IV. Prototype Implementation

The main motivation for developing the demand response prototype is to validate the feasibility of the DRS, evaluate its scalability and performance and test the connectivity of it with DSO grid constraints and other components. Moreover, SEMIAH project aims to establish a research platform for understanding the feasibility and practicality of demand response and load shifting in residential household on a large scale. This is needed to build an ICT infrastructure that can support the emerging business of European aggregators acting in the liberalized electricity market.

Fig. 7 shows Energy Management System (EMS) interface with consumers from where they can enable flexibility of their appliances. Enabling each appliance activates the corresponding part in the lower half of the GUI. Consumers can set at which time they are interested to operate their appliances by Start Time. The corresponding operating program is set in Program. Finally, flexibility is provided in the Flexibility box. After successfully generating load profiles, pressing Next to DSO button accompanies you to reach the next GUI.

SALSA receives a number of consumers from the DSO. Fig. 8 presents the GUI of DSO which is connected to SALSA. Electricity prices and CO_2 emission intensities of the current day are fetched from Energinet and Nord Pool Spot[1]. The corresponding table shows their values where the first column relates to electricity prices while the second column lists CO_2

[1]http://www.nordpoolspot.com

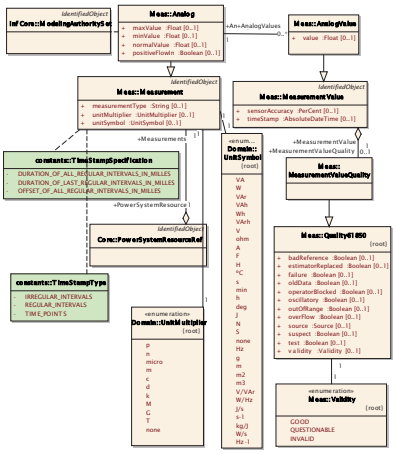


Fig. 6: Model of timeseries for data exchange in the back-end.

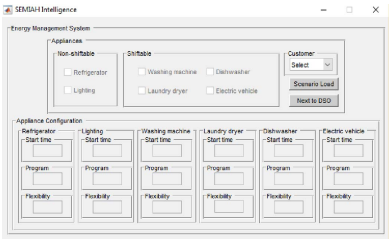


Fig. 7: GUI for the consumer.

emission intensities for today. The same procedure is also applicable to DSO Grid Constraints. Here, the first column shows the values of the soft ECT while the second column lists the values of hard ECT for today. These values are fetched from the XML file produced by the web application of the DSO grid constraint.

SALSA starts scheduling generated scenarios, as shown in Fig. 9. The upper diagram shows the aggregated load consumption of consumers before (red) and after (green) scheduling. Black and blue lines are hard and soft ECT, respectively. Lower diagram shows aggregated load consumption

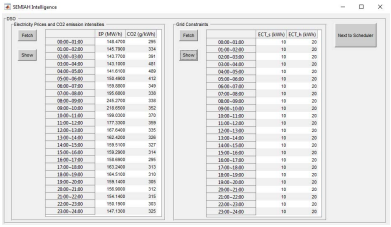


Fig. 8: GUI for DSO.

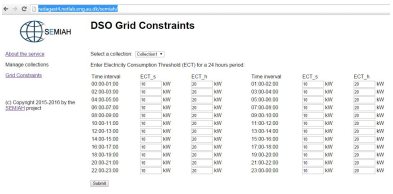


Fig. 10: Snapshot of DSO grid constraint web application.

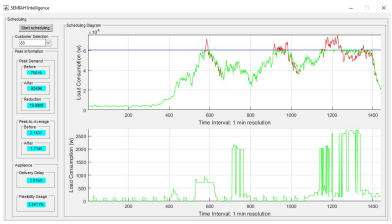


Fig. 9: Scheduling control GUI.

of a chosen consumer before and after scheduling. Finally, left column shows the various peak and demand response information. Appliance delivery delay shows an average time in minutes where each consumer should wait more to receive his/her appliances in a completed state (comparing with the time at which he/she expects before providing the flexibility). Consumer flexibility usage indicates how much percentage SALSA has used the provided flexibilities.

The DSO Grid Constraints is a service to configure grid constraints for a collection of households belonging to an aggregator. There can be a different set of constraints for each collection (see Fig. 10). ECT_s and ECT_h correspond to soft and hard ECTs, respectively. For each day, there is a time-series analog including 24 values each per hour. Submitting the adjusted constraints results in producing an XML file, as shown in Fig. 10. Grid constraints are given as energy in kWh/h for particular time slots during a 24 hour period.

V. Case Study

This section presents a case study to demonstrate the applicability of the proposed demand response infrastructure. Two main consecutive validation steps are studied. In the first step, a constant number of households are simulated. Moreover, three independent ECT policies are defined to enable the scheduling of households' consumptions. The first step evaluates the results according to different demand response performance criteria. It confirms the usability and profitability of the infrastructure. Second step depends on the results of the first step. One of the ECT policies is chosen to analyze the sensitivity of SALSA algorithm toward increasing the number of households. This step is two-fold. First, the impact of increasing the number of households on the same demand response performance criteria is investigated. Then, it is shown that how this increasing influences the computation time of running the SALSA algorithm. Therefore, this step validates the scalability of the infrastructure.

A. Step 1

The case study uses the BehavSim as a consumption simulation tool [23]. It can simulate 20 different appliances. For each, it generates various types of load profiles with a normal distribution at *one-minute* time resolution. Taking these load profiles, a specific daily electricity usage pattern is created for each household. The framework proposed has been modeled and coded in Matlab R2016a. It runs on a single CPU core (Intel Core i7 2.0 GHz) with 12 GB memory. In the first step, 100 households are simulated including three appliances: dishwasher, washing machine, and laundry dryer. It is considered that each appliance is operated once upon a day. To allow the DRS control these appliances, consumers are flexible to receive their appliances in the finished status until the midnight. SALSA intends to schedule load requests based on optimizing the comfort level of consumers. It uses the dynamic programming technique to schedule the load requests. This provides the optimal scheduling solution at each time interval. For more information, the reader is referred to [24].

Three ECT policies are defined as follows:

- **ECT-P1**: This policy aims to keep the aggregated consumption below a constant threshold over the day. Let us assume that the peak demand consumption of 100 households in the previous day is already known. According to the results achieved in [24], 60% of this value is considered as the current day's ECT. This policy is called a day-ahead approach.
- **ECT-P2**: This policy follows the real-time normalized electricity prices over time. At each time interval, ECT is the multiplication of the aggregated consumption of incoming load requests and the normalized electricity price. Obviously, the DRS attempts to allow more load requests to operate during periods with low electricity price and shift load requests during periods with high

electricity price. This policy is called a real-time approach.

- **ECT-P3**: This is basically similar to **ECT-P2**. The difference is in the nature of the trigger. This policy uses the real-time normalized CO_2 emission intensities to schedule the incoming load requests over time. This policy is also called a real-time approach.

Table I analyzes the performance of SALSA in scheduling 100 households based on three ECT policies. It is worth noting that values listed in Table I depend on the nature of randomly generated load profiles. AAS shows the Average Amount of Shifting during scheduling. Fluctuations in the electricity prices provide more flexibility to the SALSA to shift load requests from periods with high electricity prices to periods with relatively low prices. However, since there is no considerable fluctuation in CO_2 emission intensities, the SALSA cannot gain so much of shifting, when it utilizes ECT-P3. PDR analyzes the Peak Demand Reduction percentage. ECT-P1 gives better results compared to other policies since the SALSA always has the same threshold boundary to shift load requests. Compared to ECT-P2, which uses electricity prices as triggers, fluctuations in prices give no flexibility to the SALSA. The reason is that when electricity prices are low, it allows more load requests to operate. This is also reflected in ECR values, which show average Electricity Cost Reduction of households. Finally, CER studies the average percentage of CO_2 Emission Reduction. ECT-P3 helps consumers reduce more CO_2 emission. However, it fails to reach reasonable values in other criteria.

TABLE I: Performance analysis of the algorithm in step 1

	ECT-P1	ECT-P2	ECT-P3
AAS (kW)	24.94	30.11	5.27
PDR (%)	39.98	18.65	-8.05
ECR (%)	3.45	8.45	3.74
CER (%)	7.14	5.20	12.28

B. Step 2

This step studies the scalability of the SALSA algorithm in two sub-steps. Both sub-steps select ECT-P1 to investigate how SALSA behaves.

1) Sub-Step 1: This step increases the number of households from 100 to 500, as Table II shows the demand response evaluation results. AAS increases while the number of households increase. However, this is not applicable to PDR. The reason is having the similar increase in ECTs over time. In this case, PDR values do not follow a straight line. Increasing the number of households support them to averagely benefit more in terms of ECR and CER. It should be emphasized that AAS, ECR, and CER do not grow linearly due to having a randomness in load profile generation and flexibility values.

2) Sub-Step 2: This part aims to analyze the computation time of running the SALSA algorithm when the number of households increases from 500 to 10000. This causes the SALSA to receive and schedule much more load requests at each time interval. The event-driven load scheduling problem is NP-complete by a reduction from the "0-1" Knapsack problem [24]. However, to still use the dynamic programming technique to receive optimal scheduling solutions over time, it is needed to enlarge the time resolution. This results in a decrease in the number of referrals to the load request selection procedure in the SALSA algorithm (see Fig. 3). In the first step, the SALSA is called 1440 times due to generating the load profiles at one-minute time resolution. However, in this part, they are regenerated at one-hour time resolution, which results in calling the SALSA algorithm for only 24 times. Fig. 11 demonstrates the whole simulation time versus the number of households. When the number of households equals to 5500, Matlab needs more memory to buffer load requests. However, this can be mitigated by increasing the time resolution. For instance, if the time resolution increases to two-hour intervals, scheduling 5500 households takes 31 minutes. Therefore, there is a trade-off between the number of households, the time resolution, and the computation time.

TABLE II: Performance analysis of the SALSA algorithm towards increasing the number of households

	100	150	200	250	300	350	400	450	500
AAS (kW)	24.94	25.04	25.14	25.74	25.65	26.57	27.61	27.35	27.39
PDR (%)	39.98	40.01	42.15	40.71	41.30	41.33	41.92	41.45	42.87
ECR (%)	3.45	3.75	3.79	3.97	4.05	4.45	4.87	4.88	4.99
CER (%)	7.14	7.20	7.87	8.14	8.17	8.26	8.32	8.40	8.45

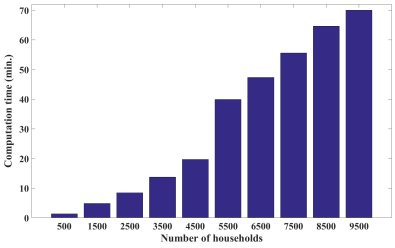


Fig. 11: Computation time versus number of households.

VI. Conclusions

Residential demand response is in a nascent stage in Europe. An only recently the first set of commercial aggregators have emerged. So far focus has been on the big consumer and very little attention has been paid to the residential consumers. However, to secure the future smart grid all types of flexibilities are needed. For residential consumers this means aggregating of a large number of flexible loads possible in the order of 100 thousands of households. This poses challenge to the ICT infrastructure that is needed to control the flexible loads. The paper has reported on the design of such infrastructure. Emphasis has been put on the scaling of the centralized aggregator step i.e., the back-end part of the infrastructure.

ACKNOWLEDGMENT

The work in this paper has received funding from the European Union Seventh Framework program (FP7/2007-2013) under grant agreement n° 619560 (SEMIAH). The authors would also like to thank Jung Min Kim for providing an initial survey of European demand response.

REFERENCES

[1] P. Capros, A. D. Vita, N. Tasios, D. Papadopoulos, P. Siskos, E. Apostolaki, M. Zampara, L. Paroussos, F. K., N. Kouvaritakis, L. Htiglundlsaksson, W. Winiwarter, P. Purohit, H. Bottcher, S. Frank, P. Havlik, M. Gusti, and H. P. Witzke, "Eu energy, transport and ghg emissions trends to 2050," Tech. Rep., 2013.

[2] "Energy Roadmap 2050," European Commission, Tech. Rep. COM(2011) 885 final, 15.12.2011 2011. [Online]. Available: https://ec.europa.eu/energy/en/topics/energy-strategy/2050-energy-strategy

[3] 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 2015 Euromicro Conference on Digital System Design (DSD), 2015, pp. 470–477.

[4] A. Soares, A. 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, 2014, pp. 490–503.

[5] J. Verhoosel, F.-J. Rumph, and M. Konsman, "Modeling of flexibility in electricity demand and supply for renewables integration," in Workshop on eeBuildings Data Models, Sophia Antipolis, France, 2011.

[6] A. Mohd, E. Ortjohann, A. Schmelter, N. Hamsic, and D. Morton, "Challenges in integrating distributed Energy storage systems into future smart grid," in 2008 IEEE International Symposium on Industrial Electronics, 2008, pp. 1627–1632.

[7] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, "A Survey on Demand Response Programs in Smart Grids: Pricing Methods and Optimization Algorithms," IEEE Communications Surveys & Tutorials, vol. 17, no. 1, 2015, pp. 152–178.

[8] S. Rahnama, J. D. Bendtsen, J. Stoustrup, and H. Rasmussen, "Robust Aggregator Design for Industrial Thermal Energy Storages in Smart Grid," pp. 1–15, 2015.

[9] M. Boehm, L. Dannecker, A. Doms, E. Dovgan, B. Filipic, U. Fischer, W. Lehner, T. B. Pedersen, Y. Pitarch, L. Siksnys, and T. Tusar, "Data Management in the MIRABEL Smart Grid System," in Proceedings of the 2012 Joint EDBT/ICDT Workshops, ser. EDBT-ICDT '12. New York, NY, USA: ACM, 2012, pp. 95–102.

[10] J. Hu, S. You, M. Lind, and J. stergaard, "Coordinated Charging of Electric Vehicles for Congestion Prevention in the Distribution Grid," IEEE Transactions on Smart Grid, vol. 5, no. 2, 2014, pp. 703–711.

[11] S. Rahnama, S. E. Shafiei, J. Stoustrup, H. Rasmussen, and J. Bendtsen, "Evaluation of Aggregators for Integration of Large-scale Consumers in Smart Grid," {IFAC} Proceedings Volumes, vol. 47, no. 3, 2014, p. 1879.

[12] P. Siano, "Demand Response and Smart Grids – A survey," Renewable and Sustainable Energy Reviews, vol. 30, no. 0, 2014, p. 461.

[13] A. D. Giorgio and L. Pimpinella, "An event driven smart home controller enabling consumer economic saving and automated demand side management," Applied Energy, vol. 96, 8 2012, pp. 92–103.

[14] P. Palensky and D. Dietrich, "Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads," Industrial Informatics, IEEE Transactions on, vol. 7, no. 3, 2011, pp. 381–388.

[15] Y. Wang, I. R. Pordanjani, and W. Xu, "An event-driven demand response scheme for power system security enhancement," IEEE Transactions on Smart Grid, vol. 2, no. 1, 2011, pp. 23–29.

[16] R. H. Jacobsen and S. A. Mikkelsen, "Infrastructure for Intelligent Automation Services in the Smart Grid," Wireless Personal Communications, vol. 76, no. 2, 2014, pp. 125–147. [Online]. Available: http://dx.doi.org/10.1007/s11277-014-1682-6

[17] T. Mancini, F. Mari, I. Melatti, I. Salvo, E. Tronci, J. K. Gruber, B. Hayes, M. Prodanovic, and L. Elmegaard, "Demand-Aware Price Policy Synthesis and Verification Services for Smart Grids," in 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), 2014, pp. 794–799.

[18] A. Backers, F. Bliek, M. Broekmans, C. Groosman, H. de Heer, M. van der Laan, M. de Koning, J. Nijtmans, P. Nuygen, T. Sanberg, B. Staring, M. Volkerts, and E. Woittiez, "An introduction to the Universal Smart Energy Framework," Tech. Rep. USEF2014:I.II, 2014.

[19] M. T. Beyerle, J. A. Broniak, J. M. Brian, and D. C. Bingham, "Manage whole home appliances/loads to a peak energy consumption," March 2011, uS Patent App. 13/042,550.

[20] J. Ringelstein, M. Vogt, and B. Klebow, "An energy management gateway for facilities based on open middleware," in 2013 IEEE International Workshop on Intelligent Energy Systems (IWIES), 2013, pp. 46–51.

[21] "Energy management system application program interface (EMS-API) - Part 301: Common information model (CIM) base," International Electrotechnical Commission (IEC), Standard IEC61970-301:2013, 2013.

[22] "Application integration at electric utilities - System interfaces for distribution management - Part 11: Common information model (CIM) extensions for distribution," International Electrotechnical Commission (IEC);, Standard IEC 61968-11:2013, 2013-03-06 2013.

[23] G. Basso, P. Ferrez, D. Gabioud, and P. Roduit, "An Extensible Simulator for Dynamic Control of Residential Area: Case Study on Heating Control," in 2015 Euromicro Conference on Digital System Design (DSD), 2015, pp. 486–493.

[24] A. G. Azar and R. H. Jacobsen, "Appliance Scheduling Optimization for Demand Response," International Journal on Advances in Intelligent Systems, vol. 9, no. 1&2, 2016.

Paper F

A Formal Framework for Modeling Smart Grid Applications: Demand Response Case Study

The paper presented in this chapter has been accepted as a conference publication [7].

[7] Armin Ghasem Azar, Emad Samuel Malki Ebeid, and Rune Hylsberg Jacobsen, "A Formal Framework for Modeling Smart Grid Applications: Demand Response Case Study," In *IEEE Euromicro Conference on Digital System Design (DSD)*, 2016, pages 46-54, doi: 10.1109/DSD.2016.61

Authors contribution: Armin Ghasem Azar conceived the initial design of the formal smart grid framework; Armin Ghasem Azar carried out the literature survey in selected areas and formulated the system model; Emad Samuel Malki Ebeid contributed to the design of UML diagrams and profiles; All the authors discussed the findings; Rune Hylsberg Jacobsen provided support on the system performance analysis; Armin Ghasem Azar and Emad Samuel Malki Ebeid were the main editors of the manuscript; Rune Hylsberg Jacobsen contributed to the reviewing process.

2016 Euromicro Conference on Digital System Design

A Formal Framework for Modeling Smart Grid Applications: Demand Response Case Study

Armin Ghasem Azar, Emad Ebeid, and Rune Hylsberg Jacobsen
Department of Engineering, Aarhus University, Denmark
{aga, esme, rhj}@eng.au.dk

***Abstract*—Smart grid applications belong to a diverse set of technologies, which combine behaviors and actions of all grid actors. Building such application is a challenging task requiring modeling, integrating, and validating different grid aspects efficiently. The design flow should adapt to the grid's requirements ranging from basic appliance functions to complex load scheduling algorithms and their integration's consequences. This paper tackles such challenges by proposing a framework to describe smart grid elements formally along with defining their interactions. The framework provides a smooth way for upgrading grid components independently and evaluate their performance. A case study utilizing an infrastructure of electric vehicles is demonstrated to validate the framework and prove its applicability in modeling smart grid applications. The case study enables a demand response service that provides a solution to the coordinated charging scheduling problem of electric vehicles.**

Keywords*—Smart grid, formal modeling, validation, UML profile, demand response, load scheduling.

I. Introduction

The current structure of the electrical grid is ineffective in responding to the growing demand for electricity. National Institute of Standards and Technology (NIST) has developed a smart grid conceptual model, which identifies main smart grid domains and stakeholders describing their possible communication paths [1]. Domains are customers, markets, service providers, operations, bulk generation, transmission, and distribution. Standardizing and formalizing these domains in the smart grid is a challenging procedure, particularly when interdisciplinary engineering approaches are concerned. International Electrotechnical Commission (IEC) has proposed reliable and reproducible standards for the smart grid, e.g., IEC 61970, which is based on Common Information Model (CIM) providing various elements and definitions of energy management systems in the smart grid [2]. Smart Grid Architecture Model (SGAM) represents a technical reference architecture to compare different architectures and identify gaps in the smart grid standardization. It enriches the NIST's model by integrating *interoperability layers* and *zones* with the domains already defined [3].

Nevertheless, far too little attention has been paid by smart grid application designers to build a formal pragmatic framework. To effectively design such frameworks, the following requirements are crucial: 1) the grid infrastructure must be scalable and specified together with its constituents, 2) the grid model must be interoperable and supportive to its future extensions, 3) the grid design must consider its topology and hardware/software/network aspects, and 4) the grid updating process must perform independently of its aspects. This paper proposes a formal framework to model smart grid applications, as Fig. 1 presents its top-down design methodology. This strategy helps the framework build reliable and self-healing smart grid applications. Moreover, it is entirely adaptable to various domains, their actors, and interior entities.

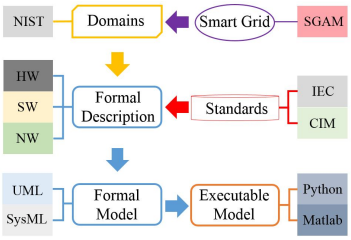


Fig. 1. Top-down design methodology of the proposed framework

First, the framework formally describes each domain and its actors using software engineering design principles to ensure its scalability and interoperability. The framework is based on three main aspects: hardware (HW), software (SW), and network (NW). The hardware aspect discusses the physical entities of the smart grid actors. The software aspect represents some functionalities that are running inside these hardware entities. Meanwhile, the network aspect establishes the communication flow between the grid's actors. Next, the framework exploits generic modeling languages such as Unified Modeling Language (UML) for formalizing its aspects. This step enables the interoperability, reusability, and scalability of smart grid models. Finally, these models are transformed into executable models, i.e., following the Model-to-Text transformation approach, such as Matlab.

This paper analyzes *customers* and *operations* domains in the smart grid. The framework is indeed open to being extended by other domains. Customers possess a set of electricity consuming devices. They aim to manage and control their consumption behaviors. Operators continuously stabilize grid functions to ease the power system operation using various management systems.

A Demand Response (DR) scenario is employed as a case study to validate the usability and applicability of the proposed formal framework. DR is a cooperation between customers and grid operators through doing incentive-based businesses to reduce/stabilize the customers' aggregated load consumption at peak periods [4]. The importance of synthesizing the formal framework and DR is demonstrated by challenging the coor-

978-1-5090-2817-7/16 $31.00 © 2016 IEEE
DOI 10.1109/DSD.2016.61

dination of a high penetration of Electric Vehicles (EVs) [5]. It focuses on peak shaving and customers' inconvenience reduction considering physical grid stability constraints and EVs' arrival and departure times [6][7].

As a result, this paper makes the following contributions:

- Three essential grid *aspects*, i.e., hardware, software, and network, are presented to demonstrate the framework's scalability, reusability, interoperability, and updatability focusing on both power and Information and Communication Technology (ICT) perspectives;
- A new UML *profile* is developed to capture the framework aspects inspiring by diverse meta-classes;
- A robust formal framework is proposed to mathematically formalize different domain elements;
- A novel UML *SmartGrid* profile is produced to efficiently model various smart grid applications;
- The applicability of the formal framework is validated using a case study that challenges the charging scheduling problem of EVs using DR as an application of the formal smart grid framework.

The paper is organized as follows: Section II overviews the related work. Section III provides the prerequisite aspects of the framework. Section IV proposes the formal framework. Section V studies the framework's applicability using a case study. Finally, Section VI concludes the paper and provides possible future extensions.

II. Related Work

In recent years, there has been an increasing interest in investigating concerns about the inefficient structure of the current electrical grid for responding the growing electricity demand [8][9]. Farhangi [8] examined different impacts of transforming the current electrical grid to a complex system of systems named the smart grid. Fang *et al.* [9] discussed that engaging ICT with the smart grid could play a major role in enabling technologies for smart grid data communications.

To handle these data communications, Godfrey *et al.* [10] presented a co-simulation framework to model both the communication network and the power system. They employed a baseline scenario and demonstrated the responses to power fluctuations subject to considering any communication efficiency (e.g., Quality of Service (QoS)). Afterwards, Schutte *et al.* [11] delivered the Mosaik framework for a co-simulation of various scenario descriptions, grid topology, and control strategies using their semantic information. The framework lacks precise details in the model description since it has just considered a single naive life entity while current electrical infrastructure includes different varieties of entities such as cables, step-down transformers, etc. Finally, Montenegro *et al.* [12] presented an Open Distribution System Simulator (OpenDSS) for the smart grid. It is a simulation tool for the electrical power system principally for the electricity distribution grid. However, it failed in calculating mathematical models to develop a real-time simulation for available devices in the power grid to show their real behavior and their communications.

To the best of the authors' knowledge, the model-based design of the smart grid as a robust formal framework is currently limited and not well supported. Andrén *et al.* [13] also recognized this issue and proposed a semantically-driven design method using CIM for transmission (IEC 61970-301) and distribution (IEC 61968-11). These standards have been highly promoted for modeling grid issues and the corresponding device/component communications. Additionally, since the smart grid requires a specific standard for communication networks and power utility automation systems, IEC 61850 has been launched in the course of an object-oriented information model. Nonetheless, an ICT-driven formal framework is needed to overcome the major shortcoming of the standards above, i.e., a limited number of covered domains beside discarding grid physics, communication and control issues.

SGAM intended to present the design of smart grid use cases in an architectural way [3]. To handle the model, it introduced five interoperability layers to allow the representation of entities and their relationships in the context of smart grid domains. This paper maps the *component*, *information*, and *communication* layers into *hardware*, *software*, and *network* aspects of the framework, respectively, to build a more generic profile. Although SGAM provides the structural design of the smart grid applications in a high-level approach, however, it lacks the behavioral part describing feasible actions and behaviors of each actor. This paper covers this gap by mathematically formulating actors and defining their proper behavioral actions.

UML is widely applied to software modeling and the demonstration of its specifications based on hardware/software/network co-designs [14][15][16]. De Miguel *et al.* [14] introduced UML extensions for the representation of temporal requirements and resource usage of real-time systems. Their tools generated a model for the OPNET simulator. Hennig *et al.* [15] described a UML-based simulation framework for performance assessment of hardware/software systems described as sequence diagrams. The proposed simulator was based on the discrete event simulation package OMNet++. Finally, Ebeid *et al.* [16] proposed a formal framework and supporting tools to represent networked-embedded systems in terms of application requirements, the library of network components, the environment description, and rules to compose them. Their framework allowed to generate code for design validation by simulation and return annotation mechanism of the simulation results to refine the original model.

III. Aspects and Modeling

This section sets the basis for building a comprehensive formal framework to model, simulate, and validate smart grid applications. This paper establishes the proposed framework according to the "separation of concerns" design principle. It allows the reusability, development, and upgrading of its components independently. Therefore, the framework orchestrates the smart grid system using three most important aspects; *hardware*, *software*, and *network*.

A. Aspects

Fig. 2 demonstrates an overview of the framework presenting these aspects. The framework is formalized using grid component definitions followed in architectural guidelines

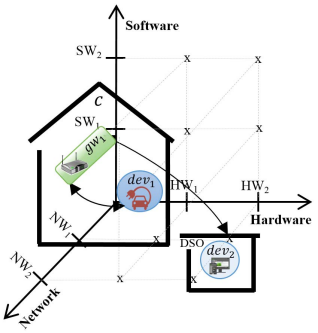


Fig. 2. Overview of the hardware, software, and network aspects

of the future smart grid such as IEC standards [2][13] and SGAM [3]. They organize how application characteristics exchange data model specifications among described aspects. The following parts explain how these aspects are formalized.

1) Hardware Aspect: The smart grid comprises different hardware *devices* employed in various domains. Devices can be digital, analog, or heterogeneous with discrete or continuous behavior. They are characterized based on different structures and responsibilities. The *hardware* aspect defines the physical model of these devices. To understand the general constituents of a device helps the formal framework allocate appropriate responsibilities to each different device from both computational and communicational points of view. For instance, a home energy management system can perform optimization techniques and message broadcasting operations with certain specifications. Equation (1) defines the device entity:

$$\begin{aligned} &dev = [dig, ana] \in \mathbb{D} \text{ where:} \\ &dig = [comp, comm], \\ &ana = [phy, mech]. \end{aligned} \tag{1}$$

Each device *dev* is a member of a multiset of devices $\mathbb{D}$. It includes digital *dig* and analog *ana* components. The latter component will be investigated more in the future extensions of the framework. A digital component consists of computational *comp* and communication *comm* components. The computational component, e.g., a CPU, performs computing operations. The communication component, e.g., a network interface, distributes the achieved/updated information to other devices (using the network aspect). Equation (2) defines the computational component:

$$\begin{aligned} &comp = [r, q] \text{ where:} \\ &r = [f_1, f_2, \ldots, f_n] \in app, \\ &q = [o_1, o_2, \ldots, o_m] \in \mathbb{R}_0^+. \end{aligned} \tag{2}$$

The computational component *comm* includes computation r and overhead q vectors. The former includes n function elements. Each element performs a specific procedure, for instance, starting/stopping the appliance operating cycle. This is done as a software application *app* running in a hardware entity, e.g., CPU. The latter contains m overhead elements. Each represents the processing time of running a subset of function elements. The information, processing in the computational component, should be distributed to other devices through the communication component using the network aspect *nw*. Equation (3) defines this component:

$$\begin{aligned} &comm = [i, z] \text{ where:} \\ &i = [elec, info] \in nw, \\ &z = [e_1, e_2, \ldots, e_v] \in \mathbb{R}_0^+. \end{aligned} \tag{3}$$

The communicational component *comm* includes vectors of communication interfaces i and communication overheads z. The vector of communication interfaces i, as a member of a network component *nw*, includes electricity *elec* and information *info* elements. The former is responsible for satisfying the electricity demand of the device through physical power lines. The latter is performed on top of a communication protocol to exchange the information with other devices. The vector of communication overheads z comprises v communication overhead elements. Each element is caused by characteristics of the communication media of each corresponding device. Packet loss rate, latency, and throughput are some examples of communication overheads.

2) Software Aspect: The *software* aspect aims to develop platform-independent models of software *applications* that can be executable in *hardware* entities. Software applications can describe miscellaneous functionalities, ranging from the basic operation of an appliance (e.g., ON/OFF) to heterogeneous communication protocols (e.g., the Smart Energy Profile 2.0 (SEP2) [17]). The framework considers the software development from the object-oriented programming point of view. Equation (4) defines the software application:

$$\begin{aligned} &app = [sv, bv] \in Apps \text{ where:} \\ &sv = [en, re] \in SV, \\ &bv = [s, \iota] \in BV. \end{aligned} \tag{4}$$

Each application *app* is a member of a multiset of applications *Apps*. It consists of two correlated structural *sv* and behavioral *bv* views. The former describes the structure of the application while the latter presents its dynamics. The structural view *sv* is a member of a multiset of structural views *SV*. It is a combination of entities *en* and relationships *re*. An entity represents the functional part of an application, which can be periodic or aperiodic. A relationship defines a logical connection between entities. Afterwards, the behavioral view *bv* is a member of a multiset of behavioral views *BV*. It represents the dynamical view 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 ι provides conditional paths between them. The detailed description of EFSM can be found in [18].

3) Network Aspect: The *network* aspect enables the communication between two or more devices based on CIM in IEC 61970 standards series [2]. It defines the communication media that delivers the information broadcast between *software* and *hardware* aspects. Its characteristics shape its performance,

for instance, the latency, throughput, and packet loss. Equation (5) formulates this aspect:

$$\begin{aligned} &nw = [QoS, dist, mob] \in NW \text{ where:} \\ &QoS = [x_1, x_2, \ldots, x_y] \in \mathbb{R}_0^+, \\ &dist \in \mathbb{R}_0^+, mob \in \mathbb{B}. \end{aligned} \tag{5}$$

Each network component *nw* is a member of a multiset of network components *NW*. The *QoS* vector includes y elements of parameters x to represent the overall performance of the communication network. The distance *dist* indicates the topological distance value between two connected device entities *dev*. Finally, mobility *mob* represents the device's mobility type, i.e., wired or wireless (see Equation (1)).

B. Modeling

This section shows the trajection of the previously described mathematical equations of the framework's aspects into a formal model. Fig. 3 illustrates a novel UML profile, in which aspects are labels with red-dashed rounded rectangles.

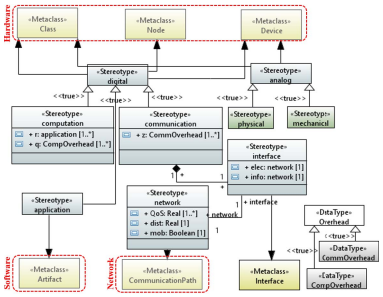


Fig. 3. UML profile diagram of the framework's aspects

The used UML meta-classes are:

1) **Class**: is an extensible template for creating objects, providing initial values for attributes and implementations of the behaviors. It is used to model the `Digital` and `Analog` hardware aspects of the smart grid applications.
2) **Node**: is a physical object that represents a computational resource of the system, such as servers. It is used to model the `Digital` hardware aspect of the smart grid applications.
3) **Device**: is a type of node that represents a physical computational resource, such as a gateway. The device is used to model the `Digital` and `Analog` hardware aspects of the smart grid applications.
4) **Artifact**: is a software component, such as an executable software component, files or libraries, deployed inside the **Node**. It is used to model the software `Application` aspect.
5) **CommunicationPath**: defines the path between two nodes that are able to exchange signals and messages such as a wired/wireless communication channel. It is used to model the `Network` aspect of the smart grid applications.
6) **Interface**: is a collection of operation signature and attribute definitions that ideally defines a cohesive set of behaviors. It is used to model the `Network` aspect of the smart grid applications in the sense of connections between devices and the network.

The profile will be used to enrich the semantics of the UML diagrams in representing smart grid applications, which the following section will emphasize this step.

IV. Proposed Smart Grid Framework

This section first, describes the formal framework proposed based on the aspects defined in Section III. Then, it models the framework mathematically using a UML smart grid class diagram. The smart grid formalization requires a high-level conceptual framework taking its domains into account. This paper serves the formal framework as an application. It intends to both identify the actors inside each smart grid domain and establish their possible communication routes appropriately. Equation (6) defines a smart grid:

$$\begin{aligned} &sg = [w] \in \Psi \text{ where:} \\ &w \subseteq \omega. \end{aligned} \tag{6}$$

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 significant challenge about these domains is how to organize them to work consistently focusing on delivering right services to their relevant interior actors. As the concept of the separation of concerns is employed, each domain corresponds to an *add-in* feature to the framework. Hence, adding or removing a domain will not affect the framework's functionality, which strengthens its robustness and flexibility. The following parts present how smart grid domains are formalized and modeled.

A. Formalization: Smart Grid Domains

This paper considers $w = [\mathbb{C}, \mathbb{O}]$ as the first step in formalizing a smart grid, where $\mathbb{C}$ and $\mathbb{O}$ correspond to *Customers* and *Operations* domains, respectively [1].

1) Customers Domain: The Customers domain typically provides applications to customers to manage their electricity consumption behaviors. Equation (7) defines this domain:

$$\begin{aligned} &\mathbb{C} = [c_1, c_2, \ldots, c_h] \text{ where:} \\ &c = [SA, ems, sm, gw], \\ &SA \subseteq \mathbb{A}, ems \in app, \{sm, gw\} \in dev. \end{aligned} \tag{7}$$

This domain includes h customers. 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 to enhance 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 running on a hardware device. Moreover, each customer has a smart meter *sm* installed in the dwelling. The smart meter measures the electricity consumption of smart appliances and periodically sends them to the electric 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.

Smart Appliances: They are the main drivers of electricity demands, as Equation (8) defines:

$$\begin{aligned} &SA = [sa_1, sa_2, \ldots, sa_p] \text{ where:} \\ &sa = [sf, ct_{sa}, lp], \\ &sf = [shift, intr] \in \mathbb{B}, \\ &ct_{sa} = [\zeta_1, \zeta_2, \ldots, \zeta_a] \in \mathbb{R}_0^+, \\ &lp = [ec, \Delta\tau], \\ &ec \in \mathbb{R}_0^+, \Delta\tau \in \mathbb{R}^+. \end{aligned} \tag{8}$$

Each customer c possesses p smart appliances. Each smart appliance $sa \in dev$ has a smart feature pair *sf* including two dependent Boolean functions 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. Dependency between these features indicates that if a smart appliance is shiftable, it can be either interruptible, i.e., $sf = [\text{TRUE}, \text{TRUE}]$ or uninterruptible, i.e., $sf = [\text{TRUE}, \text{FALSE}]$. Nevertheless, if it is non-shiftable, then, it is also uninterruptible, i.e., $sf = [\text{FALSE}, \text{FALSE}]$. In addition, each smart appliance has a set of constraints ct_{sa} including a constraint elements ζ. For instance, each appliance should finish its operation cycle in the defined period. Finally, the smart appliance *sa* follows a specific load profile *lp* in each operating cycle. It is determined with respect to its program predefined by the corresponding customer. Each *lp* is presented as a vector of time-series electricity consumptions *ec* with a specific time resolution $\Delta\tau$.

Energy Management System: It is a software application *app* running on a device *dev*. It can be a web service in the cloud or an application installed on a server. In fact, *ems* is a combination of energy optimization and information processing functions. It integrates the efficiency into advanced control and optimization strategies. Equation (9) defines the $ems \in app$:

$$\begin{aligned} &ems = [obj_{\mathbb{C}}, pref^p, ev^p, rsp^p] \text{ where:} \\ &obj_{\mathbb{C}} = [\sigma_1, \sigma_2, \ldots, \sigma_t] \in \mathbb{R}_0^+, \\ &pref = [ost, opr, ofl] \in \mathbb{R}_0^+, \\ &ev = [est, ept, obj_{\mathbb{C}}, pref, sf, ct_{sa}, lp], \\ &\{est, ept\} \in \mathbb{R}_0^+, \\ &rsp = [dec, rst], \\ &dec \in \mathbb{B}, rst \in \mathbb{R}_0^+. \end{aligned} \tag{9}$$

Each customer, using an *ems*, adjusts his/her own objective set $obj_{\mathbb{C}}$ including t distinct objectives. These objectives can be in conflict with or in line with each other, for instance, minimizing the electricity cost and CO_2 emission, maximizing comfort level, minimizing appliance service delay, etc. In addition, for each smart appliance *sa* in *SA*, the customer provides a vector of operating preferences *pref*. For the sake of simplicity, *pref* is assumed as a 3-tuple including customer's interested operating start time *ost*, operating program *opr*, and operating flexibility *ofl*. As the first element, customers adjust the time, at which they want to operate their smart appliance. As the second element, customers set a specific program to operate each smart appliance (e.g., washing the clothes at 60 °C). This program directly influences the load profile *lp* mentioned before. As the last element, customers offer a voluntarily flexibility *ofl* to operate each shiftable smart appliance. Two flexibility types, named *deadline* and *temperature*, are defined. Deadline flexibility is an additional time to the required period of the main operating cycle of physically-controllable smart appliances. Providing this flexibility, corresponding smart appliances can be shifted and interrupted until reaching the adjusted deadline flexibility. Temperature flexibility is a feature of thermostatically-controllable smart appliances, e.g., Heating, Ventilating, and Air Conditioning (HVAC). It is notable that *ost* for these smart appliances equals to the operating set temperature. Similarly, offering the temperature flexibility, the dwelling's temperature can fluctuate over the operating set temperature.

Once customers set the objectives and preferences, the EMS sends events *ev* to the Operation Management System (OMS) of the Operations domain (described later). Then, the EMS waits to receive responses *rsp* of similarly sent events. Notation *est* refers to the time, at which the event has been sent. The event pooling time *ept* defines a length of time, at which each smart appliance waits to receive a response from the EMS after sending the event. If no response arrives, another event is forwarded after *ept* minutes/seconds. 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 actuating smart appliances in accordance with the received responses.

2) Operations Domain: This domain handles the movement of electricity. It facilitates the continuous grid management functions. Its responsibilities include maintaining and operating the electricity distribution infrastructure efficiently while delivering the electricity to customers securely. Equation (10) defines this domain:

$$\begin{aligned} &\mathbb{O} = [oms, nms] \text{ where:} \\ &\{oms, nms\} \in app. \end{aligned} \tag{10}$$

This domain includes two representative software applications Operation Management System (OMS) and Network Management System (NMS) derived from IEC 61970 [2]. The former, as an operational planning software, is responsible for performance monitoring and optimization of the electrical grid, e.g., load balancing and scheduling. The latter, as a network maintenance software, monitors the communication network of smart grid domains, for instance, fault management, overhead and delay calculation, etc. (beyond the scope of this paper).

Operation Management System: Equation (11) defines *oms*:

$$\begin{aligned} &oms = [obj_{\mathbb{O}}, ct_{dso}, ep, drs] \text{ where:} \\ &obj_{\mathbb{O}} = [\varsigma_1, \varsigma_2, \ldots, \varsigma_g] \in \mathbb{R}_0^+, \\ &ct_{dso} = [\chi_1, \chi_2, \ldots, \chi_l] \in \mathbb{R}_0^+, \\ &ep = [\rho_1, \rho_2, \ldots, \rho_d] \in \mathbb{R}_0^+, \\ &drs = [buf, sch, rsp^p], \\ &buf = [iw, is, do, dw] \in \mathbb{R}_0^+, \\ &sch = [obj_{\mathbb{O}}, ct_{dso}, ep, ev^p, \lambda], \\ &\lambda \in \mathbb{R}^+. \end{aligned} \tag{11}$$

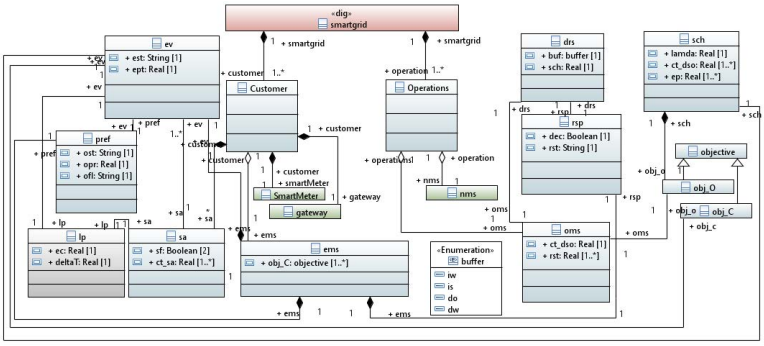


Fig. 4. UML class diagram of the framework's elements

Here, Distribution System Operators (DSOs) are the 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 electricity distribution. Each DSO, similar to customers (see Equation (9)), has a set of objectives $obj_{\mathbb{O}}$ including g distinguishable objectives ς. Flattening the aggregated electricity consumption, reducing the outages, reducing the CO_2 emission, etc. are some examples of objectives. 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 electricity consumption thresholds in the feeder/substation level, active and reactive power flow capacities, etc. are some of such examples. The DSO adjusts the set of electricity prices *ep* over d time periods in different schemes, i.e., real-time, day-ahead, etc. Finally, the DSO employs a Demand Response System (DRS) to respond to the events received from the EMSs of customers.

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 relocate them to different buffers. The scheduler *sch* follows a scheduling approach λ to make these decisions. Decisions are made based on the information stored in the events. The scheduling approach can be either stochastic or deterministic applying single-objective or multi-objective optimization techniques.

B. Modeling: UML Class diagram

Fig. 4 shows a complete description of the proposed framework as UML class diagram combined with the profile diagram (see Fig. 3) The `smartgrid` class is extended by the `<<digital>>` stereotype that contains the predefined framework's aspects (analog part is not considered at this stage). Objects can be instantiated and linked together to compose a variety of smart grid applications. Next section exemplifies such application as a case study.

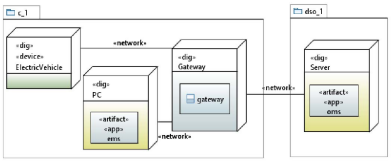


Fig. 5. UML deployment diagram of the instantiated objects

V. Case Study: Charging Scheduling Coordination of Electric Vehicles

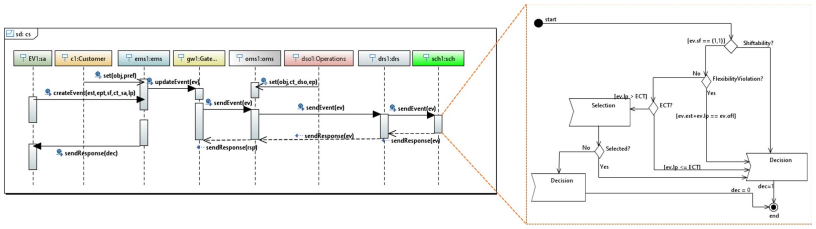
This section presents a case study to demonstrate the applicability of the proposed formal framework.

A. Description

In the recent years, the electric transportation system has gained the largest share of total energy consumption growth in the world [5]. DR is one of the promising solutions to meet these high-consuming electricity demands at peak periods [4]. Therefore, designing a smart grid application for coordinated charging process of EVs in residential DR is substantial. The case study aims to model $h = 100$ customers that are connected to a utility company, which handles their electric demands. The company uses a direct load control approach, including a load scheduling algorithm. This algorithm intends to shave demands at peak periods.

B. Modeling

The structure of one customer of the case study is captured in Fig. 5 using a UML deployment diagram along with the developed profile annotations (see Section III-B). The diagram shows a customer, who has an EV and communicates with



(a) (b)

Fig. 6. (a) UML sequence diagram of the interactions of the instantiated objects, (b) UML activity diagram of the scheduling algorithm

the DSO though a gateway that handles its preferences via the EMS system. The DSO contains a server that has an OMS, which runs a DR scheduling algorithm. The interactions between the elements of the case study are presented in Fig. 6 (for one customer). Fig. 6(a) shows the UML sequence and activity diagram. Fig. 6(b) demonstrates the behavioral part of the DR scheduler elaborating how the scheduler decides about the incoming events.

C. Code Generation

UML diagrams have been manually mapped into Matlab code to validate these high level models. Table I shows the main mapping patterns of the UML diagrams and the generated executable Matlab code. However, in order to build an automatic synthesis tool, the developer needs to use a model of computation to capture the UML models and generate the corresponding code [18]. In order to evaluate the performance of the scheduler, 100 customer objects have been instantiated in the Matlab code (see c_1 box in Fig. 5). It is assumed that these customers are connected to one substation.

D. Simulation Results

Arrival and departure times of EVs follow a normal distribution with $\mathcal{N} = (19, 10)$ and $\mathcal{N} = (7.5, 1)$, respectively, when state of charges of EVs' batteries at that times are 0 and 100, respectively [7]. The DSO attempts to keep the aggregated load consumption below an Electricity Consumption Threshold (ECT) over time. ECTs are adapted to the normalized Danish real-time electricity prices *ep* [19]. The DRS, inside the OMS, couples ECTs with a load scheduler. This scheduling algorithm uses a deterministic event-driven DR approach, which applies a selection procedure on incoming events to select some at each time interval and postpone the remaining to the future. This procedures is a single-objective optimization problem, which is continuously solved by a dynamic programming technique at each time interval. For more information, the reader is referred to [6].

Fig. 7 demonstrates the charging process aggregation of 100 EVs in two consecutive days before and after running the scheduling. ECTs follow the multiplication of the normalized vector of electricity prices by the current aggregation of load requests at each time interval. The scheduler receives the

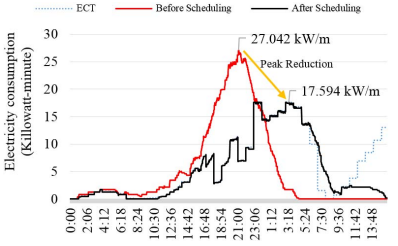


Fig. 7. Charging process aggregation of 100 EVs before and after running the scheduling during two-consecutive days

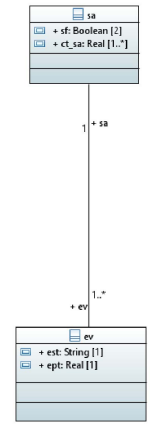
events in one-minute time resolution ($\Delta\tau = 1$). At each time interval, it runs the scheduling algorithm to allow a subset of EVs to operate and postpone the remaining subset to the next time interval. In a future time interval, the scheduler receives the previously postponed events along with the events sent for the first time from new EVs. It is worth noting that in the worst case, the simulation takes 0.38 seconds to buffer the incoming events, schedule them, and forward the responses back.

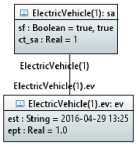
The peak consumption before scheduling occurs at 21:05 with 27.04 Kilowatt-minute of electricity consumption. Running the scheduling shifts the peak to 03:05 (the next day) with 17.59 Kilowatt-minute of electricity consumption. This confirms that the DSO succeeds in shaving the peak by 34.94%. This peak reduction uses averagely 60.29% of customers' flexibilities, which causes the customers to wait averagely 356 minutes more to have their EV fully charged. Intuitively, this waiting time falls in the flexibility period of each customer (starting from the EV's arrival time ending to its departure time).

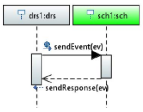
Fig. 8 shows the charging process of the EV of customer 52. Here, the EV has arrived at 20:05 in the present day and will depart at 08:22 in the next day. If the battery takes 300 minutes to be charged, the customer can receive the fully charged EV at 01:05 (the next day). The period from

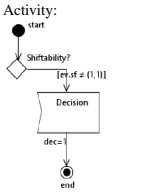
TABLE I. MAPPING PATTERNS FROM UML DIAGRAMS TO MATLAB CODE

UML entity	Matlab code

Classes: 	`classdef sa`
	`  properties`
	`  sf`
	`  ct_sa`
	`  ev`
	`  end`
	`  methods`
	`  function obj=sa(sf,`
	`  ↪ ct_sa)`
	`  obj.sf = sf;`
	`  obj.ct_sa =`
	`  ↪ ct_sa;`
	`  end`
	`  function obj =`
	`  ↪ CreateEvent(obj)`
	`  est = datetime;`
	`  ept = 2;`
	`  obj.ev = Event(`
	`  ↪ est,ept);`
	`  end`
	`  end`
	`end`
	`classdef Event`
	`  properties`
	`  est`
	`  ept`
	`  end`
	`  methods`
	`  function obj=Event(`
	`  ↪ est,ept)`
	`  obj.ept = ept;`
	`  obj.est = est;`
	`  end`
	`  end`
	`end`

Objects: 	`sf = [1,1];`
	`ct_sa = 1;`
	`ElectricVehicle(1) = sa(sf,`
	`↪ ct_sa);`
	`ElectricVehicle(1) =`
	`↪ CreateEvent(`
	`↪ ElectricVehicle(1));`

Sequence: 	`sendEvent(ev);`
	`iw = {iw,ev};`
	`Scheduler(iw);`
	`sendResponse(ev);`

Activity: 	`if ev.sf ~= [1,1];`
	`  dec = 1;`
	`end`

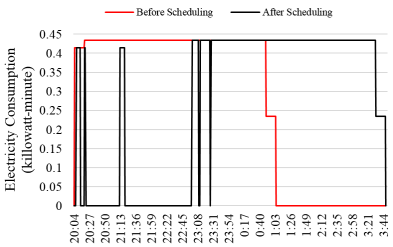


Fig. 8. The charging process of the EV of customer 52

this time until the departure time contributes 438 minutes of flexibility. According to the simulation, the scheduler delivers the EV at 03:47 in the fully charged status. This indicates that the scheduler has used 37.21% of the provided flexibility, which equals to 163 minutes of delay.

Fig. 9 demonstrates the effect of increasing the number of customers on the computation time in the worst case. The selection procedure in the scheduling algorithm is the main driver of the computation time. When $h = 1$, obviously, no selection procedure is executed during the simulation. While the number of customers increases, the number of referrals to the selection procedure during the simulation also increases. The ascending slope of the computation time is a consequence of having more incoming events at each time interval. This indeed makes the scheduling algorithm take longer time to return the decisions. However, the computation time does not increase exponentially, when the number of customers increases.

VI. CONCLUSION AND FUTURE WORK

This paper proposed a formal framework for modeling smart grid applications. It defined the main grid elements using three aspects: hardware, software, and network. A UML profile was developed to integrate these aspects into a generic profile. Employing this profile, a formal framework for modeling the main semantics of smart grid systems was defined and mathematically formalized with an emphasis on the customers and operations domains. A novel UML profile and a class diagram were developed to support the implementation of the framework, reflect the mathematical formulas, and create formal grid models. A case study was used to prove the validity of the formal framework demonstrating how to synthesize the formal framework into an executable code to build an efficient

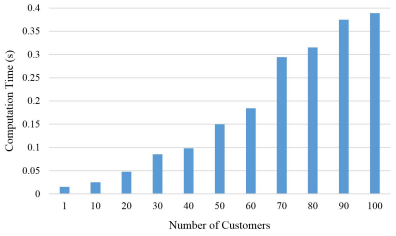


Fig. 9. Computation time in the worst case versus increasing the number of customers

simulator for grid applications. Preliminary results confirmed the significant benefit of orchestrating the framework as a demand response service for solving the coordinated charging scheduling problem of electric vehicles in the smart grid.

As on-going future extensions, various case studies will be investigated to confirm the generality and re-usability of the framework in different directions. Furthermore, smart grid communication protocols will be integrated into the framework (updating the network aspect) to evaluate how efficient the framework behaves in a simulated ICT-based power grid environment. At the moment, the code is not automatically generated by the framework, but manually mapped from the UML model. This will be updated to an automatic code generation, using e.g., Acceleo, which is an open-source code generator. It allows people to use a model-driven approach to build applications. Finally, an important element to define a smart grid application is the structure of the grid's topology. The future version of the framework will be explicitly adaptable to the current top-down topology.

ACKNOWLEDGMENT

The work in this document has been funded by European Union Seventh Framework program (FP7/2007-2013) under grant agreement n° 619560 (SEMIAH).

REFERENCES

[1] N. Framework, "Roadmap for Smart Grid Interoperability Standards," *NIST special publication*, vol. 1108, 2010.

[2] "IEC 61970: Energy Management System Application Program Interface (EMS-API) - Part 301: Common Information Model (CIM) base, Third Edition," Tech. Rep., 2009.

[3] Smart Grid Coordination Group, "Reference Architecture for the Smart Grid," Technical Report, 2012.

[4] P. Palensky and D. Dietrich, "Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 381–388, 2011.

[5] Z. Xu, W. Su, Z. Hu, Y. Song, and H. Zhang, "A Hierarchical Framework for Coordinated Charging of Plug-in Electric Vehicles in China," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 428–438, 2016.

[6] A. G. Azar and R. H. Jacobsen, "Appliance Scheduling Optimization for Demand Response," *International Journal on Advances in Intelligent Systems*, vol. 9, no. 1&2, 2016.

[7] F. Rassaei, W.-S. Soh, and K.-C. Chua, "Demand Response for Residential Electric Vehicles With Random Usage Patterns in Smart Grids," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1367–1376, 2015.

[8] H. Farhangi, "The Path of the Smart Grid," *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, 2010.

[9] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid-The New and Improved Power Grid: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944–980, 2012.

[10] T. Godfrey, S. Mullen, R. C. Dugan, C. Rodine, D. W. Griffith, and N. Golmie, "Modeling Smart Grid Applications With Co-Simulation," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2010, pp. 291–296.

[11] S. Schutte, S. Scherfke, and M. Troschel, "Mosaik: A Framework for Modular Simulation of Active Components in Smart Grids," in *IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS)*, Oct. 2011, pp. 50–60.

[12] D. Montenegro, M. Hernandez, and G. Ramos, "Real Time OpenDSS Framework for Distribution Systems Simulation and Analysis," in *Transmission and Distribution: Latin America Conference and Exposition (T D-LA), 2012 Sixth IEEE/PES*, Sept 2012, pp. 1–5.

[13] F. Andrén, M. Stifter, and T. Strasser, "Towards a Semantic Driven Framework for Smart Grid Applications: Model-Driven Development Using CIM, IEC 61850 and IEC 61499," *Informatik-Spektrum*, vol. 36, no. 1, pp. 58–68, 2013.

[14] M. de Miguel, T. Lambolais, M. Hannouz, S. Betgé-Brezetz, and S. Piekarec, "UML Extensions for the Specification and Evaluation of Latency Constraints in Architectural Models," in *2nd international workshop on Software and performance*. New York, NY, USA: ACM, 2000, pp. 83–88.

[15] A. Hennig, D. Revill, and M. Ponitsch, "From UML to Performance Measures–Simulative Performance Predictions of IT-Systems Using the Jboss Application Server With OMNET++," *Al-Dabass [1]*, vol. 1, no. 1.6, pp. 1–11, 2003.

[16] E. Ebeid, F. Fummi, and D. Quaglia, "Model-Driven Design of Network Aspects of Distributed Embedded Systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 4, pp. 603–614, 2015.

[17] E. Ebeid, S. Rotger-Griful, S. A. Mikkelsen, and R. H. Jacobsen, "A Methodology to Evaluate Demand Response Communication Protocols for the Smart Grid," in *International Conference on Communication Workshop (ICCW)*. IEEE, 2015, pp. 2012–2017.

[18] 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.

[19] Nord Pool Spot. [Online]. Available: http://www.nordpoolspot.com

Paper G

Agent-based Charging Scheduling of Electric Vehicles

The paper presented in this chapter has been accepted as a conference publication [8].

[8] Armin Ghasem Azar and Rune Hylsberg Jacobsen, "Agent-Based Charging Scheduling of Electric Vehicles," In *IEEE Online Conference on Green Communications (OnlineGreenComm)*, 2016, pages 64-69, doi: 10.1109/OnlineGreenCom.2016.7805408

Authors contribution: Armin Ghasem Azar and Rune Hylsberg Jacobsen conceived the main design of the agent-based demand response system; Rune Hylsberg Jacobsen contributed to the design and analysis of all simulations; Armin Ghasem Azar was the main editor of the manuscript; Rune Hylsberg Jacobsen contributed to the reviewing process.

2016 IEEE Online Conference on Green Communications (OnlineGreenComm)

Agent-based Charging Scheduling of Electric Vehicles

Armin Ghasem Azar and Rune Hylsberg Jacobsen
Department of Engineering, Aarhus University, Denmark
Emails: {aga, rhj}@eng.au.dk

***Abstract*—The electric vehicle technology intends to mitigate negative impacts of the energy challenge on the current transportation infrastructure. However, integrating a large number of such vehicles imposes a significant additional load to the grid and may overload it. This paper proposes a hierarchical event-driven multi-agent system framework for coordinated charging scheduling of electric vehicles. Household agents negotiate temporal travel patterns with substation agents to decide when electric vehicles should charge their batteries. A scalable load scheduling algorithm is proposed to schedule charging process of electric vehicles in real-time regardless of using any forecasting method. It aims to permit as many electric vehicles as possible to operate while keeping their aggregated charging energy consumption below continuous electricity-price-dependent thresholds over time. Simulations confirm that the framework benefits from charging flexibilities, reduces the charging cost, and shaves the grid's peak.**

Keywords*—Smart grid; Electric vehicle; Demand response; Coordinated charging scheduling; Multi-agent systems.

I. Introduction

Recently, the electrification of the current transportation system has emerged as one of the most crucial challenges in the power grid control. This revolution has yielded to the largest share of total energy consumption growth in the world [1]. Electric Vehicles (EVs) are environmentally friendly alternatives to ordinary vehicles. However, integrating an increasing number of EVs into the current grid imposes a significant additional load. Because of this large impact, charging scheduling of EVs is an important research topic since their uncoordinated charging process can jeopardize the efficiency and reliability of the power grid (see Fig. 1).

Demand Response (DR) programs enable the management of such challenge by providing incentives and technical assistance to consumers encouraging them to expand their energy management capabilities. Their main purpose is to enable the shifting of high energy-consuming loads to off-peak periods. Indeed, it is hard for consumers and grid operators to automatically adapt the charging behavior of EVs to grid's requirements. This paper claims that Multi-Agent Systems (MASs) significantly help the grid accomplish these operations in residential DR programs [2]. It proposes a hierarchical event-driven MAS framework for the coordination of charging scheduling of EVs. Here, agents are self-interested that need to maximize their profit. An individual household agent is characterized with a resident and an EV. Its objective is to control the charging process and cost of the EV. An operator and a management system specify each (secondary) substation agent. Its objective is to prevent overloading of the substation while satisfying household agents. It is assumed that both communication links and power lines are available between agents. The former exchanges the information between agents while the latter supplies households' electricity needs. This paper considers that agents communicate with each other through Internet without any limitation.

(a) Uncoordinated charging (b) Coordinated charging

Fig. 1. Grid's status according to uncoordinated and coordinated for charging scheduling of electric vehicles. The dashed red line is the grid's capacity.

Each substation performs independently and serves a finite set of households. Residents are required to set their transportation needs, i.e., arrival and departure times. Then, EVs communicate with substation agents and send charging events with a specific time resolution. In practice, these communications are instantiated asynchronously. Each event specifies the required charging load at the current time. Since the framework is decentralized, scheduling of events is executed at each substation, which significantly reduces the computational complexity of the scheduling problem. Each substation uses an identical instance of the scalable load scheduling algorithm to manage its incoming events. The algorithm employs real-time electricity prices to structure Electricity Consumption Thresholds (ECTs) [3]. In particular, it attempts to keep aggregated charging loads below these ECTs over time. It decides which EVs should charge their batteries at each time interval. Substation agents return these decisions to corresponding household agents to continue the remaining charging process.

Hence, as the main novelty of this framework, it does not utilize: 1) any historical data, 2) any forecasting methods, and 3) any complex optimization techniques. Instead, the framework considers how much energy EVs need now to charge their batteries and for how long they provide flexibility from the current moment. Since the grid operates in real-time, the framework requires a simple but intelligent optimization method. Relying on day-ahead global optimization techniques is not a feasible solution since the uncertainty in consumers' behavior is not negligible. Therefore, this paper studies the charging scheduling problem of EVs under a MAS framework as a continuous behavior-dependent optimization problem.

The paper is structured as follows. Section II reviews the related work. Section III presents the MAS framework. Section IV describes the negotiation model and EV charging scheduling algorithm. Section V provides the simulation setup and results. Finally, Section VI concludes the paper and provides some future work.

978-1-5090-2753-8/16/$31.00 ©2016 IEEE

II. Related Work

Coordinated charging scheduling of EVs is a challenging problem, in which Mukherjee *et al.* [1] have recently reviewed the recent contributions. They argued that considering only grid-to-EV power flows could be a logical first step toward challenging more complex bidirectional models. Rassaei *et al.* [4] studied the impact of a game-based DR framework on shaping the aggregated charging profiles taking uncertain arrival times into account. Although the work succeeded to minimize the peak, however, consumers' comfort level was not considered. Mohsenian-Rad *et al.* [5] proved that the dependability of optimizing the charging scheduling of EVs on the uncertain departure times is undeniable. They developed a closed-form solution to this problem with respect to time-of-use electricity prices.

However, a majority of current unidirectional model-driven works made centralized decisions for coordinating a high penetration of EVs, which was computationally intractable to handle. Xu *et al.* [6] framed a distributed concept into a hierarchical framework for coordinated charging of EVs. He *et al.* [7] proposed a centralized charging scheduling framework for charging and discharging of EVs, in which consumers could use it to minimize their energy cost. Deilami *et al.* [8] proposed a charging load scheduling algorithm for residential EVs using the amount of energy purchased in the day-ahead market based on forecasting methods.

Veit *et al.* [9] provided a MAS-based framework to the DR scheduling problem in response to the real-time supply. They formulated the constraints of individual scheduable electrical devices under the agent's control. Hu *et al.* [10] proposed an agent-based centralized concept for scheduling EVs including different layered agent types. Unda *et al.* [11] also presented an agent-based method for managing the battery charging problem of EVs in power distribution network according to electricity prices and grid stability constraints.

III. Multi-Agent System Framework

Agent-based modeling of real-world problems leads to more flexibility since: 1) different behavioral criteria can be analyzed, and 2) it is possible to add more *properties to agents* or add more *distributed agents* to the model. Fig. 2 pictures a conceptual view of the proposed framework. Agents are organized in a hierarchical way, where each household agent is serviced through a substation agent. Substation agents work independently. Let us assume the MAS framework includes

$$\begin{aligned} &\mathbb{S} = \{S_1, \ldots, S_m\}, \\ &\mathbb{H} = \{\{H_{1,1}, \ldots, H_{n,1}\}, \ldots, \{H_{1,m}, \ldots, H_{n',m}\}\}, \end{aligned} \tag{1}$$

where $\mathbb{S}$ and $\mathbb{H}$ are sets of substation and household agents, respectively. S_j is substation agent j, where $1 \leq j \leq m$. Then, $H_{i,j}$ is denoted as household agent i connected to substation agent j, where $1 \leq i \leq n$. According to grid's topology, $\forall S_j \in \mathbb{S}, \exists \{H_{1,m}, \ldots, H_{n,j}\} \in \mathbb{H}$. Substation agents can control different number of household agents, i.e., $n \neq n'$. Hereinafter, for readability of notations, subscripts (i.e., i and j) are eliminated. Since the framework is designed as an object-oriented system, the following formulates specifications of household and substation agent classes.

A. Household Agent

Once an EV arrives at home, the resident updates

$$\{\alpha, \beta\} \in \{\texttt{YYYY-MM-DD HH:mm:ss}\}, \tag{2}$$

where α and β denote the current arrival and desired departure times, respectively. To evaluate the framework under the maximum load, it is assumed that batteries are empty upon arrival and must be fully charged until departure. Let

$$\mathbb{P} = \left\{\sigma^\tau, \ldots, \sigma^t, \ldots, \sigma^\theta\right\}, \tag{3}$$

where $\mathbb{P}$ defines the known vector of EV's charging profile. Each $\sigma^t \in \mathbb{R}^*$ (kWh) is a charging event at time interval $t \in \{\texttt{YYYY-MM-DD HH:mm:ss}\}$ defining the energy needed between t and $t + \Delta t$. Vector $\mathbb{P}$ is divided into equal time slots of duration $\Delta t \in \mathbb{N}$ (sec/min). It should be considered that events are created continuously over time. The EV starts trying to charge its battery upon arrival, i.e., $\alpha \leq \tau$. Charging must finish before departure, i.e., $\theta \leq (\beta - \Delta t)$. Let

$$C^t = C^{t-\Delta t} + \left(dec^t \times \sigma^t\right), \tag{4}$$

where $C^t \in \mathbb{R}^+$ (kWh) is battery's capacity at time interval t. Also, $dec^t \in \{0, 1\}$ is a binary decision variable triggered by the substation agent stating that the EV can charge its battery at time interval t ($dec^t = 1$) or should resend the event at the next time interval $t + \Delta t$ ($dec^t = 0$). This decision is mainly dependent on

$$flex^t = \begin{cases} 1 & ((|\mathbb{P}'| \times \Delta t) + t) < (\beta - \Delta t), \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

where $flex^t$ is the flexibility status of EV at time interval t. $|\{\}|$ returns the vector's length. Here, $\mathbb{P}' \subseteq \mathbb{P}$, where $\mathbb{P}' = \left\{\sigma^t, \ldots, \sigma^\theta\right\}$. The period between α and β is the flexibility period. It enables the possibility of both *shifting the start time* and *aperiodically interrupting* the charging process. $flex^t = 1$ if there is still any possibility to interrupt the charging process and send charging events at future intervals.

Let us assume a resident decides to shift the departure time back, i.e., $\beta' < \beta$, where β' is the new departure time. Indeed, in reality, the less the difference between available time and the time required for charge, the higher the risk of actually not being able to complete the charging if the departure happens before. Nevertheless, the framework is flexible to accommodate sudden behavioral changes. Here, the EV should recalculate its flexibility status. If $flex^t = 1$, then, the resident will definitely receive a fully charged EV. Otherwise, the EV cannot be fully charged until the departure time. However, it will be in the charging mode until departure uninterruptedly.

As two main battery constraints, first, let

$$C^t \leq C^{t+\Delta t} \leq C^{max}, \tag{6}$$

where $C^{max} \in \mathbb{R}^+$ (kWh) is the battery's maximum capacity. This constraint avoids overcharging or discharging the battery over time. As the second constraint, let

$$\sum_{t=\tau}^{\theta} \left(dec^t \times \sigma^t\right) = C^{max}, \tag{7}$$

where it ensures that the battery should be fully charged until the departure time at the latest. Next part describes how a substation agent handles incoming events.

978-1-5090-2753-8/16/$31.00 ©2016 IEEE

Fig. 2. Conceptual view of the proposed hierarchical multi-agent system framework for the problem of coordinated charging scheduling of electric vehicles.

B. Substation Agent

A substation agent consists of two parts named *buffers* and *scheduler*. Once an event arrives at a substation agent, it is forwarded to the buffers part. Let

$$\mathbb{B} = \left\{ wb^t, ab^{t'}, db^{t'} \right\}, \tag{8}$$

where wb^t, $ab^{t'}$, and $db^{t'}$ are *waiting*, *allowed*, and *denied* buffers, respectively. In practice, $t \neq t'$, since these intervals, at which buffers are updated, are asynchronous. The first buffer is updated when a new event is received while the others are updated when the scheduler returns decisions. Let

$$\mathbb{B} \leftarrow \Phi\left(wb^t, \Pi^t\right), \tag{9}$$

where Φ is the scheduling algorithm that makes decisions. When the buffers part receives decisions, it removes events from wb^t and appends them to either $ab^{t'}$ if $dec^{t'} = 1$ or $db^{t'}$ if $dec^{t'} = 0$. The substation agent uses flexibilities provided by household agents to shave peak demand periods. Let

$$\Pi^t = \left(\sum_{\sigma^t \in wb^t} \sigma^t \right) \times \widehat{ep}^t, \tag{10}$$

where $\Pi^t \in \mathbb{R}^*$ (kWh) is a real-time normalized electricity-price-dependent Electricity Consumption Threshold (ECT). Each time interval's threshold is applied on events in wb^t to activate the decision making process. The following section describes the negotiation model and scheduling strategy.

IV. Negotiation Model and Charging Load Scheduling Algorithm

Agents are self-interested and have the following goals:

- Household agent: *Controlling the charging process of EV while managing its charging cost.*
- Substation agent: *Preventing overloading of the substation while satisfying household agents.*

These goals are not independent of each other. Household agents should control charging processes of EVs between arrival and departure times. They are interested to perform these processes in lower electricity cost periods. Substation agents also have to respect substation's capacity over time while attempting not to confront unnecessary rebound peaks. Therefore, all agents need to coordinate with each other to meet the individual goals of all agents.

A. Negotiation Model

Fig. 3 pictures the information flow diagram of the negotiation model. It functions as follows:

1) Household agents send their events to the substation agent asynchronously.
2) The substation agent aggregates these requests and sends them to the buffers part. Meanwhile, it updates the ECT of the current time interval accordingly.
3) The buffers parts appends incoming events to the waiting buffer and send the updated version to the scheduling algorithm.
4) The scheduling algorithm decides about events according to updated ECT and returns decisions to the buffers part.
5) The buffers part updates the buffers with respect to decisions and forwards the allowed and denied buffers to the substation agent.
6) The substation agent returns the decisions to household agents.

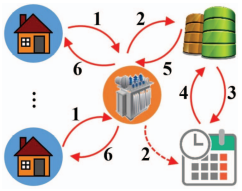


Fig. 3. Information flow diagram of the negotiation model. Numbers are in line with the text.

This negotiation model impacts the charging process and grid's status over time. Fig. 4 demonstrates how an uncoordinated charging process is shaped into a coordinated one from a household agent's perspective. Brown ovals symbolize the amount of energy exchanged with the system. If there is no coordination, the EV is continuously permitted to charge its battery until completion. This results in not using the potentiality of the provided flexibility period (refer to *Idle Period*). In the coordinated status, the substation agent divides both charging and idle periods of the EV on account of the sending-receiving time resolution of events. Obviously, there is no charging operation in idle periods.

From the substation's perspective, Fig. 5 shows a conceptual view of how the grid is influenced at three random consecutive time intervals by respecting the negotiation model. The substation's real capacity is mapped to dashed red lines. Furthermore, ECTs are shown by dashed green lines. The remaining space between red and green lines are denoted as the substation's *reserved* capacity. In practice, having no negotiation model causes overloading, which damages the substation. Obviously, this model considers the same priority to *previously denied* and *newly arrived* events. Each event block

978-1-5090-2753-8/16/$31.00 ©2016 IEEE

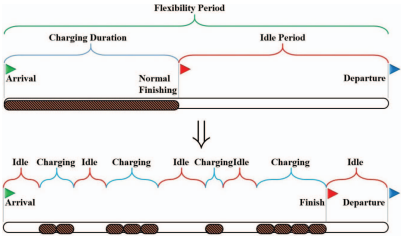


Fig. 4. Reshaping an uncoordinated charging process to a coordinated one.

belongs to one specific household agent. Next part presents the scheduling algorithm and how it decides about incoming events at each time interval.

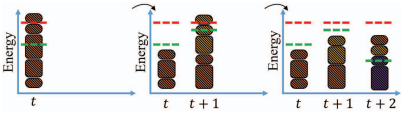


Fig. 5. A schematic view of the grid's status after three consecutive time intervals. New events at each time interval are known with a different color. Dashed red and green lines are substation's capacity and ECTs, respectively.

B. Scalable Charging Load Scheduling Algorithm

Algorithm 1 describes the EV charging load scheduling algorithm. For each event, stored in the waiting buffer, its flexibility is checked to decide whether shifting is possible. Then, if the remaining ECT is unable to cover the events located in the temp buffer, an *event selection procedure* is employed to select a subset of events subject to the remaining ECT. This paper considers the *Earliest Deadline First (EDF)* mechanism, where departure times are mapped to deadlines. EDF is derived from real-time systems theory, which is particularly heuristic while presenting better scalability properties [12]. The scheduling algorithm should return the decisions as soon as possible while not sacrificing the local optimality. Having complicated algorithms may delay the responses, which causes household agents resend new events. This brings chaos to the system. For more information about the scheduling algorithm, the reader is referred to [3].

According to Fig. 2, each substation agent runs an instance of this scheduling algorithm independently and continuously. This eases the complexity of the load scheduling problem compared to the situation, where an agent is responsible for responding events of all household agents in the grid. In that case, the problem would be computationally intractable [13]. If all household agents send events simultaneously, the scheduler has to run at maximum capacity. This will create a challenge since the scheduler will need much more time to handle these events. This will also cause household agents not to receive responses in a reasonable time frame. Therefore, they will start sending new events, which it will make a congestion. The buffers part handles this challenge. Since in practice, there are

Algorithm 1: Load scheduling algorithm

Input: Events in the waiting buffer, corresponding flexibilities, and the current ECT.
Output: Decisions.

Create a temp buffer tb^t;
foreach $\sigma^t \in wb^t$ **do**
  if $flex^t = 1$ **then**
    $tb^t = tb^t \cup \{\sigma^t\}$;
  else
    $dec^t = 1$;
    $\Pi^t = \Pi^t - \sigma^t$;
  end
end
if $\Pi^t \geq \sum_{\sigma^t \in wb^t} \sigma^t$ **then**
  foreach $\sigma^t \in tb^t$ **do**
    $dec^t = 1$;
    $\Pi^t = \Pi^t - \sigma^t$;
  end
else
  Call the event selection procedure (EDF mechanism);
  foreach $\sigma^t \in tb^t$ **do**
    if σ^t *is selected* **then**
      $dec^t = 1$;
      $\Pi^t = \Pi^t - \sigma^t$;
    end
    $dec^t = 0$;
  end
end
return *all decisions*;

some limitations on the communication side, e.g., bandwidth, throughput, etc., these events do not arrive at the same time. The times, at which the *scheduler fetches the waiting buffer* and the *waiting buffer is updated with new events* are asynchronous. These features reduce the computational complexity of the scheduling algorithm while confirming its scalability.

V. Simulation Setup and Results

This section first adjusts the simulation setup and then, provides the results and analyzes their performance.

A. Simulation Setup

The framework proposed has been modeled and coded in Matlab®. It runs on a computer with an Intel Core i7 2.0 GHz CPU and 12 GB memory. However, CPU and memory usages never exceed 33% and 400 MB in simulations, respectively. Since substation agents perform their responsibilities independently, the results of only one substation are reflected. We use specifications of batteries equipped in General Motors EV1 (share=30%), Toyota RAV1 (share=20%), and Nissan Altra (share=50%) [14]. Then, arrival and departure times follow normal distributions $\mathcal{N}(19, 10)$ and $\mathcal{N}(7.5, 1)$, respectively [4]. EVs arrive with empty batteries and will depart with fully charged status. Fig. 6 pictures the aggregated charging loads of 100 EVs before scheduling (uncoordinated). Fig. 7 illustrates EVs' probability density and real-time hourly basis electricity prices of Scandinavian countries[1]. Events are sent in $\Delta t = 1$

[1] http://www.nordpoolspot.com

978-1-5090-2753-8/16/$31.00 ©2016 IEEE

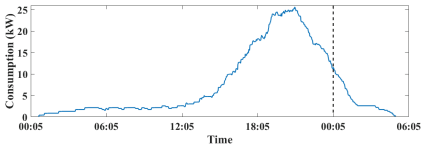


Fig. 6. Aggregated charging loads of 100 EVs. Two consecutive days are divided by a vertical dashed line. EVs arrive in the first day with probability of $\mathcal{N}(19, 10)$ and depart the day after with the probability of $\mathcal{N}(7.5, 1)$.

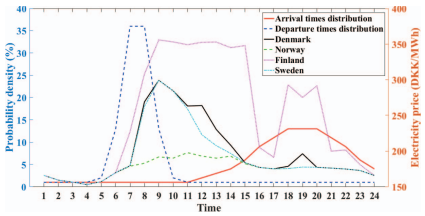


Fig. 7. EVs' probability density and real-time hourly basis electricity prices of Scandinavian countries (May 17, 2016).

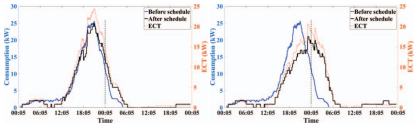
minute time resolution. The framework simulates two consecutive days, in which EVs arrive at the first day and depart in the second day. EVs are limited to charge their batteries once during this period.

B. Simulation Results

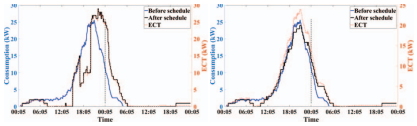
Fig. 8 shows the aggregated charging loads of EVs before and after scheduling according to the time-dependent ECTs applied over time. Table I analyzes the scheduling performance accordingly. All information are the result of averaging over multiple experiments in the same time frame.

For Denmark, its highest and lowest electricity prices are adjusted almost after departure and arrival times, respectively (see Fig. 8(a)). Their difference ratio is 1.95. This results in averagely 3.45% Charging Cost Reduction (CCR) for each household while averagely 5.22% Peak Demand Reduction (PDR), as pictured in Fig. 8(a). This PDR is the result of 10% Flexibility Usage Percentage (FUP) and 94.19 minutes of Average EV Delay (AED). Residents provide averagely 932.79 minutes of flexibility (refer to Section III-A). For instance, a Nissan Altra arrives at 19:00 and will depart at 08:00 the next day. This includes 780 minutes of flexibility. According to its charging duration (300 minutes [14]), although its battery can be fully charged until 24:00, however, charging is continued until almost 01:34 the next day. Finally, Maximum Computation Time (MCT) reports how long it takes for the substation agent to return decisions in the worst case among all intervals. For instance, household agents in Denmark averagely wait 0.28 seconds to receive a response to their events.

For Norway, as Fig. 8(b) pictures, PDR is 20.20%. The reason is the low difference ratio 1.29 in electricity prices (see Fig. 7). Having an inconsiderable fluctuation in electricity prices, ECTs are relatively low (compared to Denmark), which helps the scheduler shift more charging loads from the first



(a) Denmark (b) Norway



(c) Finland (d) Sweden

Fig. 8. Aggregated charging loads of 100 EVs before and after scheduling subject to real-time electricity-price-dependent ECTs.

TABLE I. PERFORMANCE ANALYSIS OF THE ALGORITHM

	CCR (%)	PDR (%)	FUP (%)	AED (min.)	MCT (sec.)
Denmark	3.45	5.22	10	94.19	0.28
Norway	3.58	20.20	28.23	263.40	0.21
Finland	19.53	-11.87	20	187.19	0.35
Sweden	2.68	6.63	9	84.25	0.27

CCR is Charging Cost Reduction. PDR is Peak Demand Reduction. FUP is Flexibility Usage Percentage. AED is Average EV Delay. MCT is Maximum Computation Time.

day to the second day. However, this influences AED, in which EVs are delayed averagely 263.40 minutes to fully charge their batteries. CCR is averagely 3.58%, which is due to the low difference ratio. However, CCR for Danish residents is lower compared to Norwegian residents, which depends on PDR.

For Finland, as Fig. 8(c) shows, there is a significant trade-off between CCR and PDR. Here, the different ratio is 2.32, which results in 19.53% CCR. This reduction is the result of using 20% FUP meaning averagely 263.40 minutes of AED. Finland's electricity prices are higher than other discussing countries and more fluctuating during arrival times (see Fig. 7). This causes the scheduler not to succeed in shaving the peak, which produces a significant rebound peak with -11.87% PDR. Nevertheless, this rebound peak uses a part of the substation's reserved capacity (see Fig. 5). This analysis proves that electricity-price-dependent ECTs significantly behaves differently in these countries.

For Sweden, as Fig. 8(d) shows, the evaluation results almost the same as Denmark. Here, the different ratio is also 1.95. However, the small difference in results is caused by a small fluctuation in the electricity prices during arrival times.

Fig. 9 shows the distribution of charging EVs over time (reflected by Fig. 8). Before scheduling, 61 EVs make the peak demand. PDRs, mentioned in Table I, also represent the reduction in the number of charging EVs. This analysis helps operators configure the grid's structure efficiently.

Fig. 10 presents the charging evolution of a Nissan Altra's battery. It arrives at 18:07 on May 17 and departs at 02:32 on May 19. After scheduling, the charging period is split into 571 one-minute slots starting from 17:01 on May 18 ending to 02:31 on May 19. From 18:00 to 22:33 on May 18, the battery remains with 20.28% charge.

978-1-5090-2753-8/16/$31.00 ©2016 IEEE

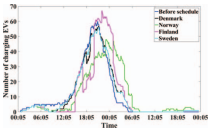


Fig. 9. Distribution of charging EVs.

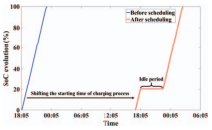
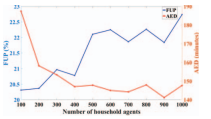


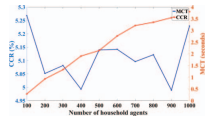
Fig. 10. Charging evolution of EV

As the final step, Fig. 11 demonstrates how increasing the number of household agents impacts on the performance. Fig. 11(a) shows how FUP and AED are changed when the number of households increases from 100 to 1000. Although FUPs and AEDs behave reversely, however, the result shows that household agents benefit more (in terms of AED) when the DR participation percentage increases. FUPs increase since the scheduler uses more flexibility to be able to shave the peak. Nevertheless, since ECTs are higher than the aggregated load consumptions at some intervals (because of previously denied and newly received events), more events are responded to operate, which consequently, leads to a reduction in AEDs.

Fig. 11(b) complements this analysis based on CCRs and MCTs. The fluctuating nature of CCRs is due to having: 1) high ECTs because of the number of household agents, and 2) more complexity in selecting an optimal subset of events at each time interval. CCRs do not follow a normal and expected behavior compared to others. The ascending slope of MCTs is a consequence of having more events in the waiting buffer at each time interval. This indeed makes the scheduling algorithm take longer time than anticipated to return decisions. However, taking almost only 4 seconds to schedule 1000 events in the worst case scenario is surprisingly interesting, which confirms the scalability of the algorithm. In Danish electricity grid's infrastructure, averagely 100 households are serviced via a single substation. According to Table I, each household agent will see a delay of 0.28 seconds to receive a response to each event it sends. Since decision making takes place at each substation independently, the framework is able to accommodate for a large number of requests in a fair manner.



(a) FUP and AED



(b) CCR and MCT

Fig. 11. Performance analysis based on increasing the number of households.

VI. Conclusion and Future Work

This paper studied the problem of the uncoordinated charging process of electric vehicles. It proposed a hierarchical event-driven multi-agent system framework considering households' needs and substations' capacity limitations. A scalable load scheduling algorithm was presented to schedule incoming charging events efficiently in real-time. It takes flexibilities and required energies into account and runs an event selection procedure to allow some electric vehicles to charge their batteries at the current time interval and force the remaining ones to resend their events in the future. Simulation results supported the work while confirming that the framework was successful in shaving the peak and reducing the charging cost regardless of using any historical data or forecasting methods. Currently, we are working on a scalable solution to integrate communication networks with the demand response framework to measure network delays and quality of services. Other grid stability constraints, such as voltage violation limits, will also be integrated with the framework. Finally, the framework will be synthesized with the vehicle-to-grid technology, to study when and how much an electric vehicle should give the energy, stored in its battery, back to the grid.

Acknowledgment

This work has been funded by European Union FP7 program under grant agreement n° 619560 (SEMIAH).

References

[1] J. C. Mukherjee and A. Gupta, "A Review of Charge Scheduling of Electric Vehicles in Smart Grid," *IEEE Systems Journal*, vol. 9, no. 4, pp. 1541–1553, 2015.

[2] A. Clausen, A. Umair, Z. Ma, and B. N. Jørgensen, "Demand Response Integration Through Agent-Based Coordination of Consumers in Virtual Power Plants," in *International Conference on Principles and Practice of Multi-Agent Systems*. Springer, 2016, pp. 313–322.

[3] A. G. Azar and R. H. Jacobsen, "Appliance Scheduling Optimization for Demand Response," *International Journal on Advances in Intelligent Systems*, 2016.

[4] F. Rassaei, W.-S. Soh, and K.-C. Chua, "Demand Response for Residential Electric Vehicles With Random Usage Patterns in Smart Grids," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1367–1376, 2015.

[5] H. Mohsenian-Rad and M. Ghamkhari, "Optimal Charging of Electric Vehicles With Uncertain Departure Times: A Closed-Form Solution," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 940–942, 2015.

[6] Z. Xu, W. Su, Z. Hu, Y. Song, and H. Zhang, "A Hierarchical Framework for Coordinated Charging of Plug-in Electric Vehicles in China," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 428–438, 2016.

[7] Y. He, B. Venkatesh, and L. Guan, "Optimal Scheduling for Charging and Discharging of Electric Vehicles," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1095–1105, 2012.

[8] S. Deilami, A. S. Masoum, P. S. Moses, and M. A. Masoum, "Real-Time Coordination of Plug-in Electric Vehicle Charging in Smart Grids to Minimize Power Losses and Improve Voltage Profile," *IEEE Transactions on Smart Grid*, vol. 2, no. 3, pp. 456–467, 2011.

[9] A. Veit and H.-A. Jacobsen, "Multi-Agent Device-Level Modeling Framework for Demand Scheduling," in *IEEE International Conference on Smart Grid Communications*, 2015, pp. 169–174.

[10] J. Hu, H. Morais, M. Lind, and H. W. Bindner, "Multi-Agent Based Modeling for Electric Vehicle Integration in a Distribution Network Operation," *Electric Power Systems Research*, vol. 136, pp. 341–351, 2016.

[11] I. G. Unda, P. Papadopoulos, S. Skarvelis-Kazakos, L. M. Cipcigan, N. Jenkins, and E. Zabala, "Management of Electric Vehicle Battery Charging in Distribution Networks With Multi-Agent Systems," *Electric Power Systems Research*, vol. 110, pp. 172–179, 2014.

[12] G. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Springer Science & Business Media, 2011, vol. 24.

[13] J. C. Mukherjee and A. Gupta, "Distributed Charge Scheduling of Plug-In Electric Vehicles Using Inter-Aggregator Collaboration," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–11, 2016.

[14] K. Qian, C. Zhou, M. Allan, and Y. Yuan, "Load Model for Prediction of Electric Vehicle Charging Demand," in *IEEE International Conference on Power System Technology*, 2010, pp. 1–6.

978-1-5090-2753-8/16/$31.00 ©2016 IEEE

Paper H

A Non-Cooperative Framework for Coordinating a Neighborhood of Distributed Prosumers

The paper presented in this chapter has been submitted to IEEE Transactions on Smart Grid as a journal publication [9].

[9] Armin Ghasem Azar, Hamidreza Nazaripouya, Behnam Khaki, Chi-Cheng Chu, Rajit Gadh, and Rune Hylsberg Jacobsen, "A Non-Cooperative Framework for Coordinating a Neighborhood of Distributed Prosumers," Submitted to *IEEE Transactions on Smart Grid*, 2017

Authors contribution: Armin Ghasem Azar conceived the main design of the decentralized negotiation framework; Hamidreza Nazaripouya and Behnam Khaki matured and formalized the proposed methodology and contributed to the design, execution and analysis of all simulations; Armin Ghasem Azar was the main editor of the manuscript; Hamidreza Nazaripouya, Behnam Khaki, and Rune Hylsberg Jacobsen contributed to the text and reviewing process.

A Non-Cooperative Framework for Coordinating a Neighborhood of Distributed Prosumers

Armin Ghasem Azar, Hamidreza Nazaripouya, Behnam Khaki,
Chi-Cheng Chu, Rajit Gadh, and Rune Hylsberg Jacobsen

***Abstract*—This paper introduces a scalable framework for coordinating a neighborhood of residential prosumers including smart appliances, photovoltaics, and battery energy storage systems. Prosumers, to maximize their comfort level and profit at each instant of time, individually take advantage of their consumption, generation, and storage flexibilities by negotiating with an aggregator in a non-cooperative environment. The aggregator matches prosumers' supply and demand with the objectives of maximizing its profit and minimizing the grid purchase. The framework is comprised of two separate multi-objective mixed integer nonlinear programming models for prosumers and the aggregator, which are solved by the evolutionary NSGA-III algorithm independently. A distributed negotiation approach, to enable prosumers and the aggregator to negotiate on concurrent packaged power and price offers with private utility functions and preferences, is incorporated into the framework. This approach, to converge to an acceptable agreement, follows an alternating-offer production protocol and a reactive utility value concession strategy. Moreover, to reduce the computation overhead in parallel bilateral negotiations, a virtual power plant is employed to proceed the negotiation on behalf of prosumers. The effectiveness of the framework is evaluated through four case studies based on several economic and environmental assessment metrics.**

***Keywords*—Prosumers, scheduling and matching, distributed non-cooperative negotiation, multi-objective optimization.**

ACRONYMS

AOD	Average Appliance Operation Delay
BESS	Battery Energy Storage System
EDF	Earliest Deadline First
FUR	Average Flexibility Usage Rate
MO-MINLP	Multi-Objective Mixed Integer Nonlinear Programming
MOO	Multi-Objective Optimization
NSGA-III	Non-dominated Sorting Genetic Algorithm-III
PAR	Peak-to-Average Ratio
PCB	Average Prosumer Cost-Benefit
PDR	Peak Demand Reduction
PV	Photovoltaic
SLR	Average Self Load-Satisfaction Rate
SOC	State of Charge
SSR	Average Self Sufficiency Rate
VPP	Virtual Power Plant

A. G. Azar and R. H. Jacobsen are with the Department of Engineering, Aarhus University, Denmark, e-mails: {aga, rhj}@eng.au.dk.

H. Nazaripouya, B. Khaki, C. Chu, and R. Gadh are with the Department of Mechanical Engineering, University of California Los Angeles (UCLA), USA, emails: {hnazari, behnamkhaki, peterchu, gadh}@ucla.edu.

NOMENCLATURE

Note: Notations follow $①^{③}_{②}(④)$, where ①-④ refer to an entity, its index, its feature, and the time/iteration index, respectively.

Constants

$\mathcal{A}$	Aggregator
$\widetilde{K}$	Number of feasible behavior matrices of the aggregator
$a_{j,i}$	j-th appliance of prosumer ρ_i
$\varepsilon_{j,i}$	Exact operating end time of appliance $a_{j,i}$
$\beta_{j,i}$	Desired operating end time of appliance $a_{j,i}$
$\theta_{j,i}$	Desired operating flexibility of appliance $a_{j,i}$
n_i	Number of appliances of prosumer ρ_i
$\alpha_{j,i}$	Desired operating start time of appliance $a_{j,i}$
B_i	BESS of prosumer ρ_i
B_i^{cap}	BESS capacity of prosumer ρ_i (kWh)
$B_i^{\overline{c}}$	Maximum charging power of the BESS of prosumer ρ_i (kW)
$B_i^{\overline{d}}$	Maximum discharging power of the BESS of prosumer ρ_i (kW)
$B_i^{\overline{\text{SOC}}}$	Maximum SOC for the BESS of prosumer ρ_i
$B_i^{\underline{\text{SOC}}}$	Minimum SOC for the BESS of prosumer ρ_i
δ	Negotiation convergence tolerance
ϵ	Decay rate controller for time-dependent concession values
W	Number of generations in the NSGA-III
$\mathcal{G}$	The grid
$\mathcal{T}$	Number of negotiation iterations
P	Projection operator
ρ_i	i-th prosumer
K	Number of feasible behavior pairs of prosumers
PV_i	PV system of prosumer ρ_i
PV_i^{cap}	Power generation capacity of the PV of prosumer ρ_i (kW)
Δt	Time interval resolution
T	Number of time intervals
$\psi_{\mathcal{A}}$	Utility function of the aggregator
$\psi_{\mathcal{V}}$	Utility function of the VPP
$\mathcal{V}$	VPP

Indexes

$\widetilde{k}$	Behavior matrix index
j	Appliance index
ι	Negotiation iteration index
i	Prosumer index
k, k'	Behavior pair indexes
t	Time interval index

Sets

$\mathcal{AP}_i$	Set of appliances of prosumer ρ_i
$\varnothing$	Empty set
$lp_{j,i}$	Load profile of appliance $a_{j,i}$
$\mathbb{B}$	Binary numbers
$\mathbb{N}$	Natural numbers
$\mathbb{R}$	Real numbers
$\mathcal{NS}_i$	Set of non-shiftable appliances of prosumer ρ_i
$\mathcal{P}$	Set of prosumers
$\mathcal{S}_i$	Set of shiftable appliances of prosumer ρ_i
Variables	
$\widetilde{\Upsilon}_{\mathcal{A}}(t)$	Set of feasible behavior matrices of the aggregator
$\widetilde{\mathscr{Z}}(t)$	The zone of agreement in the negotiation
$\tau_{j,i}(t)$	Load demand of appliance $a_{j,i}$ (kW)
$B_i^c(t)$	Charging power of the BESS of prosumer ρ_i (kW)
$B_i^d(t)$	Discharging power of the BESS of prosumer ρ_i (kW)
$B_i^e(t)$	Amount of energy stored in the BESS of prosumer ρ_i until time interval t (kWh)
$B_i^{\text{SOC}}(t)$	SOC value of the BESS of prosumer ρ_i
$v_i^c(t)$	Binary charging status of the BESS of prosumer ρ_i
$v_i^d(t)$	Binary discharging status of the BESS of prosumer ρ_i
$\widetilde{\mathcal{BM}}_{\mathcal{A}}^{\widetilde{k}}(t)$	$\widetilde{k}$-th behavior matrix of the aggregator
$\mathcal{BP}_i^k(t)$	k-th behavior pair of prosumer ρ_i
$\mathcal{BP}_i^{res}(t)$	Reservation behavior pair of prosumer ρ_i
$\widetilde{\mathcal{BP}}_i^{\widetilde{k}}(t)$	$\widetilde{k}$-th behavior pair of the aggregator concerning prosumer ρ_i
$\widetilde{\mathcal{BP}}_i^{res}(t)$	Reservation behavior pair of the aggregator assumed for prosumer ρ_i
$dec_{j,i}(t)$	Binary decision variable of the operating status of appliance $a_{j,i}$
$\omega_{\mathcal{A}}(\iota)$	Reactive concession value of the aggregator
$\omega_{\mathcal{V}}(\iota)$	Reactive concession value of the VPP
$\Pi_{\mathcal{A}}(\iota)$	Desired utility value of the aggregator
$\Pi_{\mathcal{V}}(\iota)$	Desired utility value of the VPP
$\phi^u(t)$	Maximum offerable price for prosumers to trade electric energy (\$/kWh)
$\phi^l(t)$	Minimum offerable price for prosumers to trade electric energy (\$/kWh)
$\phi_{\mathcal{G}}^u(t)$	Price of the grid to sell electric energy to the aggregator (\$/kWh)
$\phi_{\mathcal{G}}^l(t)$	Price of the grid to buy electric energy from the aggregator (\$/kWh)
$\phi_i(t)$	Price of prosumer ρ_i offered to the aggregator via the VPP (\$/kWh)
$\phi_i^{res}(t)$	Reservation price offer of prosumer ρ_i (\$/kWh)
$\widetilde{\phi}_i(t)$	Price of the aggregator offered to prosumer ρ_i via the VPP (\$/kWh)
$Z_{\mathcal{A}}(t)$	Set of feasible desired offer packages of the aggregator
$Z_{\mathcal{V}}(t)$	Set of feasible desired offer packages of the VPP
$\overline{Z}_{\mathcal{A}}(\iota)$	Subset of feasible desired offer packages of the aggregator
$\overline{Z}_{\mathcal{V}}(\iota)$	Subset of feasible desired offer packages of the VPP
$flex_{j,i}(t)$	Binary flexibility status of appliance $a_{j,i}$
$\sigma_{\mathcal{A}}(\iota)$	Offer package sent from the aggregator to the VPP
$\sigma_{\mathcal{A}}^{res}(t)$	Reservation offer package of the aggregator
$\sigma_{\mathcal{V}}(\iota)$	Offer package sent from the VPP to the aggregator
$\sigma_{\mathcal{V}}^{res}(t)$	Reservation offer package of the VPP
$\aleph_i(t)$	Power exchanged between prosumer ρ_i and the grid (kW)
$\aleph_i^{\text{B2G}}(t)$	Power transferred from the BESS of prosumer ρ_i to the grid (kW)
$\aleph_i^{\text{B2L}}(t)$	Power transferred from the BESS to appliances of prosumer ρ_i (kW)
$\aleph_i^{\text{G2B}}(t)$	Power transferred from the grid to the BESS of prosumer ρ_i (kW)
$\aleph_i^{\text{G2L}}(t)$	Power trasnferred from the grid to appliances of prosumer ρ_i (kW)
$\aleph_i^{\text{P2B}}(t)$	Power transferred from the PV to the BESS of prosumer ρ_i (kW)
$\aleph_i^{\text{P2G}}(t)$	Power transferred from the PV of prosumer ρ_i to the grid (kW)
$\aleph_i^{\text{P2L}}(t)$	Power transferred from the PV to appliances of prosumer ρ_i (kW)
$\aleph_i^{res}(t)$	Reservation power offer of prosumer ρ_i to exchange (kW)
$\widetilde{\aleph}_{\mathcal{A}}(t)$	Power exchanged between the aggregator and the grid (kW)
$\widetilde{\aleph}_i(t)$	Power exchanged between the aggregator and prosumer ρ_i (kW)
$\overline{\aleph}_i(t)$	Extreme value of maximizing the comfort level of prosumer ρ_i
$\underline{\aleph}_i(t)$	Extreme value of maximizing the benefit of prosumer ρ_i
$\varphi_{\mathcal{A}}(\iota)$	Offer package projection weight of the aggregator imposed on the offer package received from the VPP
$\varphi_{\mathcal{V}}(\iota)$	Offer package projection weight of the VPP imposed on the offer package received from the aggregator
$\Upsilon_i(t)$	Set of feasible behavior pairs of prosumer ρ_i
$PV_i^g(t)$	PV generation of prosumer ρ_i (kW)
$\mathcal{SI}_{\mathcal{A}}^{\widetilde{k}}(t)$	Satisfaction index of offer package $\sigma_{\mathcal{A}}^{\widetilde{k}}(\iota)$ sent from the aggregator to the VPP
$\mathcal{SI}_{\mathcal{A}}^{res}(t)$	Satisfaction index of reservation offer package $\sigma_{\mathcal{A}}^{res}(t)$ of the aggregator
$\mathcal{SI}_i^k(t)$	Satisfaction index of behavior pair $\mathcal{BP}_i^k(t)$ of prosumer ρ_i
$\mathcal{SI}_i^{res}(t)$	Satisfaction index of reservation behavior pair $\mathcal{BP}_i^{res}(t)$ of prosumer ρ_i
$\mathcal{SI}_{\mathcal{V}}(\iota)$	Satisfaction index of offer package sent from the VPP to the aggregator
$\mathcal{SI}_{\mathcal{V}}^{res}(t)$	Satisfaction index of reservation offer package $\sigma_{\mathcal{V}}^{res}(t)$ of the VPP
$\varsigma_{\mathcal{A}}(\iota)$	Time-dependent concession value of the aggregator
$\varsigma_{\mathcal{V}}(\iota)$	Time-dependent concession value of the VPP
$\chi(\iota)$	Weighted offer package

I. Introduction

MODERNIZING the current power grid through the integration of distributed Photovoltaics (PVs) and Battery Energy Storage Systems (BESSs) paves the road towards the

materialization of the vision of a smart grid [1]. In an attempt to promote this revolution, such integration provides electricity *prosumers* with the opportunity to supply load demands of their smart appliances locally and trade their surplus power with the grid through balancing services, e.g., aggregators [2]. Two main challenges pertaining to introducing prosumers and aggregators within the smart grid are: i) the development of a practical decision-making model for them, and ii) the exploitation of an efficient coordination strategy to enable the communication between them. Hence, the following formally states the research problem.

A. Research Problem

To address the above-mentioned challenges in a smart grid composed of a neighborhood of distributed prosumers, each of them including a set of smart appliances, a PV, and a BESS, which communicate with an aggregator:

i) Develop a distributed framework such that:
   - Prosumers with multiple conflicting objectives, by taking advantage of their consumption, generation, and storage flexibilities, attempt to schedule the operation of their appliances, utilize their PV production, and share their surplus power with the grid; and
   - The aggregator with similar but independent multiple conflicting objectives, to ensure balance in the electricity grid, intends to match prosumers' supply to demand.

ii) Propose a negotiation approach such that:
   - The automated negotiation of concurrent power and price between prosumers and the aggregator is enabled;
   - The convergence in the negotiation to a feasible solution within a reasonable time and acceptable to all negotiators is guaranteed; and
   - The flexibility information, objective (utility) functions, and matching contracts during the course of a negotiation are kept private.

B. Research Motivation

Fig. 1 shows a high-level view of the system model proposed in this paper. We develop a Multi-Objective Mixed Integer Nonlinear Programming (MO-MINLP) model for prosumers to enable them to manage their resources following their available flexibilities and the electricity prices adjusted by the market [3]. The model for each prosumer handles mathematical constraints of its equipment beside confronting the conflicting objectives of maximizing its: i) *comfort level* by operating the smart appliances in time and charging the BESS during low-price periods as much as possible (to utilize it during high-price periods), and ii) *profit* by selling more power to the grid [4]. Prosumers, by benefiting from the Non-dominated Sorting Genetic Algorithm-III (NSGA-III) [5], strategically make trade-offs over *behavior pairs*, each of which declaring the amount of "power" to sell/buy and "price" to trade it. The

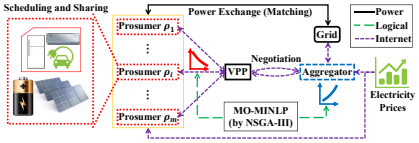


Fig. 1. System model of the proposed framework.

NSGA-III translates this trade-off into a set of non-dominated solutions lying on the first Pareto-front produced.

Aggregation services are fundamental to prevent any imbalance between supply and demand in the grid, and to ensure that both utility companies as well as prosumers derive benefits [6]. We assume each group of prosumers and the corresponding aggregator is determined beforehand, and different groups do not share members, essentially allowing us to concentrate on the interactions of a single group. The aggregator matches surplus power to demand. This matching is activated through electricity price fluctuations in the market. It intends to, by considering "demand amount with buying price" from buyer prosumers and "surplus amount with selling price" from seller prosumers, optimize its objectives of "maximizing the profit" and "minimizing the grid purchase" simultaneously. This is done by using a similar MO-MINLP model, which is also approached by the NSGA-III to generate a set of non-dominated solutions (resting on the first Pareto-front) to the aggregator's optimization problem.

Currently, clear mathematical models for trading transactions of energy and ancillary services have not been defined yet, particularly due to a lack of methodologies for the quantification of social and financial benefits of prosumers as well as aggregators. Concurrent *bilateral* negotiations between prosumers and the aggregator at each instant of time enable such services in the grid. However, there is a possibility that the amount of power that each prosumer intends to trade during the course of a negotiation changes. Such fluctuations may lead to an infeasible aggregated matching solution [2]. Furthermore, bilaterally negotiating with an increasing number of prosumers to reach an overall agreement is computationally expensive and imposes a substantial delay in communications.

This paper employs a multi-issue negotiation approach, which is conducted in a non-cooperative environment [7]. This approach follows an alternating-offer production protocol and a reactive utility value concession strategy. An intermediary Virtual Power Plant (VPP), acting on behalf of prosumers, negotiates with the aggregator about *aggregated* power and price. In fact, in a non-cooperative environment, no private information is shared: i) among prosumers, and ii) between the VPP and the aggregator. This private information consists of: i) prosumers' available flexibility and objective functions, ii) the VPP's utility function, and iii) the aggregator's objective and utility functions.

At each instant of time, prosumers provide the VPP with their first Pareto-fronts including a finite number of behavior

pairs. The aggregator, at the same time, produces its Pareto-front consisting of a set of feasible behavior matrices. The VPP, to respect the negotiation protocol, produces behavior matrices based on the received behavior pairs. The negotiation approach assumes the negotiators (the VPP and aggregator) have nonlinear utility functions and start the negotiation with an offer package (i.e., behavior matrix) providing the highest possible utility value. A novel evaluation metric, named *satisfaction index*, is also introduced to quantify to which extend each behavior pair/matrix: i) benefits from the current available flexibility, and ii) provides a price offer close to its extreme value (maximum or minimum electricity price depending on being a seller or buyer prosumer, respectively). This metric help negotiators at each negotiation interval identify the most beneficial offer package. Negotiators continuously concede to their pre-defined reservation offer package (the worst but feasible behavior in terms of power and price). That is, they neither propose nor accept any offer package with utility value lower than their reservation utility values. The VPP in the end of negotiation returns the index of the behavior pair, agreed with the aggregator on, to each prosumer.

C. Related Work

Residential appliance load scheduling, to reduce the electricity bills following price fluctuations, has been explored by a number of researchers [3], [8]–[11]. In [3], a multi-objective demand response system concentrating on reducing the total electricity bills and flattening the aggregated peak demand at the same time is proposed, where each resident provides the system with the desired expectation of his load demand scenario accompanied with flexibility time limits of his controllable appliances. The agent-based version of such system to coordinate charging scheduling of a large number electric vehicles is developed in [8], in which households, to decide when electric vehicles should charge, negotiate on their temporal travel patterns with substation agents. A central energy management system, with the aim of minimizing the grid purchase, is proposed in [9] and [10], where it controls households' appliances based on their reputations in storing their surplus PV generation in a shared BESS. A distributed version of such system is proposed in [11], where under a dynamic pricing system, a coordination strategy fairly controls the operation of appliances while respecting the transformer capacity limits. Even though these models incentivize the households to modify their consumption pattern to achieve lower electricity bills, however, they fail to study the impact of the high penetration of households, with different ownership levels of shiftable appliances and various flexibility types, on households' economic as well as on the distribution grid.

A very comprehensive review of scheduling problems of distributed energy resources, such as PVs, from various aspects is done in [2], where the authors propose considering microgrids and VPPs as two suitable potential solutions. Limited research has been conducted on developing a scalable real-time framework for coordinating scheduling, sharing, and matching tasks engaged with a non-cooperative negotiation approach respecting negotiators' private information [12]–[24].

In [12], a day-ahead demand-side management mechanism for prosumers formulated as a non-cooperative game with a single objective of reducing monetary expenses is proposed. It preserves prosumers' privacy, limited communication with the central unit is needed, and the peak is reduced by 12.6%. However, prosumers' load demand scenarios should be known in advance and cannot change during the process. They should also commit to follow strictly the resulting consumption pattern. In [13] and [14], a lead-acid BESS coupled with PV is modeled through a home energy management system. To quantify the self-consumption and self-sufficiency of the model, load demands are satisfied first by the PV, then by the energy stored in the BESS, and finally by the grid. The main challenge with this single-objective system is that they simply consider the excess energy is injected to the grid with a fixed rate without any negotiation. The challenges of rapid residential PV installations in the recent years is discussed in [15], where the authors, to overcome the difficulty in balancing supply and demand, propose three independent centralized, decentralized, and distributed approaches using small-scale distributed BESSs based on model predictive control methodologies.

In [16], given a real-time pricing scheme, a simple model for buildings with the basic components of a generator, a BESS, and loads is proposed allowing two-way energy trading via a broker based on differential game theory. The convergence condition and time, however, are only characterized based on a limited number of buildings. A distributed power sharing framework formulated as a repeated game between households in a microgrid is proposed in [17], where each household decides on amount of power to trade with the grid. Households, by taking advantage of the variability in their load consumption patterns, achieve cost savings up to 20%. However, they require to have a list of preferences of households, with which they individually prefer to negotiate. A submission-based double auction mechanism with linear functions for a set of prosumers, possessing PVs and BESSs, is proposed in [18], where the mechanism is able to achieve an exact demand supply balance in a day-ahead electricity market subject to having a full information of consumption and generation profiles.

In [19], a set of computationally expensive off-line and on-line algorithms for the real-time cooperative energy management of only two microgrids are presented. These algorithms, however, assume that the renewable energy generation offset, by the aggregated load of individual microgrids, is known ahead of time. A similar double cooperative game to minimize the overall costs of both the utility companies and the residential prosumers is formulated in [20]. In [21], the authors develop a strategy including a heuristic algorithm for optimizing decentralized energy exchange depending on prosumers' involvement and physical constraints of distribution networks, in which prosumers' cost is averagely reduced by 66% and the proportion of energy self-satisfaction reaches 98%.

In [22], a peer-to-peer energy sharing model with price-based demand response for prosumer including PVs are introduced. Although an energy sharing provider is defined to coordinate the power exchanges, however, no method for ensuring the match between demand and supply is proposed. Furthermore, peak demand reduction before and after using

the proposed model is only 2% while the computation time of the simulation is very high, e.g., around 175 seconds for 25 prosumers. A similar work to obtain flexible and efficient distributed energy management is also studied in [23], where the main purpose is to minimize the economic cost of prosumers in the energy sharing problem, which is tailored to a convex optimization problem.

D. Contributions

This paper, to account for the gaps identified in [9]–[24], makes the following key contributions:

- Proposing a scalable framework for coordinating scheduling, sharing, and matching tasks of a neighborhood of distributed prosumers communicating with an aggregator in real-time regardless of any historical data or forecasting services;
- Developing two MO-MINLP models for: i) prosumers to schedule their appliances and share their surplus energy with the grid (based on mathematical models defined for their consumption, generation, and storage flexibilities), and ii) the aggregator to match seller prosumers' supply to buyer prosumers' demand;
- Introducing a generic formulation for non-dominated solutions to MO-MINLP models and utilizing the NSGA-III to produce a set of such solutions accordingly;
- Implementing a novel satisfaction index metric for quantifying each non-dominated solution according to the amount of flexibility used and the profitability of the price offer;
- Incorporating a multi-issue negotiation approach into the framework following an alternating-offer production protocol and a reactive utility value concession strategy with a guarantee to a feasible solution acceptable to all within a finite time;
- Enabling an automated bilateral negotiation between a VPP, working on behalf of prosumers, and the aggregator in a non-cooperative environment, without sharing any private information (prosumers' available flexibility and objective functions, the VPP's utility function, and the aggregator's objective and utility functions) during the negotiation; and
- Evaluating the effectiveness of the framework through simulation of four case studies based on several economic and environmental assessment metrics.

E. Structure of the Paper

The rest of paper is structured as follows: Section II presents the system model of the framework; Section III introduces the multi-objective optimization algorithm; Section IV describes the negotiation approach; Section V provides the simulation setup and results; and finally, Section VI concludes the paper and provides the future work.

II. System Model of the Framework

The smart grid system in this paper is represented by a set of prosumers $\mathcal{P} = \{\rho_1, \ldots, \rho_i, \ldots, \rho_m\}$, which communicate with an aggregator $\mathcal{A}$ through a VPP $\mathcal{V}$ (will be described in Section IV-A1).

A. Prosumers

Prosumer ρ_i consists of a set of smart appliances, a PV system, and a BESS. It has two main responsibilities: i) scheduling appliances, and ii) selling/buying power to/from the grid [4]. Fig. 2 shows the prosumer's *power actions* at each time interval $t \in \mathbb{N}$.

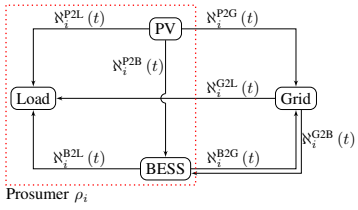


Fig. 2. Model diagram of power actions of prosumer ρ_i. The dotted red box conceptualizes the prosumer's physical equipment. "Load" points to the set of appliances. Exchanging power between prosumers and the "Grid" is controlled by the aggregator. Notations are described in Sections II-A1 to II-A3.

1) Appliances: Appliances are main drivers of prosumers' electricity consumption. Let

$$\mathcal{AP}_i = \left\{ \overbrace{a_{1,i}, \ldots, a_{j,i}}^{\mathcal{NS}_i}, \overbrace{a_{j+1,i}, \ldots, a_{n_i,i}}^{\mathcal{S}_i} \right\}, \tag{1}$$

$$lp_{j,i} = \{\tau_{j,i}(\alpha_{j,i}), \tau_{j,i}(\alpha_{j,i} + \Delta t), \ldots, \tau_{j,i}(\beta_{j,i})\}, \tag{2}$$

$$\sum_{t=\alpha_{j,i}}^{\beta_{j,i}} \tau_{j,i}(t) \times dec_{j,i} = \left| lp_{j,i} \right|, \tag{3}$$

$$\sum_{j=1}^{n_i} \tau_{j,i}(t) \times dec_{j,i}(t) = \aleph_i^{\text{P2L}}(t) + \aleph_i^{\text{B2L}}(t) + \aleph_i^{\text{G2L}}(t), \tag{4}$$

$$flex_{j,i}(t) = \begin{cases} 0 & (\beta_{j,i} - \alpha_{j,i}) \leq (\theta_{j,i} - t), \\ 1 & \text{otherwise}, \end{cases} \tag{5}$$

$$\begin{cases} dec_{j,i}(t) \in \{0,1\} & flex_{j,i}(t) = 1, \\ dec_{j,i}(t) = 1 & \text{otherwise}, \end{cases} \tag{6}$$

where $\mathcal{AP}_i$ is set of appliances of prosumer ρ_i. Each smart appliance $a_{j,i} \in \mathcal{AP}_i, 1 \leq j \leq n_i \in \mathbb{N}$ is a member of subsets of either non-shiftable $\mathcal{NS}_i \subseteq \mathcal{AP}_i$ or shiftable $\mathcal{S}_i \subseteq \mathcal{AP}_i$ appliances. *Shiftability* feature provides the prosumer with a *flexibility* degree to interrupt the operating cycle of appliances [3], [25]. Appliance $a_{j,i}$ follows a specific load profile $lp_{j,i}$ during its operating cycle, which is determined with respect to the program preset by the prosumer. $\tau_{j,i}(t) \in \mathbb{R}_{\geq 0}$ (kW) is the load demand of appliance $a_{j,i}$ at time interval t specifying the amount of power it needs to operate between t and $t + \Delta t$. Note that $\Delta t \in \mathbb{N}$ is the time interval resolution. $dec_{j,i}(t) \in \mathbb{B}$ is the binary decision variable of load demand $\tau_{j,i}(t)$ at time interval t. For shiftable appliances, the prosumer can decide to satisfy the load demand ($dec_{j,i}(t) = 1$) or postpone it to the next interval ($dec_{j,i}(t) = 0$) [8], [26]. The prosumer adjusts an operating deadline $\theta_{j,i} \in \mathbb{N}$ defining for how long the prosumer

is flexible in having the appliance's operation completed after its normal end time $\beta_{j,i} \in \mathbb{N}$. $\mathit{flex}_{j,i}(t) \in \mathbb{B}$ is the binary flexibility status of appliance $a_{j,i}$. Decision variable for non-shiftable appliances always equals one. These appliances, which are provided with no flexibility ($\beta_{j,i} = \theta_{j,i}$), operate uninterruptedly until their completion. This type of flexibility for shiftable appliances depends on the *desired* start time $\alpha_{j,i} \in \mathbb{N}$ and the appliance's load profile, which together adjust the appliance's normal end time. Calculating the flexibility value at each time interval (see (5)) allows prosumers to dynamically change the availability of their flexibility subject to the number of load demands remained until the full completion. Fig. 3 illustrates how the concept of flexibility reshapes the load profile of a shiftable appliance. Brown ovals, which their length depends on Δt, symbolize the load demands [8]. Note that the maximum upper bound in (3) can be $\theta_{j,i}$. Any appliance schedule that lies outside the desired start time ($t \not< \alpha_{j,i}$) and the assigned flexibility deadline ($t \not> \theta_{j,i}$) is considered invalid. $\aleph_i^{\text{P2L}}(t)$, $\aleph_i^{\text{B2L}}(t), \aleph_i^{\text{G2L}}(t) \in \mathbb{R}_{\geq 0}$ (kW) denote the power transferred from the PV, the BESS, and the grid to appliances, respectively. The proposed model ensures that the energy needed for each shiftable appliance over a given time horizon is fully satisfied.

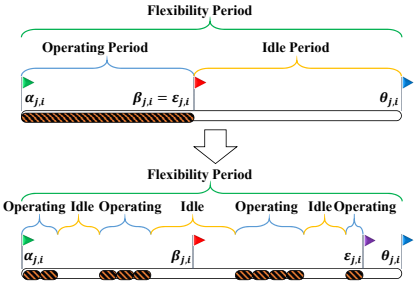


Fig. 3. Reshaping the operating cycle of a shiftable appliance $a_{j,i}$ through the concept of flexibility. $\varepsilon_{j,i} \in \mathbb{N}$ is its exact end time after scheduling.

2) Photovoltaic: Each prosumer is equipped with a locally installed PV system (behind the meter). Let

$$PV_i^g(t) = \aleph_i^{\text{P2L}}(t) + \aleph_i^{\text{P2B}}(t) + \aleph_i^{\text{P2G}}(t) \leq PV_i^{cap}, \quad (7)$$

where $PV_i^g(t), PV_i^{cap} \in \mathbb{R}_{\geq 0}$ (kW) are the amount of power that the PV generates at time interval t and its maximum generating capacity, respectively. $\aleph_i^{\text{P2B}}(t), \aleph_i^{\text{P2G}}(t) \in \mathbb{R}_{\geq 0}$ (kW) are the amounts of power transferred from the PV into the BESS and the grid, respectively [13]. PV production depends on external factors, such as weather condition [27]. Moreover, demand for electricity changes through the day and does not necessarily match to the PV production. Next part describes how BESSs, by storing the energy during off-peak and utilizing it during peak periods, can alleviate such challenges.

3) Battery Energy Storage System: Each prosumer owns a BESS. It accumulates excess energy created by the local PV and stores it to be used when there is an insufficient amount of energy to supply the demands. Let

$$B_i^e(t+1) = B_i^e(t) + \left(B_i^c(t) \times v_i^c(t) - B_i^d(t) \times v_i^d(t)\right) \times \Delta t, \quad (8)$$

$$B_i^c(t) = \aleph_i^{\text{P2B}}(t) + \aleph_i^{\text{G2B}}(t) \leq B_i^{\overline{c}}, \quad (9)$$

$$B_i^d(t) = \aleph_i^{\text{B2G}}(t) + \aleph_i^{\text{B2L}}(t) \leq B_i^{\overline{d}}, \quad (10)$$

$$v_i^c(t) + v_i^d(t) \leq 1, \quad (11)$$

$$B_i^e(0) = B_i^{cap} \times \frac{B_i^{\underline{\text{SOC}}} + B_i^{\overline{\text{SOC}}}}{2}, \quad (12)$$

$$B_i^{\underline{\text{SOC}}} \leq B_i^{\text{SOC}}(t) \leq B_i^{\overline{\text{SOC}}}, \quad (13)$$

$$B_i^{\text{SOC}}(t) = \frac{B_i^e(t)}{B_i^{cap}}, \quad (14)$$

where $B_i^e(t), B_i^{cap} \in \mathbb{R}_{>0}$ (kWh) are the amount of energy stored in the BESS until time interval t and the BESS capacity, respectively. Notations $B_i^c(t), B_i^d(t) \in \mathbb{R}_{\geq 0}$ (kW) denote the amounts of power the battery is "charged" and "discharged" with, respectively, subject to $B_i^{\overline{c}}, B_i^{\overline{d}} \in \mathbb{R}_{>0}$ (kW) as maximum charging and discharging power, respectively. Notations $\aleph_i^{\text{B2G}}(t), \aleph_i^{\text{G2B}}(t) \in \mathbb{R}_{\geq 0}$ (kW) denote the amounts of power transferred from the BESS to the grid and vice versa, respectively. $v_i^c(t), v_i^d(t) \in \mathbb{B}$ are binary charging and discharging variables, respectively. Concurrent charging and discharging are not allowed. The BESS at each time interval can charge, discharge, or remain silent (see (11)). $B_i^e(0)$ is the initial available amount of energy. $B_i^{\underline{\text{SOC}}}, B_i^{\overline{\text{SOC}}}, B_i^{\text{SOC}}(t) \in [0,1]$ are the lowest and highest possible State of Charges (SOCs) of the BESS, and its value at time interval t, respectively. Charging and discharging efficiencies, for clarity of presentation, are assumed to be one [28].

4) Optimization Model: Multi-Objective Optimization (MOO) is an area of multiple criteria decision-making, where mathematical optimization problems involving more than one objective function are solved [29]. Optimal decisions need to be taken in the presence of trade-offs between such conflicting objectives. When decision making is emphasized, the purpose of solving a MOO problem is referred to support *decision maker* (here prosumers) in finding the most preferred *non-dominated* solutions. The objective functions are said to be conflicting, whenever there exists an infinite number of non-dominated solutions. A solution does not improve for one objective unless it satisfies others. The main goal in MOO problems is to find a finite number of diverse solutions in the objective space.

The following defines a MOO model for each prosumer to tune its power actions. The model only accounts to the information given at time interval t and is independent of the information provided at time interval $t-\Delta t$ or will be provided at time interval $t+\Delta t$. Let

$$\underset{\{dec_{j,i}(t)\}_{j=1}^{n_i}, B_i^c(t)}{\text{maximize}} \sum_{j=1}^{n_i} \tau_{j,i}(t) \times dec_{j,i}(t) + B_i^c(t), \quad (15)$$

$$\underset{\aleph_i(t),\phi_i(t)}{\text{maximize}}\ \aleph_i(t)\times\phi_i(t)\times\Delta t, \tag{16}$$

subject to

$$(1)-(14),$$

$$\underline{\aleph}_i(t)\le\aleph_i(t)\le\overline{\aleph}_i(t), \tag{17}$$

$$\phi^l(t)\le\phi_i(t)\le\phi^u(t), \tag{18}$$

$$\underline{\aleph}_i(t)=PV_i^g(t)-\sum_{j=1}^{n_i}\tau_{j,i}(t)-B_i^{\overline{c}}, \tag{19}$$

$$\overline{\aleph}_i(t)=PV_i^g(t)+B_i^{\overline{d}}-\sum_{\substack{flex_{j,i}(t)=0,\\ \forall a_{j,i}\in\mathcal{AP}_i}}\tau_{j,i}(t), \tag{20}$$

$$\aleph_i(t)=\aleph_i^{\text{B2G}}(t)+\aleph_i^{\text{P2G}}(t)-\aleph_i^{\text{G2B}}(t)-\aleph_i^{\text{G2L}}(t), \tag{21}$$

where (15) aims at maximizing the comfort level by satisfying as many load demands as possible and charging the BESS as much as possible (subject to (13)) while (16) by selling more power to the grid intends to maximize the profit. These two objectives are in conflict with each other, since trying to inject more power to the grid results in jeopardizing the comfort level and vice versa. On one hand, this paper assumes satisfying load demands at each time interval partially affects the prosumer's comfort level [3]. On the other hand, the reason for considering the charging completeness of the BESS [30] in (15) is twofold: i) to allow wide solution-space exploration by making the corresponding objective function continuous, since variables $\{dec_{j,i}(t)\}_{j=1}^{n_i}$ are discrete; and ii) to store energy during low-price and utilize it (to either satisfy load demands or inject to the grid) during high-price periods. $\aleph_i(t)\in\mathbb{R}$ (kW) is the *desired* amount of power the prosumer strives to exchange with the grid coupled with a price offer $\phi_i(t)\in\mathbb{R}_{>0}$ (\$/kWh). This price is selected between $\left[\phi^l(t),\phi^u(t)\right]\in\mathbb{R}_{>0}$ (\$/kWh) as the minimum and maximum offerable price for trading energy, respectively. $\underline{\aleph}_i(t)$ and $\overline{\aleph}_i(t)$ are the optimum values of maximizing the comfort level and profit, respectively. If $\aleph_i(t)=\underline{\aleph}_i(t)$, all demanding appliances are allowed to operate and the BESS is fully charged subject to (13). If $\aleph_i(t)=\overline{\aleph}_i(t)$, the profit is maximized and appliances, without enough flexibility, are only allowed to operate. The remaining is sold to the grid.

For each prosumer at each time interval, one of the following cases can happen:

- The prosumer is a buyer, i.e., $\underline{\aleph}_i(t)\le\overline{\aleph}_i(t)<0$: The prosumer, through the electric power generated by the PV, the energy stored in the BESS, and the amount intended to purchase externally, fully satisfies appliances with insufficient flexibility. The profit, however, is always below zero. To lower the cost, the prosumer strives to pay less by offering a buying price close to $\phi^l(t)$.
- The prosumer is a seller, i.e., $0<\underline{\aleph}_i(t)\le\overline{\aleph}_i(t)$: Appliances with insufficient flexibility, by the PV generation and energy stored in the BESS, are fully satisfied and the surplus power is injected to the grid. The prosumer, to make the most beneficial contract, attempts to offer a selling price close to $\phi^u(t)$.
- The prosumer is flexible, i.e., $\underline{\aleph}_i(t)<0$ and $\overline{\aleph}_i(t)>0$: Both of the previous cases can happen, which makes a trade-off between the comfort level and profit. In this case, the prosumer can also be silent, i.e., $\aleph_i(t)=0$, where $\aleph_i^{\text{G2B}}(t)=\aleph_i^{\text{G2L}}(t)=\aleph_i^{\text{P2G}}(t)=\aleph_i^{\text{B2G}}(t)=0$. Power actions $\aleph_i^{\text{B2L}}(t),\aleph_i^{\text{P2L}}(t),\aleph_i^{\text{P2B}}(t)$ are tuned so as to maximize the comfort level, where the profit is zero.

Prosumers are not allowed to buy and sell at the same time. If the prosumer intends to buy electric power from the grid, i.e., $\aleph_i^{\text{G2B}}(t)>0$ or $\aleph_i^{\text{G2L}}(t)>0$, then, $\aleph_i^{\text{B2G}}(t)=0$ and $\aleph_i^{\text{P2G}}(t)=0$. If the prosumer is a seller, i.e., $\aleph_i^{\text{B2G}}(t)>0$ or $\aleph_i^{\text{P2G}}(t)>0$, then, $\aleph_i^{\text{G2B}}(t)=0$ and $\aleph_i^{\text{G2L}}(t)=0$.

B. Aggregator

The aggregator, which is accountable for balancing services [2], holds no physical connection with the grid and is only responsible for: i) trading prosumers' flexibilities in the market, and ii) making feasible and profitable contracts with them. Analysis of market mechanism and trading strategy falls outside of the scope of this paper [31].

1) Optimization Model: To enable the aggregator to make the decisions in response to matching prosumers' surplus to shortage, it runs the following optimization model. Let

$$\underset{\{\widetilde{\aleph}_i(t),\widetilde{\phi}_i(t)\}_{i=1}^m}{\text{maximize}}\ \Delta t\times\sum_{i=1}^{m}\begin{cases}\widetilde{\aleph}_i(t)\times\left(\phi_{\mathcal{G}}^l(t)-\widetilde{\phi}_i(t)\right) & \widetilde{\aleph}_i(t)>0,\\ \widetilde{\aleph}_i(t)\times\left(\widetilde{\phi}_i(t)-\phi_{\mathcal{G}}^u(t)\right) & \widetilde{\aleph}_i(t)<0,\end{cases} \tag{22}$$

$$\underset{\{\widetilde{\aleph}_i(t)\}_{i=1}^m}{\text{minimize}}\ -\sum_{i=1}^{m}\widetilde{\aleph}_i(t), \tag{23}$$

subject to

$$\begin{cases}0<\widetilde{\aleph}_i(t)\le\max\limits_{\forall\rho_i\in\mathcal{P}}\overline{\aleph}_i(t) & \aleph_i(t)>0,\forall\rho_i\in\mathcal{P},\\ \phi^l(t)\le\widetilde{\phi}_i(t)\le\phi_{\mathcal{G}}^l(t) & \\ \min\limits_{\forall\rho_i\in\mathcal{P}}\underline{\aleph}_i(t)\le\widetilde{\aleph}_i(t)<0 & \aleph_i(t)<0,\forall\rho_i\in\mathcal{P},\\ \phi_{\mathcal{G}}^u(t)\le\widetilde{\phi}_i(t)\le\phi^u(t) & \end{cases} \tag{24}$$

$$\sum_{i=1}^{m}\widetilde{\aleph}_i(t)+\widetilde{\aleph}_{\mathcal{A}}(t)=0, \tag{25}$$

where (22) attempts to maximize the aggregator's profit while (23) aims at minimizing the grid purchase. These objectives are in conflict with each other, since selling more to buyer prosumers and buying less from seller prosumers leads to buying more from the grid. $\widetilde{\aleph}_i(t)\in\mathbb{R}$ (kW) is the amount of power the aggregator trades with prosumer ρ_i coupled with a price offer $\widetilde{\phi}_i(t)\in\mathbb{R}_{>0}$ (\$/kWh). Note that (24) prevents the aggregator from requesting buyer prosumers to sell and vice versa. The aggregator, by (25), identifies $\widetilde{\aleph}_{\mathcal{A}}(t)\in\mathbb{R}$ (kW), as the amount of electric power at time interval t to exchange with the grid. This constraint ensures that the supply at each time interval meets demand in the grid. $\phi_{\mathcal{G}}^l(t),\phi_{\mathcal{G}}^u(t)\in\mathbb{R}_{>0}$ (\$/kWh) are the grid's prices for buying/selling energy from/to the aggregator, respectively. It should be emphasized

that, hereinafter, the *tilde* symbol in all notations are related to the aggregator.

The following section describes the NSGA-III algorithm employed to produce a set of non-dominated solutions for optimization problems of prosumers and the aggregator.

III. Multi-Objective Optimization Algorithm

Evolutionary algorithms for MOO problems, due to their independent search space structure, are among the most well-known meta-heuristic search mechanisms [32]. These algorithms form a subset of evolutionary computations, in which they generally involve techniques and implementation mechanisms inspired by biological evolutions, such as reproduction, mutation, recombination, natural selection, and survival of the fittest. This paper, by employing the evolutionary NSGA-III, enables each of prosumers and the aggregator at each time interval to generate a finite number of non-dominated solutions to their individual optimization problem.

The algorithm, in each generation, starts by generating an initial parent population including $Q \in \mathbb{N}$ feasible solutions. Section III-A for prosumers and Section III-B for the aggregator propose a generic formulation for producing these solutions. It, then, produces new solutions (offspring) and combines them with the parent population. The NSGA-III, to guarantee the diversity among such solutions, uses a *reference-point-based non-dominated sorting approach*. These points, at each time interval, are all permutations of extreme values of power $\left(\text{i.e., } \min_{\forall \rho_i \in \mathcal{P}} \underline{\aleph}_i(t), \max_{\forall \rho_i \in \mathcal{P}} \overline{\aleph}_i(t)\right)$ and price $\left(\text{i.e., } \phi^l(t), \phi^u(t)\right)$ located on a normalized hyper-plane (see Fig. 1 in [5]). Therefore, it associates each solution a reference value according to the reference points. To create the first Pareto-front, it determines Q closest solutions (in the combined parent and offspring populations) to the reference points using a niche-preservation operation and places them in the fronts accordingly. It continues until the maximum number of generations $W \in \mathbb{N}$ is reached. For more information, the reader is referred to [5] and [33].

A. Pareto-Solution Formulation for Prosumers

The following defines a generic formulation for Pareto-solutions of prosumers, which is independent of the algorithm being used to produce them. Let

$$\Upsilon_i(t) = \begin{bmatrix} \mathcal{BP}_i^1(t) & \mathcal{SI}_i^1(t) \\ \vdots & \vdots \\ \mathcal{BP}_i^K(t) & \mathcal{SI}_i^K(t) \end{bmatrix}, \tag{26}$$

$$\mathcal{BP}_i^k(t) \triangleq \left(\aleph_i^k(t), \phi_i^k(t)\right), \tag{27}$$

$$\mathcal{SI}_i^k(t) \triangleq \begin{cases} \frac{\aleph_i^k(t)}{\overline{\aleph}_i(t)} + \frac{\phi_i^k(t)}{\phi^u(t)} & \aleph_i^k(t) > 0, \\ \frac{\aleph_i^k(t)}{\underline{\aleph}_i(t)} + \frac{\phi^l(t)}{\phi_i^k(t)} & \aleph_i^k(t) < 0, \end{cases} \tag{28}$$

where $\Upsilon_i(t)$ at each time interval is produced once and is comprised of feasible *behavior pairs* $\mathcal{BP}_i^k(t), 1 \le k \le K$. Fig. 4 shows how these behavior pairs, according to (1)-(14) and (17)-(21), are randomly generated. $\Upsilon_i(t)$ is ultimately sorted descendingly by its second column. *Satisfaction index* $\mathcal{SI}_i^k(t) \in (0, 2]$ is a measure of to which extent $\mathcal{BP}_i^k(t)$, depending on the prosumer's status, uses the available flexibility and provides a more beneficial price offer [9], [10]. Its maximum value is reached, whenever: i) a buyer prosumer ($\aleph_i^k(t) < 0$) purchases the lowest possible amount of power, i.e., $\underline{\aleph}_i(t)$, for the lowest possible price, i.e., $\phi^l(t)$; ii) a seller prosumer ($\aleph_i^k(t) > 0$) sells the maximum possible amount of electricity, i.e., $\overline{\aleph}_i(t)$, for the highest possible price, i.e., $\phi^u(t)$.

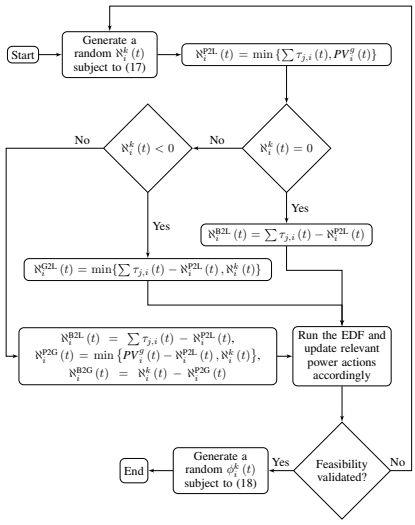


Fig. 4. Flow chart of generating a feasible behavior pair $\mathcal{BP}_i^k(t)$. $\sum \tau_{j,i}(t)$ is the summation of load demands of appliances with insufficient flexibility. The Earliest Deadline First (EDF) mechanism, which is derived from real-time systems theory [8], is applied on the remaining appliances. Choosing proper power actions to update after running the EDF depends on $\aleph_i^k(t)$. Feasibility is related to constraints formulated in Section II-A4.

B. Pareto-Solution Formulation for the Aggregator

The aggregator, similar to prosumers, uses an algorithm-independent model for generating a set of feasible non-dominated solutions to its optimization problem. Let

$$\widetilde{\Upsilon}_{\mathcal{A}}(t) = \begin{bmatrix} \widetilde{\mathcal{BM}}_{\mathcal{A}}^1(t) & \widetilde{\mathcal{SI}}_{\mathcal{A}}^1(t) \\ \vdots & \vdots \\ \widetilde{\mathcal{BM}}_{\mathcal{A}}^{\widetilde{K}}(t) & \widetilde{\mathcal{SI}}_{\mathcal{A}}^{\widetilde{K}}(t) \end{bmatrix}, \tag{29}$$

$$\widetilde{\mathcal{BM}}_{\mathcal{A}}^{\widetilde{k}}(t) \triangleq \begin{bmatrix} \widetilde{\mathcal{BP}}_1^{\widetilde{k}}(t) \\ \vdots \\ \widetilde{\mathcal{BP}}_m^{\widetilde{k}}(t) \end{bmatrix}, \tag{30}$$

$$\widetilde{\mathcal{BP}}_i^{\widetilde{k}}(t) \triangleq \left(\widetilde{\aleph}_i^{\widetilde{k}}(t), \widetilde{\phi}_i^{\widetilde{k}}(t)\right), \tag{31}$$

$$\mathcal{SI}_{\mathcal{A}}^{\widetilde{k}}(t) \triangleq \frac{1}{m} \times \sum_{i=1}^{m} \begin{cases} \frac{\widetilde{\aleph}_i^{\widetilde{k}}(t)}{\max\limits_{\forall \rho_i \in \mathcal{P}} \aleph_i(t)} + \frac{\phi^l(t)}{\widetilde{\phi}_i^{\widetilde{k}}(t)} & \widetilde{\aleph}_i^{\widetilde{k}}(t) > 0, \\ \frac{\widetilde{\aleph}_i^{\widetilde{k}}(t)}{\min\limits_{\forall \rho_i \in \mathcal{P}} \aleph_i(t)} + \frac{\widetilde{\phi}_i^{\widetilde{k}}(t)}{\phi^u(t)} & \widetilde{\aleph}_i^{\widetilde{k}}(t) < 0, \end{cases} \tag{32}$$

where $\widetilde{\Upsilon}_{\mathcal{A}}(t)$ defines actions that the aggregator makes regarding behavior pair of prosumers. This matrix is produced once at each time interval. Fig. 5 shows how *behavior matrices* $\widetilde{\mathcal{BM}}_{\mathcal{A}}^{\widetilde{k}}(t)$, $1 \leq \widetilde{k} \leq \widetilde{K}$, according to (24), are generated randomly. $\widetilde{\mathcal{BM}}_{\mathcal{A}}^{\widetilde{k}}(t)$ consists of $\widetilde{\mathcal{BP}}_i^{\widetilde{k}}(t), \forall \rho_i \in \mathcal{P}$, each of which is an action that the aggreagtor makes in response to corresponding behavior pair of prosumers. $\widetilde{\Upsilon}_{\mathcal{A}}(t)$ is finally sorted descendingly by the metric of satisfaction index $\mathcal{SI}_{\mathcal{A}}^{\widetilde{k}}(t) \in (0, 2]$ determining to which extent the aggregator is satisfied with $\widetilde{\mathcal{BM}}_{\mathcal{A}}^{\widetilde{k}}(t)$. This is done by calculating how much each $\widetilde{\mathcal{BP}}_i^{\widetilde{k}}(t), \forall \rho_i \in \mathcal{P}$ depending on the status of prosumer, uses the *maximum* (minimum) available flexibility and provides a more beneficial price offer.

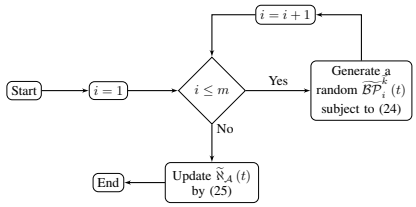


Fig. 5. Flow chart of generating a feasible behavior matrix $\widetilde{\mathcal{BM}}_{\mathcal{A}}^{\widetilde{k}}(t)$.

The following section explains how the employed negotiation approach, to reach an agreement acceptable to all, benefits from behavior pairs and matrices.

IV. Negotiation Approach

Prosumers' rational behaviors are more pronounced when their uncertainty about the decision space of the aggregator increases. Due to the promising outlook of introducing prosumers into the smart grid, this dissertation employs an approach to enable the concurrent negotiation on *power* and *price* issues with *packaged offers* given that the negotiators have no prior knowledge about the flexibility information and utility functions of each other [7]. To model such approach, the following key elements are needed: i) notion of a solution to the negotiation problem, and ii) negotiation protocol and strategy. The negotiation protocol and strategy define how negotiators provide and prepare offers, respectively.

A. Negotiators

Fig. 6 depicts the behavior of work-flow executions in the framework. The negotiation procedure is conducted between time intervals t and $t + \Delta t$ for maximum $\mathcal{T} \in \mathbb{N}$ iterations (set arbitrarily). The following part defines the negotiators.

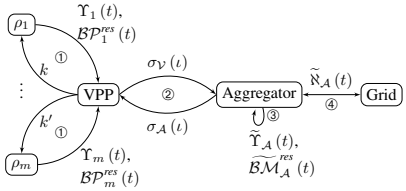


Fig. 6. Data flow diagram of the framework between each t and $t + \Delta t$. Procedures ①, ③, and ④ are done only once while ② takes maximum $\mathcal{T}$ iterations. Notations are described in Sections IV-A and IV-B.

1) Virtual Power Plant: This paper, to alleviate the challenges of m parallel bilateral negotiations between prosumers and the aggregator, where each negotiation in the worst case to reach an agreement can take $\mathcal{T}$ iterations, utilizes an intermediate VPP to negotiate, on behalf of prosumers. Let

$$\sigma_{\mathcal{V}}(\iota) = \begin{bmatrix} \mathcal{BP}_1^k(t) \\ \vdots \\ \mathcal{BP}_m^{k'}(t) \end{bmatrix}, \tag{33}$$

where $\sigma_{\mathcal{V}}(\iota)$ is an *offer package* sent from the VPP to the aggregator at negotiation iteration $1 \leq \iota \leq \mathcal{T}$. Behavior pairs $\mathcal{BP}_1^k(t)$ and $\mathcal{BP}_m^{k'}(t)$ point to rows k and k' (not necessarily equivalent) in $\Upsilon_1(t)$ and $\Upsilon_m(t)$, respectively (see (26)). Let

$$\psi_{\mathcal{V}}(\mathcal{SI}_{\mathcal{V}}(\iota)) = 1 - \frac{\sum_{i=1}^{m} \left(\frac{1}{2} \times \mathcal{SI}_i^k(t)\right)^2}{m}, \tag{34}$$

$$\mathcal{SI}_{\mathcal{V}}(\iota) \triangleq \bigcup_{i=1}^{m} \mathcal{SI}_i^k(t), 1 \leq k \leq K, \tag{35}$$

where $\psi_{\mathcal{V}} \in [0, 1)$ is the VPP's utility function. We assume a very general hyper-quadric utility function [34] for negotiators, which is private, continuous, and strictly concave [7]. By "private," negotiators have no knowledge about other negotiator's utility function. $\mathcal{SI}_{\mathcal{V}}(\iota)$ is the *union* of satisfaction indexes of prosumers' behavior pairs. Superscripts k in $\mathcal{SI}_i^k(t), \forall \rho_i \in \mathcal{P}$ are not necessarily equivalent. Let

$$\sigma_{\mathcal{V}}^{res}(t) = \begin{bmatrix} \mathcal{BP}_1^{res}(t) \\ \vdots \\ \mathcal{BP}_m^{res}(t) \end{bmatrix}, \tag{36}$$

$$\mathcal{BP}_i^{res}(t) = \left(\aleph_i^{res}(t), \phi_i^{res}(t)\right), \tag{37}$$

$$\aleph_i^{res}(t) = \begin{cases} \underline{\aleph}_i(t) & \sum_{\substack{flex_{j,i}(t)=0, \\ \forall a_{j,i} \in \mathcal{AP}_i}} \tau_{j,i}(t) \leq PV_i^g(t), \\ \overline{\aleph}_i(t) & \text{otherwise}, \end{cases} \tag{38}$$

$$\phi_i^{res}(t) = \begin{cases} \phi^l(t) & \aleph_i^{res}(t) > 0, \\ \phi^u(t) & \aleph_i^{res}(t) < 0, \end{cases} \tag{39}$$

where $\sigma_{\mathcal{V}}^{res}(t)$ is the reservation offer package of the VPP including prosumers' reservation behavior pair $\mathcal{BP}_i^{res}(t)$. $\aleph_i^{res}(t)$ and $\phi_i^{res}(t)$ are the reservation power and price offers of prosumer ρ_i at time interval t. Prosumers in the worst case have to: i) satisfy appliances with no flexibility remained, and ii) utilize the electric power generated by the PV completely. The reservation price offer equals either the lowest ($\aleph_i^{res}(t) > 0$) or highest ($\aleph_i^{res}(t) < 0$) possible offerable electricity price, respectively. $\mathcal{SI}_{\mathcal{V}}^{res}(t)$, as the satisfaction index of $\sigma_{\mathcal{V}}^{res}(t)$, is the union of $\mathcal{SI}_i^{res}(t) \in (0, 2]$ (calculated by (28)) associated with $\mathcal{BP}_i^{res}(t)$. Any offer package with the utility value less than $\psi_{\mathcal{V}}\left(\mathcal{SI}_{\mathcal{V}}^{res}(t)\right)$ is unacceptable to VPP. The VPP in the end of negotiation returns indexes of agreed behavior pairs, i.e., $k \leq K, \forall \rho_i \in \mathcal{P}$, to the prosumers (see Fig. 6).

2) Aggregator: Let

$$\sigma_{\mathcal{A}}(\iota) = \widetilde{\mathcal{BM}}_{\mathcal{A}}^{\bar{k}}(t), \tag{40}$$

where $\sigma_{\mathcal{A}}(\iota)$, equivalent to a behavior matrix in $\widetilde{\Upsilon}_{\mathcal{A}}(t)$, is an offer package sent from the aggregator to the VPP. Let

$$\psi_{\mathcal{A}}\left(\mathcal{SI}_{\mathcal{A}}^{\bar{k}}(t)\right) = 1 - \left(\frac{1}{2} \times \mathcal{SI}_{\mathcal{A}}^{\bar{k}}(t)\right)^2, \tag{41}$$

where $\psi_{\mathcal{A}} \in [0, 1)$ is the aggregator's utility function, which follows the same rule as the VPP does [7]. $\mathcal{SI}_{\mathcal{A}}^{\bar{k}}(t)$ is the satisfaction index of behavior matrix $\mathcal{BM}_{\mathcal{A}}^{\bar{k}}(t)$. Let

$$\sigma_{\mathcal{A}}^{res}(t) = \begin{bmatrix} \widetilde{\mathcal{BP}}_1^{res}(t) \\ \vdots \\ \widetilde{\mathcal{BP}}_m^{res}(t) \end{bmatrix}, \tag{42}$$

$$\widetilde{\mathcal{BP}}_i^{res}(t) = \left(\min_{\forall \rho_i \in \mathcal{P}} \underline{\aleph}_i(t), \phi^l(t)\right), \tag{43}$$

where $\sigma_{\mathcal{A}}^{res}(t)$ is the reservation offer package of the aggregator denoting $m \times \min_{\forall \rho_i \in \mathcal{P}} \underline{\aleph}_i(t)$ amount of electric power must be exchanged (in the worst case) with the grid for $\phi^l(t)$. This reservation offer package is coupled with a satisfaction index $\mathcal{SI}_{\mathcal{A}}^{res}(t) \in (0, 2]$ (calculated by (32)). Similar to the VPP, the aggregator will not accept any offer package with the utility value less than $\psi_{\mathcal{A}}\left(\mathcal{SI}_{\mathcal{A}}^{res}(t)\right)$. The following part explains the protocol and strategy the negotiators follow during the negotiation process.

B. Negotiation Protocol and Strategy

We employ an *alternating-offer protocol* [35], where the VPP produces an offer and the aggregator either accepts it or produces a new one. The negotiation begins with offer packages produced with the highest possible utility values and continues with offer packages with lower utility values. It terminates when: i) an offer on the table is acceptable to both negotiators, or ii) it reaches iteration $\mathcal{T}$ with no offer accepted. Let

$$\sigma_{\mathcal{V}}(1) = \begin{bmatrix} \mathcal{BP}_1^1(t) \\ \vdots \\ \mathcal{BP}_m^1(t) \end{bmatrix}, \tag{44}$$

$$\sigma_{\mathcal{A}}(1) = \widetilde{\mathcal{BM}}_{\mathcal{A}}^1(t), \tag{45}$$

$$\psi_{\mathcal{V}}\left(\mathcal{SI}_{\mathcal{V}}(\iota-1)\right) \leq \psi_{\mathcal{V}}\left(\mathcal{SI}_{\mathcal{V}}(\iota)\right) \leq \psi_{\mathcal{V}}\left(\mathcal{SI}_{\mathcal{V}}^{res}(t)\right), \tag{46}$$

$$\psi_{\mathcal{A}}\left(\mathcal{SI}_{\mathcal{A}}(\iota-1)\right) \leq \psi_{\mathcal{A}}\left(\mathcal{SI}_{\mathcal{A}}(\iota)\right) \leq \psi_{\mathcal{A}}\left(\mathcal{SI}_{\mathcal{A}}^{res}(t)\right), \tag{47}$$

where $\sigma_{\mathcal{V}}(1)$ and $\sigma_{\mathcal{A}}(1)$ are initial preferred offer packages of the VPP and aggregator, respectively. Since $\Upsilon_i(t), \forall \rho_i \in \mathcal{P}$ and $\widetilde{\Upsilon}_{\mathcal{A}}(t)$ are sorted descendingly, the initial offer packages provide the highest utility value. Negotiators, over negotiation iterations, produce offer packages with lower utility values. They neither propose nor accept any offer package with utility value lower than their reservation utility value. To propose a new offer package, they follow the following two consecutive procedures:

1) Reactive Utility Value Concession: The negotiation approach assumes each negotiator's utility value obtained by an agreement is higher than the one with no agreement. Therefore, they prefer to concede over risking negotiation breakdown. Let

$$\varsigma_{\mathcal{V}}(\iota) = \psi_{\mathcal{V}}\left(\sigma_{\mathcal{V}}(1)\right) - \sigma_{\mathcal{V}}^{res}(t) \times \left(\frac{\iota}{\mathcal{T}}\right)^{\frac{1}{\epsilon}}, \tag{48}$$

$$\varsigma_{\mathcal{A}}(\iota) = \psi_{\mathcal{A}}\left(\sigma_{\mathcal{A}}^1(t)\right) - \sigma_{\mathcal{A}}^{res}(t) \times \left(\frac{\iota}{\mathcal{T}}\right)^{\frac{1}{\epsilon}}, \tag{49}$$

where $\varsigma_{\mathcal{V}}(\iota), \varsigma_{\mathcal{A}}(\iota) \in [0, 1)$ are monotonically decreasing time-dependent concession values of the VPP and aggregator, respectively [7]. Their values only depend on each negotiator's reservation utility value and the number of negotiation iterations passed so far with the decay rate $\epsilon \in \mathbb{R}_{>0}$ [36]. As the second assumption, negotiators are assumed to be reactive. Hence, their concession rate should depend on their perception of the utility value of other party's offer packages given: i) whether the current offer of the opponent negotiator provides higher utility value than the negotiator's reservation utility value, and ii) the negotiator's perception of how much the other party has conceded. One reason for a negotiator to stop decreasing its desired utility value over time is to gain higher utility. This happens if the other negotiator, without realizing that the negotiator has stopped conceding, accepts time-dependent concession values at all negotiation iterations. This behavior is called the "deliberate stopping of concession." As a result, Let

$$\omega_{\mathcal{V}}(\iota) = \left(\psi_{\mathcal{V}}\left(\mathcal{SI}_{\mathcal{V}}^{temp}\right) - \psi_{\mathcal{V}}\left(\mathcal{SI}_{\mathcal{V}}^{temp'}\right)\right)^+, \tag{50}$$

$$\omega_{\mathcal{A}}(\iota) = \left(\psi_{\mathcal{A}}\left(\mathcal{SI}_{\mathcal{A}}^{temp}\right) - \psi_{\mathcal{A}}\left(\mathcal{SI}_{\mathcal{A}}^{temp'}\right)\right)^+, \tag{51}$$

where $\omega_{\mathcal{V}}(\iota), \omega_{\mathcal{A}}(\iota) \in [0, 1)$ are reactive concession values of the VPP and aggregator, respectively, and $y^+ = \max\{0, y\}$.

The VPP, using (28), calculates $\mathcal{SI}_{\mathcal{V}}^{temp}$ and $\mathcal{SI}_{\mathcal{V}}^{temp'}$ for $\sigma_{\mathcal{A}}(\iota)$ and $\sigma_{\mathcal{A}}(\iota-1)$, respectively. The aggregator, by using (32), follows a similar procedure. Then, let

$$\Pi_{\mathcal{V}}(\iota) = \min\{\varsigma_{\mathcal{V}}(\iota), \Pi_{\mathcal{V}}(\iota-1) - \omega_{\mathcal{V}}(\iota)\}, \quad (52)$$

$$\Pi_{\mathcal{A}}(\iota) = \min\{\varsigma_{\mathcal{A}}(\iota), \Pi_{\mathcal{A}}(\iota-1) - \omega_{\mathcal{A}}(\iota)\}, \quad (53)$$

where $\Pi_{\mathcal{V}}(\iota), \Pi_{\mathcal{A}}(\iota) \in [0,1)$ are desired utility values of the VPP and the aggregator at iteration ι, respectively. Negotiators only accept an offer package that provides a utility value equivalent to or higher than their desired utility value at that iteration.

2) New Offer Package Generation: Let us assume $Z_{\mathcal{V}}(t)$ (including maximum K^m offer packages, see (26)) and $Z_{\mathcal{A}}(t)$ (including maximum $\widetilde{K}$ possible feasible offer packages, see (29)) are the convex *feasible offer package sets* of the VPP and the aggregator, respectively. These offer packages provide negotiators with utility value equivalent to or no less than their reservation offer package's utility value. For an agreement to exist, let $\mathscr{Z}(t) = Z_{\mathcal{V}}(t) \bigcap Z_{\mathcal{A}}(t) \neq \varnothing, \forall t$ remain unchanged during the negotiation, where $\mathscr{Z}(t)$ is the zone of agreement denoting the common intersection of the feasible offer package sets. If an offer package is within $\mathscr{Z}(t)$, a negotiator may not accept it if it yields a utility value lower than the negotiator's *current* desired utility value. To make an acceptable agreement, negotiators keep conceding to their reservation utility values subject to the nonempty zone of agreement at each time interval. Thus, geometrically speaking, in negotiation, the negotiators' goal is to find a point in the zone of agreement, under the restriction that this zone is unknown to negotiators and none of them has any explicit knowledge about each other's utility functions [7].

Let ι be the negotiation iteration when it is the VPP's turn to produce a new offer package. Let $\mathcal{BP}_i^k(t) \in \sigma_{\mathcal{V}}(\iota-1)$. The VPP (temporarily) updates $\sigma_{\mathcal{V}}(\iota-1)$ with behavior pairs $\mathcal{BP}_i^{k'}(t), \forall k+1 \leq k' \leq K$ and expands $\overline{Z}_{\mathcal{V}}(\iota)$ with the updated offer packages *individually only if* each returns a utility value equivalent to $\Pi_{\mathcal{V}}(\iota)$. $\overline{Z}_{\mathcal{V}}(\iota) \subseteq Z_{\mathcal{V}}(t), \forall \iota \leq \mathcal{T}$ is the *continuously expanding* feasible offer package subset of the VPP. The aggregator at iteration $\iota+1$ determines $\widetilde{k}$ where $\widetilde{\mathcal{BM}}_{\mathcal{A}}^{\widetilde{k}}(t) \in \widetilde{\Upsilon}_{\mathcal{A}}(t)$. Then, it updates $\overline{Z}_{\mathcal{A}}(\iota+1)$ with new offer packages $\widetilde{\mathcal{BM}}_{\mathcal{A}}^{\widetilde{k}+1}(t), \forall \widetilde{k}+1 \leq \widetilde{K}$, where each provides the aggregator with a utility value equivalent to $\Pi_{\mathcal{A}}(\iota+1)$. $\overline{Z}_{\mathcal{A}}(\iota) \subseteq Z_{\mathcal{A}}(t), \forall \iota \leq \mathcal{T}$ is the continuously expanding feasible offer package subset of the aggregator. Let

$$\sigma_{\mathcal{V}}(\iota) = P_{\overline{Z}_{\mathcal{V}}(\iota)}[\chi(\iota)] = \underset{q \in \overline{Z}_{\mathcal{V}}(\iota)}{\arg\min} \|q - \chi(\iota)\|, \quad (54)$$

$$\sigma_{\mathcal{A}}(\iota) = P_{\overline{Z}_{\mathcal{A}}(\iota)}[\chi(\iota)] = \underset{q \in \overline{Z}_{\mathcal{A}}(\iota)}{\arg\min} \|q - \chi(\iota)\|, \quad (55)$$

$$\chi(\iota) = \varphi_{\mathcal{V}}(\iota) \times \sigma_{\mathcal{V}}(\iota-1) + \varphi_{\mathcal{A}}(\iota) \times \sigma_{\mathcal{A}}(\iota-1), \quad (56)$$

$$\varphi_{\mathcal{V}}(\iota) + \varphi_{\mathcal{A}}(\iota) = 1, \quad (57)$$

where P is the operator of projecting the weighted offer package $\chi(\iota)$, created based on the latest offers made by all agents, on current continuously expanding feasible offer package subsets $\overline{Z}_{\mathcal{V}}(\iota)$ and $\overline{Z}_{\mathcal{A}}(\iota)$ [37]. $\arg\min\|\cdot\|$ is the Frobenius norm with argument of minimum. Note this method generates an offer that is acceptable to the negotiator and is closest (in terms of Euclidean distance) to the weighted offer package $\chi(\iota)$. $\varphi_{\mathcal{V}}(\iota), \varphi_{\mathcal{A}}(\iota) \in (0,1)$ are the weights that each negotiator puts on the other's offer package.

In Algorithms 1 and 2, we provide the pseudo-code for the overall communication steps and the negotiation approach, respectively. Steps in the former algorithm are in line with the data flow diagram depicted in Fig. 6. Fig. 7 illustrates a conceptual example of the offer package space during the negotiation and shows how the VPP and the aggregator negotiate with each other over, for example, $\mathcal{T} = 9$ iterations [7]. Offer packages existing on each concession curve have equal utility values. The negotiation terminates when $\max\{\|\sigma_{\mathcal{V}}(9) - \chi(9)\|, \|\sigma_{\mathcal{A}}(9) - \chi(9)\|\} < \delta$, which denotes that if the Euclidean distances between the current iteration's offer packages and the weighted offer package are less than a constant convergence tolerance $\delta \in \mathbb{R}_{>0}$.

Algorithm 1: Communication steps in the framework between time intervals t and $t+\Delta t$

```
// Prosumers' part;
foreach ρ_i ∈ P do
    Run the NSGA-III to produce Υ_i(t);
    Determine the reservation behavior pair BP_i^res(t);
end
// VPP's part (i);
Determine the reservation offer package σ_V^res(t);
Produce the first offer package σ_V(1);
// Aggregator's part;
Run the NSGA-III to produce Υ̃_A(t);
Determine the reservation offer package σ_A^res(t);
Produce the first offer package σ_A(1);
// Negotiation approach;
Run Algorithm 2 // VPP's part (ii);
Return the indexes of agreed behavior pairs to prosumers;
```

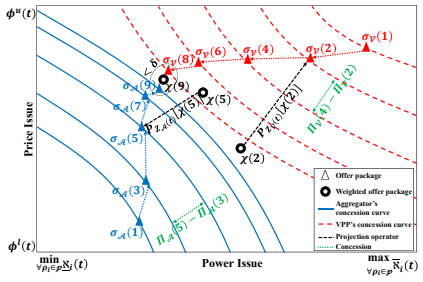


Fig. 7. Conceptual example of the offer package space during the negotiation.

Algorithm 2: The negotiation approach

IsConverge=False;
$\iota = 2$;
while $\iota \leq \mathcal{T}$ *and IsConverge=False* **do**
  Determine the negotiator's turn by $S = \mathrm{mod}\,(\iota, 2)$;
  if $S = 0$ **then** // VPP's turn
    $\sigma_{\mathcal{A}}(\iota) = \sigma_{\mathcal{A}}(\iota - 1)$;
    $\Pi_{\mathcal{A}}(\iota) = \Pi_{\mathcal{A}}(\iota - 1)$;
    Calculate $\Pi_{\mathcal{V}}(\iota)$ by (52);
    for $i = 1$ *to* m **do**
      Determine k, where $\mathcal{BP}_i^k(t) \in \sigma_{\mathcal{V}}(\iota - 1)$;
      $\sigma_{\mathcal{V}}^{temp} = \sigma_{\mathcal{V}}(\iota - 1)$;
      Found=0;
      while $k + 1 \leq K$ **do**
        Update $\sigma_{\mathcal{V}}^{temp}$ with $\mathcal{BP}_i^{k+1}(t)$;
        Calculate $\mathcal{SI}_{\mathcal{V}}^{temp}$ for $\sigma_{\mathcal{V}}^{temp}$ by (35);
        if $\psi_{\mathcal{V}}\left(\mathcal{SI}_{\mathcal{V}}^{temp}\right) = \Pi_{\mathcal{V}}(\iota)$ **then**
          Add $\sigma_{\mathcal{V}}^{temp}$ to $\overline{Z}_{\mathcal{V}}(\iota)$;
          Found=1;
          $k = k + 1$;
        else if *Found=1* **then**
          $k = K$;
        else
          $k = k + 1$;
        end
      end
    end
    Set $\sigma_{\mathcal{V}}(\iota)$ by (54);
  else // Aggregator's turn
    $\sigma_{\mathcal{V}}(\iota) = \sigma_{\mathcal{V}}(\iota - 1)$;
    $\Pi_{\mathcal{V}}(\iota) = \Pi_{\mathcal{V}}(\iota - 1)$;
    Set $\Pi_{\mathcal{A}}(\iota)$ by (53);
    Determine $\widetilde{k}$, where $\widetilde{\mathcal{BM}}_{\mathcal{A}}^{\widetilde{k}} \in \widetilde{\Upsilon}(t)$;
    Found=0;
    while $\widetilde{k} + 1 \leq \widetilde{K}$ **do**
      if $\mathcal{SI}_{\mathcal{A}}^{\widetilde{k}+1}(t) = \Pi_{\mathcal{A}}(\iota)$ **then**
        Add $\sigma_{\mathcal{A}}^{\widetilde{k}+1}$ to $\overline{Z}_{\mathcal{A}}(\iota)$;
        Found=1;
        $\widetilde{k} = \widetilde{k} + 1$;
      else if *Found=1* **then**
        $\widetilde{k} = \widetilde{K}$;
      else
        $\widetilde{k} = \widetilde{k} + 1$;
      end
    end
    Set $\sigma_{\mathcal{A}}(\iota)$ by (55);
  end
  Set $\chi(\iota)$ by (56);
  if $\max\{\|\sigma_{\mathcal{V}}(\iota) - \chi(\iota)\|, \|\sigma_{\mathcal{A}}(\iota) - \chi(\iota)\|\} < \delta$ **then**
    IsConverge=True;
  else
    $\iota = \iota + 1$;
  end
end

C. Solution Concept for the Negotiation Approach

The use of the solution concept in this paper, where the negotiators have no information about their opponents, is in the spirit of Herbert Simon [38]. Through computational experiments, The authors in [7] have demonstrated that such solution concept proposed in the negotiation approach yields a performance sufficiently close to the Nash bargaining solution, which is a different definition proposed for a proper negotiation solution [39]. The set of points that satisfy Nash bargaining solution's requirements are all subsets of the zone of agreement. However, computing them requires that all the negotiators have complete knowledge of the preference structure and utility function of the opponents.

The authors in [7] have also analytically proved that: i) the scale of the utility value of each negotiator is of no critical importance, as long as the reservation utility value and the scale of concession are consistent with it, ii) the negotiators, by utilizing the utility value concession strategy described earlier, converge to an agreement acceptable to all in maximum $\mathcal{T}$ iterations, if the zone of agreement is nonempty and they concede to reservation utility values in the worst case, and iii) the convergence holds for general concave utility functions as long as all the negotiators concede to their reservation utilities, irrespective of the specific concession strategy they adopt.

V. Simulation Setup and Analysis

This section presents the simulation results and evaluates the performance of the proposed framework, which has been implemented in Matlab® R2017a running with 16 Intel 2.3 GHz Xeon® E5-2686 CPUs and 64 GB memory.

A. Assessment Metrics for Performance Evaluation

To verify the effectiveness of the framework, we define the following performance evaluation metrics:

1) Environmental Metrics: They are designed to assess the environmental impact of the proposed framework on the power grid. The impact is primarily related to using prosumers' flexibility capabilities.

a) Peak Demand Reduction: It determines how much the proposed framework succeeds in shaving the peak demand. Let

$$\mathrm{PDR} \triangleq \left(1 - \frac{\max\limits_{\forall t} \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n_i} \tau_{j,i}(t) \times dec_{j,i}(t)}{\max\limits_{\forall t'} \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n_i} \tau_{j,i}(t')}\right) \times 100, \quad (58)$$

where t and t' are time intervals, at which the grid confronts the maximum peak demand *with* and *without* using the proposed framework, respectively. PDR will be zero, if $t = t'$.

b) Peak-to-Average Ratio: It measures how much higher the peak demand is than average demands over a single simulation. A high PAR means a large fluctuation in daily

load demand. Let

$$\text{PAR} \triangleq \frac{\max\limits_{\forall t \leq T} \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n_i} \tau_{j,i}(t) \times dec_{j,i}(t)}{\sum\limits_{t=1}^{T} \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n_i} \tau_{j,i}(t) \Big/ T}. \quad (59)$$

2) Economic and Profit Metrics: It allow the analysis of economic performance and impact of the proposed framework on prosumers' daily life (in terms of load consumption). Their main purpose is to provide the quantitative information needed to make a judgment on real deployment of the framework.

a) Average Appliance Operation Delay: It calculates the delay in delivering appliances in the completed status. Let

$$\text{AOD} \triangleq \frac{\sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n_i} (\varepsilon_{j,i} - \beta_{j,i})}{\sum\limits_{i=1}^{m} n_i}. \quad (60)$$

b) Average Flexibility Usage Rate: It considers how much of prosumers' flexibilities are traded in the market. Let

$$\text{FUR} \triangleq \frac{\sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n_i} \frac{\varepsilon_{j,i} - \beta_{j,i}}{\theta_{j,i} - \alpha_{j,i}}}{\sum\limits_{i=1}^{m} n_i} \times 100. \quad (61)$$

c) Average Prosumer Cost-Benefit: It evaluates the cost-effectiveness of the framework for prosumers. It studies how much money they averagely earn/spend *with* and *without* negotiating and exchanging power with the grid. Let

$$\text{PCB} \triangleq \left(1 - \sum_{t=1}^{T} \sum_{i=1}^{m} \frac{\aleph_i^k(t) \times \phi_i^k(t)}{\sum\limits_{j=1}^{n_i} \tau_{j,i}(t) \times \phi_{\mathcal{G}}^u(t)}\right) \times 100, \quad (62)$$

where k for each prosumer is the behavior pair index, on which the VPP in the end of negotiation process at time interval t has agreed with the aggregator. $T \in \mathbb{N}$ is the last simulated time interval.

d) Average Self Load-Satisfaction Rate: It studies the local energy utilization for prosumers. Let

$$\text{SLR} \triangleq \frac{\sum\limits_{t=1}^{T} \sum\limits_{i=1}^{m} \frac{\aleph_i^{\text{P2L}}(t) + \aleph_i^{\text{B2L}}(t)}{\sum\limits_{j=1}^{n_i} \tau_{j,i}(t) + PV_i^g(t) + B_i^{\overrightarrow{d}}}}{m \times T} \times 100. \quad (63)$$

e) Average Self Sufficiency Rate: It evaluates PVs' capability in maximizing the comfort level of prosumers without purchasing any amount of power from the grid. Let

$$\text{SSR} \triangleq \frac{\sum\limits_{t=1}^{T} \sum\limits_{i=1}^{m} \frac{\aleph_i^{\text{P2L}}(t) + \aleph_i^{\text{P2B}}(t)}{\sum\limits_{j=1}^{n_i} \tau_{j,i}(t) + PV_i^g(t)}}{m \times T} \times 100. \quad (64)$$

3) Computation Time: Measuring the CPU time of different parts of the framework is to quantify the overall busyness of the system. This is the time taken from the start until the end of a specific part as measured by an ordinary clock. This metric measures the computation time of the NSGA-III algorithm, negotiation approach, and total simulation.

B. Simulation Data

Table I lists the inputs to simulations, which are assumed constant unless otherwise stated. For the PV generation profile, the real data captured from the UCLA Ackerman Union is scaled down from the capacity of 35 kW to 7 kW [27]. Real-time hourly electricity prices are captured from Nord Pool Spot [40], where $\{\phi^l(t), \phi^u(t)\}, \forall t$ and $\{\phi_{\mathcal{G}}^l(t), \phi_{\mathcal{G}}^u(t)\}, \forall t$ are adjusted by fluctuation rates of $\pm 50\%$ and $\pm 20\%$, respectively [21]. Table II describes how consumption scenarios for appliances are created. Start, end, and deadline flexibility times are randomly generated by the normal distribution $\mathcal{N}(\mu, \sigma^2)$ with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$. Load profiles of appliances are captured from [41], [42] with the time resolution of $\Delta t = 1$ hour. Refrigerator operates uninterruptedly with no end and flexibility times. Nissan Altra is chosen as the electric vehicle with an empty battery at arrival and fully charged battery at departure [42]. The deadline flexibility concept from the perspective of the air conditioner is the *comfortable temperature range* [41], where 25°C and ±3°C are prosumers' desired temperature set point and flexibility, respectively.

TABLE I. CONSTANT INPUT VALUES FOR THE SIMULATIONS

Parameter	Value	Parameter	Value
Δt	1 hour	$^*PV_i^{cap}$	7 kW
$^*B_i^{\overline{\text{SOC}}}$	0.8	$^*B_i^{\underline{\text{SOC}}}$	0.2
$^*B_i^{cap}$	13.2 kWh	$^*B_i^{\overrightarrow{c}}$	5 kW
$^*B_i^{\overrightarrow{d}}$	5 kW	$\mathcal{T}$	100
ϵ	0.8	δ	0.01
$^\dagger\varphi_{\mathcal{V}}(\iota)$	0.5	$^\dagger\varphi_{\mathcal{A}}(\iota)$	0.5

$^*\forall \rho_i \in \mathcal{P}$, $^\dagger\forall \iota \leq \mathcal{T}$.

TABLE II. TIMETABLE OF GENERATING LOAD DEMAND SCENARIOS OF APPLIANCES

	$a_{j,i} \in \mathcal{AP}_i, \forall \rho_i \in \mathcal{P}$	$\alpha_{j,i}$	$\beta_{j,i}$	$\theta_{j,i}$
$\mathcal{NS}_i$	Refrigerator (RG)	00:00	N/A	N/A
$\mathcal{S}_i$	Washing Machine (WM)	$\mathcal{N}(10,3)$	$\alpha_{j,i} + 02{:}00$	$\mathcal{N}(16,4)$
	Laundry Dryer (LD)	$\mathcal{N}(15,1)$	$\alpha_{j,i} + 01{:}30$	$\mathcal{N}(21,5)$
	Dishwasher (DW)	$\mathcal{N}(17,2)$	$\alpha_{j,i} + 01{:}40$	$\mathcal{N}(23,2)$
	Electric Vehicle (EV)	$\mathcal{N}(19,10)$	$\alpha_{j,i} + 05{:}00$	$^*\mathcal{N}(7.5,1)$
	Air Conditioner (AC)	$\mathcal{N}(9,1)$	$\mathcal{N}(21,2)$	25°C ± 3°C

*The next day.

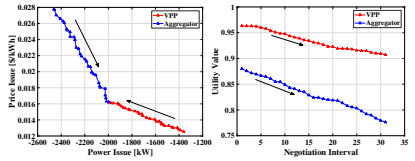
C. Analysis and Discussion

The following four parts evaluate and analyze the proposed framework and the negotiation algorithm from different perspectives. This section is concluded with evaluating the performance of the proposed framework compared with existing frameworks discussed in the related work. All statistical results have been averaged over 100 independent simulation runs.

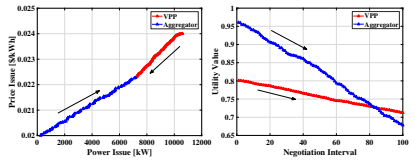
1) Impact of Status of Prosumers on the Negotiation Process: Fig. 8 shows the offer package and utility value (unitless, see (34) and (41)) concession spaces of randomly picked time interval in different circumstances. In Fig. 8(a), no PVs and BESSs are considered. The VPP, for example at negotiation iteration $\iota = 15$, is interested in buying 1630 kW of electric power for 0.0145 \$/kWh. The aggregator, at negotiation iteration $\iota = 16$, rejects this offer and makes a new one intending to sell 2180 kW of electric power for 0.022 \$/kWh. They continue negotiating until iteration $\iota = 31$, at which they come to an agreement on exchanging 2000 kW of electric power for 0.016 \$/kWh. Fig. 8(b) shows the negotiation process, where all prosumers own PV and BESS. They reach an agreement after exactly 100 negotiation iterations. Having the same utility value of 0.76 at negotiation iteration $\iota = 65$ does not terminate the process since the VPP provides an offer package with selling 8385 kW of electric power for 0.0227 \$/kWh while the aggregator returns another offer package with buying 4738 kW of electric power for 0.0212 \$/kWh. Fig. 8(c) experiences the same setting as Fig. 8(b) does, where negotiators reach an agreement after 71 negotiation iterations. Reasons for having a mixed number of buyer and seller prosumers at this interval are the absence of PV generation (outside of the PV generation period), presence of BESSs with average SOC value of 0.48, and having all refrigerators, 23 dishwashers, 12 newly arrived electric vehicles, and all air conditioners in operation. Therefore, the VPP has to averagely increase the amount of power to sell and decrease the price offer while the aggregator behaves the other way around.

2) Impact of Penetration of Prosumers on the Negotiation Process and the Grid: Fig. 9 demonstrates how increasing the number of prosumers influences the computation time and negotiation convergence iteration. Running an individual instance of NSGA-III for each prosumer and the aggregator, where population size and number of generations equal 100, takes approximately 7 seconds. To evaluate the practicality of the negotiation approach employed in the proposed framework with $m = 900$ prosumers, we simulate two setups: i) parallel bilateral negotiations between prosumers and the aggregator (with no VPP), and ii) a single bilateral negotiation between the VPP and the aggregator (introduced here). In the former, CPU and memory usages are 79% and 42 GB, respectively, and reaching agreement at each time interval takes approximately 75 seconds. In the latter, these values for the gateway of each party (each prosumer, the VPP, and the aggregator) are 34.6% (of a single core CPU) and 960 MB, respectively, and the negotiation converges in approximately 39 seconds.

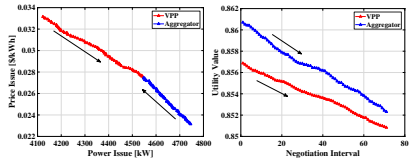
Table III evaluates the assessment metrics according to different penetration rates of prosumers. These assessment metrics, due to the presence of conflicting objectives in the framework, provide prosumers and the aggregator with trade-offs in making decisions. As the number of prosumers increases, mathematically speaking, the size of the convex feasible offer packages set $Z_{\mathcal{V}}(t), \forall t$ of the VPP also increases (including maximum K^m offer packages). This provides the VPP with more opportunities in utilizing prosumers' flexibilities, which enables it to: i) decrease the delay in satisfying load demands of appliances in average, ii) increase the Average Prosumer Cost-Benefit (PCB), and iii) increase the PDR. Increasing rates of SLR and SSR also depend on: i) the generation profiles of PVs in different weather conditions and the BESS capacities, as discussed in the next section, and ii) decrease in FUR.



(a) All prosumers are buyers.

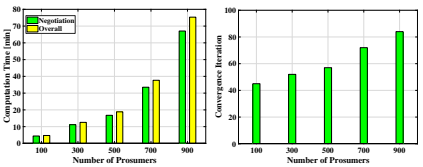


(b) All prosumers are sellers.



(c) Prosumers are mixed.

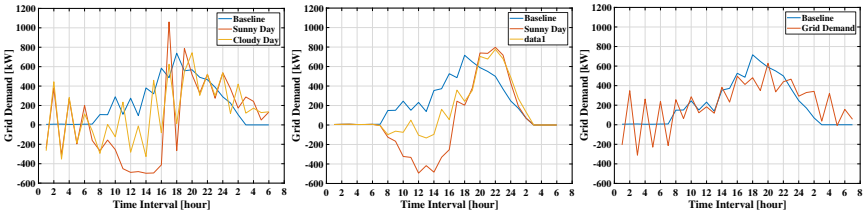
Fig. 8. Offer package (left) and utility value concession (right) spaces in different situations. Symbols in the offer package spaces, for the sake of simplicity, represent the average values of columns in the behavior matrices.



(a) Computation time. (b) Negotiation convergence.

Fig. 9. Total computation and average negotiation convergence times with different number of prosumers.

3) Impact of Penetration of PVs and BESSs on Prosumers and the Grid: Table IV evaluates to which extent the "random distribution of PVs and BESSs" impacts on the values of assessment metrics. Compared to the setting, where all pro-



(a) Full penetration of PVs and BESSs. (b) Full penetration of only PVs. (c) Full penetration of only BESSs.

Fig. 10. Grid demand with different penetration levels of PVs and BESSs in two sunny and cloudy days. Baseline shows load demands of appliances.

TABLE III. EVALUATING THE ASSESSMENT METRICS ACCORDING TO DIFFERENT PENETRATION RATES OF PROSUMERS IN THE GRID.

m	PDR (%)	PAR	AOD (hrs)	FUR (%)	PCB (%)	SLR (%)	SSR (%)
100	15.19	2.76	3.30	42.00	65.40	10.10	14.03
300	25.90	2.23	2.57	37.39	99.34	12.47	21.29
500	32.19	2.13	2.29	31.91	147.97	15.08	32.51
700	34.78	2.37	1.43	23.12	179.49	19.17	47.46
900	38.46	2.01	1.13	18.75	209.47	24.45	51.21

sumers own PVs and BESSs (see the first row in Table III), here, the grid experience lower PDR since the amount of flexibility is restricted. Decrease in AOD and FUR (due to limited flexibility) increases SLR and SSR (desire to increase the comfort), since the VPP cooperates with the aggregator to increase the PDR and PCB.

TABLE IV. EVALUATING THE ASSESSMENT METRICS ACCORDING TO THE PRESENCE OF PVS AND BESSS IN THE GRID.

$\forall \rho_i \in \mathcal{P}$	PDR (%)	PAR	AOD (hrs)	FUR (%)	PCB (%)	SLR (%)	SSR (%)
Only PV	11.43	2.95	1.56	28.18	46.73	20.16	20.16
Only BESS	33.29	2.08	1.20	11.67	35.46	0.83	0.00
Random	16.32	2.60	1.25	12.91	39.97	6.06	6.51

Table V analyzes the framework in different weather conditions. Obviously, fluctuations in the PV generation limits the VPP, in terms of available flexibility, in negotiation.

TABLE V. EVALUATING THE ASSESSMENT METRICS ACCORDING TO THE PV GENERATION PROFILE IN DIFFERENT WEATHER CONDITIONS.

Weather	PDR (%)	PAR	AOD (hrs)	FUR (%)	PCB (%)	SLR (%)	SSR (%)
Sunny	15.19	2.76	3.30	42.00	65.40	10.10	14.03
Cloudy	09.42	2.99	1.33	14.04	37.78	07.15	13.30

Fig. 10 shows grid demands for different PV and BESS penetration levels in two sunny and cloudy days. In Fig. 10(a), until the time at which PVs start the power generation (i.e., `07:00`), BESSs, by consecutive charging and discharging, try to regulate the grid demand. The grid confronts lower demand fluctuation, when there are only PVs in the system (see Fig. 10(b)). However, this setting results in lower PDR for the grid and PCB for prosumers. The reason is that prosumers, due to having no storage flexibility, are unable to provide the VPP with more flexibility. As Fig. 10(c) demonstrates, prosumers experience lower AOD and PCB. Similarly, the reason is the very limited amount of flexibility (only consumption flexibility). Table VI evaluates the assessment metrics and Fig. 11 shows the average SOCs according to the various BESS capacities. High BESS capacity provides prosumers with: i) more flexibility in storing energy, ii) lower AOD, and iii) higher PCB by selling more to the grid. The VPP, by such increase in the capacity, is able to provide the grid with more flexibility, which in turn, results in having higher PDR. BESSs with different capacities behave dissimilarly after PVs stop generating the electric power (see Fig. 11 for `20:00` to `07:00`-next day). The main reason is the arrival of the majority of electric vehicles, which impose higher load demands to the grid compared to other appliances.

TABLE VI. EVALUATING THE ASSESSMENT METRICS ACCORDING TO VARIOUS BESSS CAPACITIES.

$B_i^{cap}, \forall \rho_i \in \mathcal{P}$	PDR (%)	PAR	AOD (hrs)	FUR (%)	PCB (%)	SLR (%)	SSR (%)
13.2 kWh	15.19	2.76	3.30	42.00	65.40	10.10	14.03
26.4 kWh	24.36	2.38	2.00	34.68	102.72	14.84	14.66
39.6 kWh	39.62	1.98	0.58	25.36	165.57	18.12	14.34

Figs. 12 and 13 picture the generation profile and utilization distribution of a PV and a BESS, respectively. Considering Table II, the prosumer at `11:00`, `15:00`, and `19:00` endeavors to satisfy load demands of washing machine, laundry dryer, and dishwasher, respectively, with PV generation. At other hours, most of the PV generation is sold to the grid. These time intervals are also reflected in Fig. 13.

4) Impact of Consumption Flexibility of Prosumers on the Negotiation Process and the Grid: Table VII evaluates the assessment metrics based on different sets of appliances. A single refrigerator yields no PDR and delay due to its non-shiftability feature. Adding more shiftable appliances help prosumers provide the VPP with more consumption flexibilities. This increase has a direct correlation with the AOD and PCB, where prosumers benefit more while waiting for a longer time to receive their appliances in the completed status. Simulation results confirm that a shiftable appliance contributes to the PDR in the grid and the prosumer's PCB with averagely 0.1% and 0.37%, respectively.

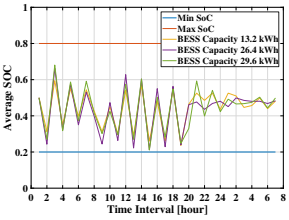


Fig. 11. Average SOC of BESSs with different capacities.

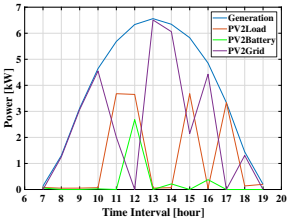


Fig. 12. Generation and utilization profiles of the PV of a prosumer.

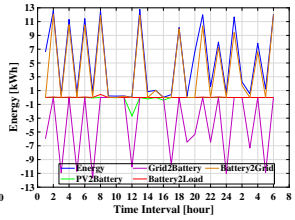


Fig. 13. Energy and utilization profiles of the BESS of a prosumer.

TABLE VII. EVALUATING THE ASSESSMENT METRICS ACCORDING TO DIFFERENT SETS OF APPLIANCES.

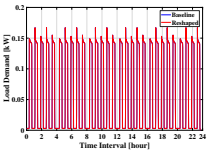
$\mathcal{AP}_i, \forall \rho_i \in \mathcal{P}$	PDR (%)	PAR	AOD (hrs)	FUR (%)	PCB (%)	SLR (%)	SSR (%)
{RG}	0	3.15	0.00	0.00	6.10	2.94	16.58
{RG, WM}	03.42	3.11	0.14	2.39	10.68	3.32	16.35
{RG, WM, LD}	09.22	3.01	0.58	13.19	26.98	6.20	15.39
{RG, WM, LD, DW}	11.63	2.92	1.37	20.90	34.33	8.06	15.14
{RG, WM, LD, DW, EV}	13.24	2.85	2.26	30.27	49.88	9.44	12.24
{RG, WM, LD, DW, EV, AC}	15.19	2.76	3.30	42.00	65.40	10.10	14.03

RG: Refrigerator, WM: Washing Machine, LD: Landry Dryer, DW: Dish Washer, EV: Electric Vehicle, and AC: Air Conditioner.

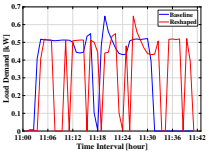
Fig. 14 demonstrates the baseline and reshaped load profiles of appliances of a prosumer (only hours in charge). Consecutive fluctuations in the baseline profile of the air conditioner is to keep the temperature constant at 25°C. Air conditioner starts using its temperature flexibility due to the load demand overlap between the laundry dryer (partly), dishwasher, and the electric vehicle. For example, the air conditioner between `19:20` and `19:40` attempts to increase the temperature since the laundry dryer has just finished operating and the operation of dishwasher has been interrupted.

Fig. 15 shows the hourly benefit/cost of prosumers with respect to real-time electricity prices. The baseline points to the case, where there are no PVs and BESSs simulated. One prosumer, for instance, to satisfy its load demands without any PV and BESS, has to daily spend (-)$2.59 while holding such equipment results in making a benefit of $2.78. Therefore, according to (62), PCB for this prosumer equals 207.41%. The results confirm that the prosumers are interested in buying less from the grid, when the electricity prices are relatively high (see `18:00` to `24:00`). The reason for buying power from the grid at `17:00` is the low PV generation, the start time of majority of dishwashers, and the arrival of some of electric vehicles. The VPP at the next hour changes its behavior and compensates the cost imposed at the previous hour. However, since PVs stop generating at `19:00`, prosumers have to buy from the grid since the majority of electric vehicles arrive and intend to charge immediately. From then on, prosumers, to satisfy their load demands, utilize their BESSs while trying to sell the surplus energy to the grid simultaneously.

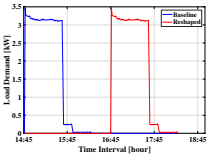
5) Performance Comparison: Fig. 16 shows comparison between recent relevant works and the current work based on four significant evaluation criteria, i.e., PDR, PCB, computation time, and negotiation convergence iteration. This comparison is based on simulation parameters, which are identical in all references. However, some references fail to provide adequate analysis with respect to some of the mentioned evaluation criteria. Interestingly, the proposed framework performs quite better than the recent related works.



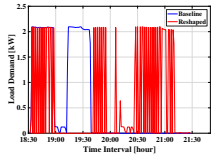
(a) Refrigerator (AOD=0 min. and FUR=N/A).



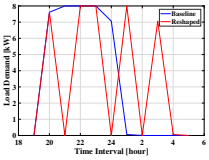
(b) Washing machine (AOD=11 min. and FUR=2.57%).



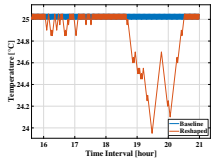
(c) Laundry dryer (AOD=119 min. and FUR=99.16%).



(d) Dishwasher (AOD=92 min. and FUR=33.57%).



(e) Electric vehicle (AOD=294 min. and FUR=59.63%).



(f) Air conditioner (temperature).

Fig. 14. Baseline and reshaped load profiles of appliances of a prosumer.

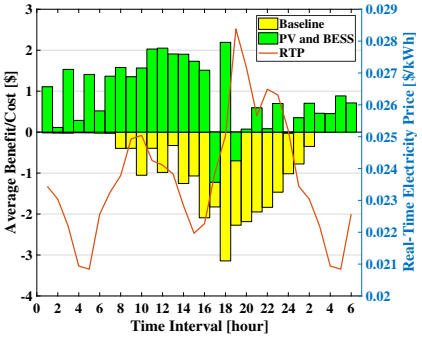
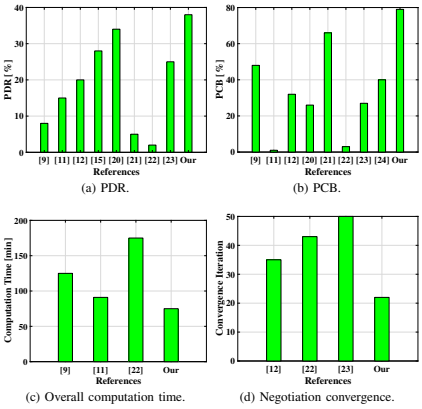


Fig. 15. Hourly benefit/cost of prosumers with and without having PVs and BESSs according to real-time electricity prices.



(a) PDR. (b) PCB.

(c) Overall computation time. (d) Negotiation convergence.

Fig. 16. Comparison between recent relevant works and the current work based on four applicable evaluation criteria.

VI. Conclusions and Future Work

This paper proposes a non-cooperative framework for coordinating a neighborhood of distributed prosumers, which possess smart appliances, photovoltaics, and battery energy storage systems. To relieve the burden of parallel bilateral communications between prosumers and the aggregator, a virtual power plant, on behalf of prosumers, communicates with an agregator to take advantage of their consumption, generation, and storage flexibilities. The framework consists of two multi-objective mixed integer nonlinear programming models for prosumers and the aggregator, by which prosumers are able to schedule their appliances and share surplus power with the grid while the aggregator controls the power matching over time. This paper employs an efficient negotiation approach, in which the virtual power plant and the aggregator negotiate on packaged power and price offers subject to having no knowledge about each other's preferences and utility functions. This approach utilizes an alternating offer package production protocol and a reactive utility value concession strategy, where negotiators have no incentive to deliberately stop conceding while the zone of agreement must remain nonempty.

Four different performance evaluation scenarios, based on real data of load demands and power generation profiles of photovoltaics, as well as real-time hourly electricity prices, are developed to evaluate the effectiveness of the framework according to several economic and environmental assessment metrics, e.g., peak demand reduction, appliance operation delay, prosumer cost-benefit, etc. Simulation results show that, for instance, 500 prosumers, by shifting their load demands and sharing their surplus power, contribute to the peak demand reduction by 32.19% and benefit by 147.97%, both compared to the case, where they do not negotiate to exchange power with the grid. It is also discussed that such assessments are closely related to the weather condition, where fluctuations in photovoltaic generation during a cloudy day decrease the peak demand reduction by 5.77% and prosumer cost-benefit by 27.62% compared to a sunny day. Even though having batteries with higher capacities (e.g., 26.4 kWh compared to 13.2 kWh) shaves the peak demand more by 9.17%, however, it imposes higher purchasing costs to prosumers. Thanks to the bilateral multi-issue negotiation approach integrated with the framework, we show that solutions to the problem with different number of prosumers and ownership levels of shiftable appliances is obtained in a reasonable computation and negotiation convergence times.

Future work will focus on adding a negotiation level between aggregators, integrating industrial and commercial prosumers, where their intermittent load consumptions and power generations can lead to a difficulty in balancing supply and demand, and investigating the network performance and communication delays between different parties.

References

[1] H. Farhangi, "The Path of the Smart Grid," *IEEE Power and Energy Magazine*, vol. 8, no. 1, 2010.

[2] S. M. Nosratabadi, R.-A. Hooshmand, and E. Gholipour, "A Comprehensive Review on Microgrid and Virtual Power Plant Concepts Employed for Distributed Energy Resources Scheduling in Power Systems," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 341–363, 2017.

[3] A. G. Azar and R. H. Jacobsen, "Appliance Scheduling Optimization for Demand Response," *International Journal on Advances in Intelligent Systems*, vol. 9, no. 1&2, pp. 50–64, 2016.

[4] R. Zafar, A. Mahmood, S. Razzaq, W. Ali, U. Naeem, and K. Shehzad, "Prosumer based energy management and sharing in smart grid," *Renewable and Sustainable Energy Reviews (In Press)*, 2017.

[5] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[6] X. Ayón, J. Gruber, B. Hayes, J. Usaola, and M. Prodanović, "An Optimal Day-Ahead Load Scheduling Approach Based on the Flexibility of Aggregate Demands," *Applied Energy*, vol. 198, pp. 1–11, 2017.

[7] R. Zheng, T. Dai, K. Sycara, and N. Chakraborty, "Automated Multilateral Negotiation on Multiple Issues with Private Information," *INFORMS Journal on Computing*, vol. 28, no. 4, pp. 612–628, 2016.

[8] A. G. Azar and R. H. Jacobsen, "Agent-Based Charging Scheduling of Electric Vehicles," in *IEEE Online Conference on Green Communications (OnlineGreenComm)*, 2016, pp. 64–69.

[9] T. AlSkaif, A. C. Luna, M. G. Zapata, J. M. Guerrero, and B. Bellalta, "Reputation-Based Joint Scheduling of Households Appliances and Storage in a Microgrid With a Shared Battery," *Energy and Buildings*, vol. 138, pp. 228–239, 2017.

[10] T. AlSkaif, M. G. Zapata, and B. Bellalta, "A Reputation-Based Centralized Energy Allocation Mechanism for Microgrids," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2015, pp. 416–421.

[11] N. G. Paterakis, O. Erdinç, I. N. Pappi, A. G. Bakirtzis, and J. P. Catalão, "Coordinated Operation of a Neighborhood of Smart Households Comprising Electric Vehicles, Energy Storage and Distributed Generation," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2736–2747, 2016.

[12] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa, "Noncooperative and Cooperative Optimization of Distributed Energy Generation and Storage in the Demand-Side of the Smart Grid," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2454–2472, 2013.

[13] G. d. O. e Silva and P. Hendrick, "Lead–Acid Batteries Coupled With Photovoltaics for Increased Electricity Self-Sufficiency in Households," *Applied Energy*, vol. 178, pp. 856–867, 2016.

[14] E. Nyholm, J. Goop, M. Odenberger, and F. Johnsson, "Solar Photovoltaic-Battery Systems in Swedish Households–Self-Consumption and Self-Sufficiency," *Applied Energy*, vol. 183, pp. 148–159, 2016.

[15] K. Worthmann, C. M. Kellett, P. Braun, L. Grüne, and S. R. Weller, "Distributed and Decentralized Control of Residential Energy Systems Incorporating Battery Storage," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1914–1923, 2015.

[16] Q. D. La, Y. W. E. Chan, and B.-H. Soong, "Power Management of Intelligent Buildings Facilitated by Smart Grid: A Market Approach," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1389–1400, 2016.

[17] T. AlSkaif, M. G. Zapata, B. Bellalta, and A. Nilsson, "A Distributed Power Sharing Framework Among Households in Microgrids: A Repeated Game Approach," *Computing*, vol. 99, no. 1, pp. 23–37, 2017.

[18] T. Taniguchi, K. Kawasaki, Y. Fukui, T. Takata, and S. Yano, "Automated Linear Function Submission-Based Double Auction as Bottom-Up Real-Time Pricing in a Regional Prosumers' Electricity Network," *Energies*, vol. 8, no. 7, pp. 7381–7406, 2015.

[19] K. Rahbar, C. C. Chai, and R. Zhang, "Energy Cooperation Optimization in Microgrids with Renewable Energy Integration," *IEEE Transactions on Smart Grid*, 2016.

[20] B. Gao, X. Liu, W. Zhang, and Y. Tang, "Autonomous Household Energy Management Based on a Double Cooperative Game Approach in the Smart Grid," *Energies*, vol. 8, no. 7, pp. 7326–7343, 2015.

[21] A. Sha and M. Aiello, "A Novel Strategy for Optimising Decentralised Energy Exchange for Prosumers," *Energies*, vol. 9, no. 7, p. 554, 2016.

[22] N. Liu, X. Yu, C. Wang, C. Li, L. Ma, and J. Lei, "Energy Sharing Model With Price-Based Demand Response for Microgrids of Peer-to-Peer Prosumers," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3569–3583, 2017.

[23] Y. Zhou, S. Ci, H. Li, and Y. Yang, "A New Framework for Peer-to-Peer Energy Sharing and Coordination in the Energy Internet," in *IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[24] M. Vinyals, M. Velay, and M. Sisinni, "A Multi-Agent System for Energy Trading Between Prosumers," in *14th International Conference on Distributed Computing and Artificial Intelligence*, vol. 620. Springer, 2018, p. 79.

[25] 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.

[26] S. Bera, S. Misra, and D. Chatterjee, "C2C: Community-Based Cooperative Energy Consumption in Smart Grid," *IEEE Transactions on Smart Grid*, 2017.

[27] H. Nazaripouya, B. Wang, Y. Wang, P. Chu, H. Pota, and R. Gadh, "Univariate Time Series Prediction of Solar Power Using a Hybrid Wavelet-ARMA-NARX Prediction Method," in *IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*, 2016, pp. 1–5.

[28] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa, "Demand-Side Management via Distributed Energy Generation and Storage Optimization," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 866–876, 2013.

[29] K. Deb, "Multi-Objective Optimization," in *Search Methodologies*. Springer, 2014, pp. 403–449.

[30] Z. Xu, W. Su, Z. Hu, Y. Song, and H. Zhang, "A Hierarchical Framework for Coordinated Charging of Plug-in Electric Vehicles in China," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 428–438, 2016.

[31] B. Jie, T. Tsuji, and K. Uchida, "An Analysis of Market Mechanism and Bidding Strategy for Power Balancing Market Mixed by Conventional and Renewable Energy," in *IEEE International Conference on the European Energy Market (EEM)*, 2017, pp. 1–6.

[32] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001, vol. 16.

[33] H. Jain and K. Deb, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach," *IEEE Transactions Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[34] A. J. Hanson, "Hyperquadrics: Smoothly Deformable Shapes With Convex Polyhedral Bounds," *Computer Vision, Graphics, and Image Processing*, vol. 44, no. 2, pp. 191–210, 1988.

[35] A. Rubinstein, "Perfect Equilibrium in a Bargaining Model," *Econometrica: Journal of the Econometric Society*, pp. 97–109, 1982.

[36] G. Lai and K. Sycara, "A Generic Framework for Automated Multi-Attribute Negotiation," *Springer Group Decision and Negotiation*, vol. 18, no. 2, pp. 169–187, 2009.

[37] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

[38] H. A. Simon, "Rational Choice and the Structure of the Environment," *Psychological Review*, vol. 63, no. 2, p. 129, 1956.

[39] J. F. Nash Jr, "The Bargaining Problem," *Econometrica: Journal of the Econometric Society*, pp. 155–162, 1950.

[40] Nord Pool Spot, "https://www.nordpoolspot.com," accessed: May 22, 2017.

[41] M. Pipattanasomporn, M. Kuzlu, S. Rahman, and Y. Teklu, "Load Profiles of Selected Major Household Appliances and Their Demand Response Opportunities," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 742–750, 2014.

[42] K. Qian, C. Zhou, M. Allan, and Y. Yuan, "Load Model for Prediction of Electric Vehicle Charging Demand," in *IEEE International Conference on Power System Technology*, 2010, pp. 1–6.