UNIVERSIDADE DO ALGARVE

GESTÃO OTIMIZADA DE CARGAS EM MICROGRIDS

OPTIMIZED MANAGEMENT OF ELECTRICAL LOADS IN MICROGRIDS

JORGE MIGUEL DA CONCEIÇÃO EDUARDO

Dissertação

Mestrado em Engenharia Elétrica e Eletrónica

Trabalho efetuado sob a orientação de:
Professor Doutor Jânio Monteiro & Professor Doutor Pedro Cardoso

2014

GESTÃO OTIMIZADA DE CARGAS EM MICROGRIDS

OPTIMIZED MANAGEMENT OF ELECTRICAL LOADS IN MICROGRIDS

Declaração de autoria de trabalho

Declaro ser o autor deste trabalho, que é original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

Resumo

A fim de alcançar uma redução de custo da energia elétrica e maximizar o potencial de investimentos feitos em fontes renováveis, um mecanismo de otimização deve ser usado para executar a programação das cargas dos dispositivos elétricos, levando em consideração diversas variáveis, tais como a produção local prevista a partir de fontes renováveis, diferentes taxas tarifárias, restrições de circuitos elétricos, restrições e níveis de conforto dos utilizadores.

Dadas estas considerações, este trabalho define e avalia uma arquitetura e protocolo de gestão de uma *Microgrid* distribuída, que é capaz de otimizar a programação de cargas, tanto para instalações elétricas de pequena como de grande dimensão, considerando todas as restrições e parâmetros mencionados.

A arquitectura proposta foi executada sob um simulador multi-agente e os testes realizados mostram que podem ser obtidas reduções significativas do custo da energia elétrica.

Palavras-chave: *Microgrid*, Algoritmo Genético (AG), Sistemas Multi-agente, Geração Distribuída, Otimização.

Abstract

In order to achieve a reduction in electricity costs and maximize potential investments made in renewable sources, an optimization mechanism should be used to perform load scheduling of the electrical devices, taking into consideration different variables such as the forecasted local production from renewable sources, different tariff rates, electrical circuit constraints, user restrictions and correspondent comfort levels.
Given these considerations, this work defines and evaluates a distributed Microgrid resource management architecture and protocol which is able of optimizing load scheduling, for both small and large electrical installations, considering all the mentioned restrictions and parameters.
The proposed architecture was implemented on a multi-agent simulator and the performed tests show that significant reductions in electricity cost can be achieved.

Keywords: Microgrid, Genetic Algorithm, Multi-Agent Systems, Distributed Generation, Optimization.

Aos meus pais.

Acknowledgements

Durante o último ano que culmina no momento de entrega desta tese de Mestrado, tenho sido constantemente apoiado pelo professor Doutor Pedro Cardoso e pelo professor Doutor Jânio Monteiro. Graças a eles esta tese chegou ao momento de ser submetida. Quero agradecer-lhes de igual forma pelas opiniões esclarecidas que prestaram, pela disponibilidade do tempo pessoal e contribuições em prol desta tese de mestrado e pelos exemplos de atitude pragmática.

Expresso um agradecimento em geral pelo apoio ao Departamento de Engenharia Elétrica e Eletrónica da Universidade do Algarve, ao projecto MTI QREN I&DT, n.º 30260, e aos meus colegas de gabinete de Investigação e do projecto MTI QREN.

Por fim um agradecimento especial aos meus pais Zulmira Eduardo e Jorge Eduardo que têm sido uma fonte inesgotável de suporte, e que merecem destaque nos meus agradecimentos.

Contents

List of Tables

List of Figures

List of Abbreviations

6LowPan	IPv6 over Low power Wireless Personal Area Networks.
ACC	Agent Communication Channel.
ADP	Adaptive Dynamic Programming.
AMS	Agent Management System.
BEM	Building Energy Management System.
CCL	Constraint Choice Language.
COAP	Constrained Application Protocol.
DB	Distribution Board.
DER	Demand Energy Response.
DF	Directory Facilitator.
DR	Demand Response.
DS	Decision Systems.
DSM	Demand Side Management.
DSO	Distribution System Operator.
EA	Evolutionary Algorithm.
EMS	Energy Management System.
EWEA	European Wind Energy Association.
FERC	Federal Energy Regulatory Comission from the United States.
FIPA	Foundation for Intelligent Physical Agents.

FIPA-ACL	FIPA - Agent Communications Language.
FIPA-SL	FIPA - Semantic Language.
GA	Genetic Algorithm.
GPRS	General Packet Radio Service.
HTTP	Hypertext Transfer Protocol.
HV	High Voltage.
IDAPS	Intelligent Distributed Autonomous Power Systems.
IEEE	Institute of Electrical and Electronics Engineers.
IoT	Internet of Things.
IP	Internet Protocol.
IPv4	Internet Protocol version 4.
IPv6	Internet Protocol version 6.
IR	Intermittent Resource.
KIF	Knowledge Interchange Format.
KQML	Knowledge Query and Manipulation Language.
MAS	Multi Agent System.
MCD	Monitoring and Control Device.
MGDB	Main General Distribution Board.
MTP	Message Transport Protocol.
MTS	Message Transport Service.
OASIS	Open Access Same-Time Information System.
OF	Objective Function.
OpenADR	Open Automated Demand Response.
OpenADR 2.0	Open Automated Demand Response - Version 2.0.
QOS	Quality of Service.
RA	Resource Allocation.
RDF	Resource Definition Framework.
RI	Resource Information.

RSVP	Resource Reservation Protocol.
SCADA	Supervisory Control and Data Acquisition.
SDM	Supply and Demand Matching.
SEP	Smart Energy Profile.
SEP 2.0	Smart Energy Profile - Version 2.0.
SPADE	Smart Python Multi-Agent Development Environment platform.
SSL	Secure Socket Layer.
TCP	Transmission Control Protocol.
UDP	User Datagram Protocol.
UGCCNet	Ubiquitous Green Community Control Network Protocol.
US	United States.
XML	eXtensible Markup Language.
XMPP	eXtensible Messaging and Presence Protocol.

List of Symbols

C	Costs vector, price unitary value.
C_0	Cost at initial instant, price unitary value.
$C_0(t)$	Cost at instant t, price unitary value.
$C(t)$	Cost at instant t.
F	Fitness, unitary value.
Hz	Frequency, in Hertz.
k	Constant value for an exponential penalty function.
kV	Voltage, in Killo-Volt.
kVA	Power, in Killo-Volt-Ampere.
kWh	Measure of power to feed a load with one Killo-Watt during one hour.
MV	Voltage, in Mega-Volt.
V	Voltage, in Volt.
$P(t)$	Power at instant t, in Watts.
P_G	Forecasted generation vector of a renewable source, in Watts.
P_{max}	Maximum power of the upward circuit, in Watts.
P_{MCD}	The aggregated power vector, at MCD in Watts.
P_r	Aggregated vector of requested power, from down level aggregator nodes in Watts.
$P_r(t)$	Aggregate vector of requested power in instant t, in Watts.

Q	Quality assessment of the scheduling solution seen from a user perspective.
R	Vector of power available generated by renewable sources.
t	Time instant, in epoch seconds.
T	Tariff associated cost, price unitary value.
ΔT	Difference between two tariffs, price unitary value.

1

Introduction

Most of the worldwide power generation supply to the electrical grids originates from centralized facilities. Those facilities rely mainly on resources such as fossil fuels like coal, petroleum or natural gas, and on energy resources from nuclear, renewable or hydroelectric plants (Indexmundi, 2011). In its early years, between 1940 and 1965, while the demand for electricity was growing, central power plants were assumed as economies of scale. Based on that theory, larger facilities could reduce the cost of production of electricity "per unit" (Phung, 1987). This meant that the production of electricity would become increasingly dictated by the larger companies of the industry and accordingly, by 1970 in the U.S., a large share of the total electricity power was produced by firms which had exhausted economies of scale (Christensen

and Greene, 1976). In the subsequent decades the economy of scale had become less evident.

During the period of 2000 and 2001 the Californian energy crisis took place, marked by a sequence of large-scale blackouts. This crisis was not restrained to California, also the entire Pacific Northwest and the Southwest had been affected by the soaring wholesale prices (Sweeney, 2002). A group of factors had contributed to this dual crisis, first electrical and then financial. The author of "The California electricity crisis" book (Sweeney, 2008), James L. Sweeney, analysed the issue in depth, concluding that one of the fundamental causes of the price volatility in the wholesale electricity markets was the lack of responsiveness of electricity demand to wholesale prices. He suggested a possible solution to the problem: making retail prices more quantitatively responsive and more quickly responsive to wholesale prices, counting on consumers, in their own interest, to reduce electricity purchases when retail electricity prices increase. Thus, as mentioned in (Sweeney, 2008) the price spike and the related economic damages would be reduced if electricity demand were to be more responsive to wholesale price increases. It was also mentioned that this strategy would require electricity regulators to abolish retail fixed-price controls. Typically, traditional electricity regulators decide their investments based on many month of historical records to examine average wholesale prices. Therefore, the response to market prices is often delayed.

After the Californian energy crisis, the Open Automated Demand Response (openADR) standard was created to respond adequately to the detected problems. Since then, OpenADR specifications are being developed to improve the balance between electric supply (independent system operator side) and demand response (customers side) (Piette, 2009).

More recently, the growing penetration of distributed renewable energy sources are increasing the risk of large-scale blackouts, due to their substantial power variations, both in an hourly and daily basis.

In fact, during the period that goes between 2003 and 2012, large-scale blackouts happened in countries such as Germany, France, Belgium, Italy, Austria, Spain, Brazil, Paraguay, Chile and Korea. They have occurred not only in developing countries but also in the developed ones (Nishioka, 2014).

By the end of July 2012, for two consecutive days, India was affected by massive electrical blackouts. The second outage was the largest in history, leaving more than 600 million people, nearly a tenth of the world's population, without electricity (Romero, 2012).

Apart from the real technical fails that led to the afore mentioned problems, these events immediately brought to the light of the discussion how could loads in Microgrids be managed to prevent such problems (Bullis, 2012; Kumar, 2013).

A Microgrid management system could be a solution for these problems, by adding quick responsiveness to the demand side. A Microgrid is a modular small-scale power grid that can operate independent or collaborating with the main electrical grid. An example of an advantage pointed to this system is the possibility to operate detached from the main grid in a so called islanded mode.

1.1 Problem Statement

When comparing current electrical grids with the ones that we had a few years ago, a very different dynamism is verified which results from the increasing introduction of renewable energy sources. Those renewable power sources are sometimes characterized as Intermittent Resources (IRs), as they depend on environmental factors that make them significantly vary over time, being difficult to predict with accuracy. This may in turn cause inefficiencies and mismatches of various kinds in the necessary equilibrium between production and consumption.

In order to reduce these mismatches several solutions can be considered. Some proposals opt for promoting an adjustment in the consumption side (so called Demand

Side Management) using dynamic tariff rates, so that the consumption may adapt to the power being produced. In this field, Distribution System Operators (DSOs) typically buy electricity in markets that already define their prices daily, reflecting the forecasted supply and demand for the following day (as for instance happens in (OMIE, 2014)). These dynamic tariffs are also being applied to DSOs customers in various regions of Europe and United States (UGCCNET-CS, 2011), because constant tariff rates do not correlate with the marginal costs of production (Vasseur and Dunkels, 2010). Based on these tariffs, either automatically or by human intervention, the working periods of equipment can be changed to take advantage of the lowest price and high self production.

In this sense, the goal of creating a system capable of energy management is to implement a set of so-called smart objects (Vasseur and Dunkels, 2010), supported in the concept of the Internet of Things, that by communicating with each other and acting based on an optimized control system, allow a better use of the energy produced by renewable energy sources and the reduction of costs.

In terms of energy control, several protocols like the Smart Energy Profile - Version 2 (SEP 2.0) (ZigBee Alliance, 2013), the IEEE 1888 (Advisory, 2011), and the OpenADR 2.0 (Morgan Hill, 2013) protocol architectures have already been defined. However, while these protocols and architectures can already be applied to Microgrids, a mechanism is necessary to enable the management and control of the distributed resources that are typically available in such grids.

One of such resources is electrical power. In fact, while until now load scheduling has been performed non-automatically, the introduction of automatic management systems in medium to large scale installations can cause demand hikes at low price periods, causing a disruption of supply, due to overloading. Thus, a Microgrid energy management system should take into consideration electrical circuit constraints (Schneider Electric, 2013), while reducing electricity costs and maximizing investments made in renewable sources equipment. That mechanism should implement

load scheduling, resulting from optimization algorithms that reflect user comfort levels and restrictions (Monteiro et al., 2014). It should also consider the forecasted renewable power generation and the different rate tariffs from the DSOs.

1.2 Objectives Definition

The goal of this work is to implement a load control system for Microgrids, taking into account: (1) local production from renewable sources, (2) the tariffs and (3) the structure and constraints imposed by the electrical grid. The system must program the loads entry based on the operation requests, so that they are scheduled to work at times when the tariffs are lower and/or the self production is higher. The system should use optimization algorithms and be scalable to cover electrical installations of large companies, besides the smaller facilities.

1.3 Dissertation Overview

Given the problem statement considerations and goals definition, this work introduces a new Microgrid energy management system which, considering a tree based electrical grid (Schneider Electric, 2013), defines a communication and control structure composed by multiple agents.

To test our proposal, a simulator based on a Multi-Agent System (Shoham and Leyton-Brown, 2008) was implemented. Experimental tests show that electricity cost reductions can be achieved once the management system is used.

The remainder of the document has the following structure. Chapter 2 introduces the use of Multi-agent systems in the context of energy management systems, including architectures, platforms, standards and specifications accepted as the current state of the art. It also presents some power dispatch algorithms considered within energy management systems and finally express market operation features that must

be taken into account in order to adapt the Microgrid system to the market.
Chapter 3 analyses a Microgrid structure and set of protocols developed for communication and control in such electrical grids.
Chapter 4 introduces a Resource Management Protocol for the distributed management of Microgrids. In this chapter are explaining the communication dynamics, metaheuristics and optimization mechanisms settled to correctly implement the proposed protocol. Also in this chapter is described the simulation platform and results obtained using the proposed protocol. Finally, Chapter 5 concludes the thesis and presents some possible future work.

2

An Overview on Microgrid Energy Management Sytems

Above all, electricity is a flexible and adaptable form of energy, which is difficult to store. The consumption from grid customers and correspondent demand is constantly varying, requiring permanent transmission and provision of energy through a distribution system (Legrand, 2009).

The power distribution system (illustrated in Figure 2.1) starts at central power plant and comprises three sections: the national or international power transport network, the distribution network and the local distribution network.

The transport network is characterized by using very high voltages, due to the high

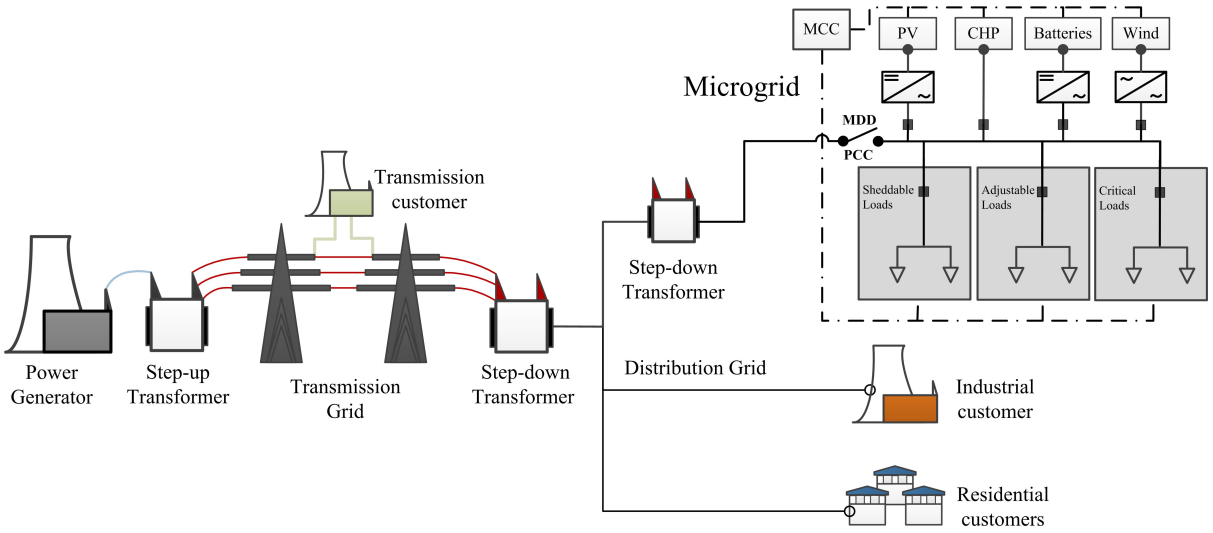


Figure 2.1: Electric power distribution system, taken from (Brucoli and Kevin O'Halloran, 2014).

distances it has to cover. Such voltages can reach $400kV$ as in the case of France, for instance (Legrand, 2009). Using step-down transformer substations (HV/HV), such high voltages are reduced to values that can be supplied to individual users (private houses, shops, tradesmen, small companies, etc).

At the end of the power distribution system, utilities such as private houses, small companies are known as Microgrids. As stated by Zhou et al. (2010), a Microgrid is by definition a power system bundle composed by the low voltage distribution network and power generation assembly with small modules and interconnected loads, which operates as a single controllable unit.

Since a new paradigm was established with the introduction of renewable sources in the electrical grid, a shift in the way the grid is planned is being verified. For example as stated in (Legrand, 2009), according to the European Wind Energy Association (EWEA) the wind-powered production base in Europe could reach an installed power of 180,000 MW in 2020, five times the amount of installed power in 2004. Furthermore, considering that the remainder energy from distributed generation could be provided from customers to the main grid, this theoretically implies power flows in both directions, upwards to supplier and downwards to the customer. In this sce-

nario clients should be considered as possible energy providers to the grid, thus being capable of selling energy to the DSO.
However electricity is a shared commodity that must be protected from disturbance. Deregulated intervention can disturb the quality of energy, bringing harmful effects, such as the deterioration of the power factor, harmonics or transients (Legrand, 2009). For that reason the quality of energy is controlled by strict standards.
Automated systems could be used to disconnect producers that do not comply with standards. However random events are not controllable, and thus may cause damages. The power system is regulated for a current frequency of 50 Hz (or in some countries 60 Hz) that need to be respected for safety (Legrand, 2009). Tension values are also required to be maintained within a range of 10 percent around the nominal value. Injecting power into the grid tends to increase the tension and thus inverters are required to monitor it and disconnect or reduce their current if the maximum tension is achieved. This prevents them from delivering all the power that they could. In this scenario, demand side management is required not only to assure the equilibrium between production and consumption, but also to deliver the highest possible percentages of production to loads.

2.1 Microgrids Market Operation

In order to support distributed generation, both legislation and technical standardization are required. This process has started many years ago when the monopolistic market has evolved to a more competitive one, enabled by a trade of electricity in a fair open access process, open to all electrical power suppliers (Shahidehpour and Alomoush, 2002). This transition was not a simple process, as the Californian energy crisis have shown, partially caused by a market deregulation.
Resulting from the Californian energy crisis, the Federal Energy Regulatory Commission (FERC) from the United states has issued two normatives, known as Orders 888

and 889. The Order 888 (from 1996) enforces open access non-discriminatory tariffs, facilitating unbundling of wholesale generation and transmission services (Chowdhury and Crossley, 2009). The Order 889 for the development of the electronic communication systems designated as Open Access Same-Time Information System (OASIS) facilitated the development of three main restructuring models namely PoolCo Model, the Bilateral Contracts Model and the Hybrid Model (Chowdhury and Crossley, 2009). OASIS is a reference for the markets in worldwide scale, because it addresses a common concern about the transparency of market based operations, reporting procedures of conformity.

More recently distributed generation was allowed to be integrated in the distribution grid, encouraged by feed-in tariffs designed to accelerate the investment in renewable energy technologies. In these solutions however DSOs were required to pay more for the electricity they buy than for the one they sell. This could not constitute a feasible long-term solution.

The integration of renewable sources into Microgrids for self-consumption, were only allowed if they didn't assured power injection into the DSO grid. Currently, this is being modified by legislation that enables self-consumption, while enabling the payment for the exceeding power at a price close to the one the DSOs pay.

In either case, the flexibility introduced by these new solutions require a communication platform to control electrical equipment. This is the aim of next section.

2.2 IoT Protocols Applied to Microgrids

The user's ability to manage their energy consumption according to the production is a critical feature of smart grids, and a base for innovation, new products and services. The home grid will be much like a small Internet, where every equipment will be interconnected and interacting with each other. This concept called Internet of Things (IoT), derives from the fact that an increasing number of devices are now able

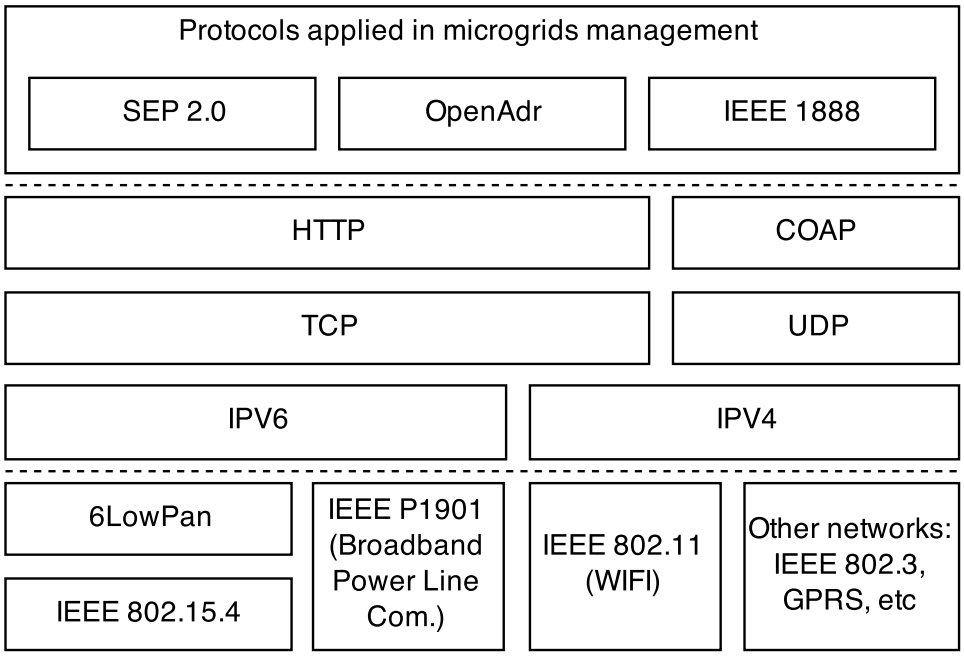


Figure 2.2: IoT protocols architecture.

of using IP networks, and thus seamlessly being able of reaching other devices that could be in the same residence, or in the other side of the world. These devices could include a freezer, a washing machine or a clock. According to van der Meulen (2013), in 2020 the number of IoT devices will be 30-fold compared to 2009, growing to 26 billion units installed, not taking into account PCs, tablets and smartphones.

Due to the limited address space of IPv4, which allows for 4.3 billion unique addresses, IoT devices will have to use IPv6 addresses to accommodate the required address space. Based on the idea that small low powered devices with limited processability could be provided with Internet Protocol then it was created the 6LoWPAN protocol, which means IPv6 over Low power Wireless Personal Area Networks. This protocol is aimed to facilitate the integration of new devices in the IoT concept.

In terms of energy management, IoT devices constitute a solution to the required balance between production and consumption. In order to support such interconnectivity, the communication between different devices such as meters, appliances, electric vehicles, energy management systems and distributed energy resources (including renewable energy and storage) must occur using secure, standard and open proce-

dures. In this context, several protocols have been defined.

Figure 2.2 shows a stack of the several protocols that were defined to support the IoT and particularly the energy management systems. Among others, the refered stack is composed by:

- HTTP - designed for human-computer interaction, it is currently being used to support the interoperability of applications and devices, using a REST architecture based on GET, HEAD, PUT, POST and DELETE methods.
- Smart Energy Profile - version 2 (SEP 2.0) (Morgan Hill, 2013) - this protocol results from the collaboration between the low-power ZigBee, Wi-Fi and HomePlug power-line technologies, building a power management architecture for Microgrids, supported on IP networks.
- IEEE 1888.x - In March 2011, the Institute of Electrical and Electronics Engineers (IEEE) announced the approval and publication of the Standard IEEE 1888 TM (Advisory, 2011) within the Ubiquitous Green Community Control Network Protocol (UGCCNet). Originating in China, the IEEE 1888 standard defines itself as a global standard within the IoT, which aims at energy efficiency through the management of renewable energy, through communication using Internet protocols and information and communication technologies.
- OpenADR - The OpenADR (Morgan Hill, 2013) is an evolution and extension of the first version, developed by the Demand Response Research Center at Lawrence Berkeley National Laboratory. It is supported by the OpenADR industrial alliance, having been developed as part of the standard OASIS Energy Interoperation 1.0 (OASIS Energy Interoperation Technical Committee OASIS, 2012).

If on one hand the protocols that allow communication between different devices of a Microgrid are being developed, a control procedure is still needed to support an opti-

mized load scheduling when managing distributed resources. This is the purpose of the forthcoming sections.

2.3 Microgrid Energy Management using Multi-Agent Systems

Supervisory Control and Data Acquisition (SCADA) systems emerged in the 1960s (Laboratories, 2012), as one of the first mechanisms to automate power systems control. Since then the SCADA systems have a clear relevance in large scale critical infrastructures such as power generation and power transmission (Calderwood et al., 2013). The SCADA system consists of a centralized software control system that collects and monitors information remotely, received from sensors, to support the control of equipment, devices, and automated functions.

The energy management products available that provide SCADA functionality for Microgrids are installed with vendor specific control software protocols which limits communication with different Demand Energy Response (DER) devices (Feroze, 2009). The advantage of using a multi-agent approach resides in the fact that it might be independent from vendor restriction and provides a common communication interface for all the elements in a distributed manner (Logenthiran et al., 2010b), dealing better with uncertain scenarios. Also SCADA operation is typically causal, which mean that it only responds to the present and past events, and do not consider future events, obtained for instance using forecasting methods.

As a promising management system, the Multi Agent Systems (MAS) could be applied in a wide variety of problems in power engineering. They have already been used for example in diagnostics (Davidson et al., 2006), power system restoration (Nagata and Sasaki, 2002), market simulation (Praça et al., 2003), handling system vulnerabilities (Liu et al., 2000) and Microgrid control (Dimeas and Hatziargyriou, 2004).

Basically, as stated in (Roche et al., 2010), a MAS can be viewed as a collection of autonomous and intelligent entities called agents, evolving in an environment where they can perceive and act on. This environment can be considered as everything but the agent itself. Many researchers believe that agents represent the most important new paradigm for software development since object oriented concept (Aranda et al., 2006; Luck et al., 2005). The concept of agent is characterized by a high level of abstraction, which enables the development of complex computational systems (Aranda et al., 2006).
The most important features of MAS include its flexibility, autonomy, reactivity, proactiveness, social ability, the distributable nature and the fault tolerance of agent systems (McArthur et al., 2007b).
In the next section we will focus on the MAS architectures when it comes to Microgrids.

2.3.1 MAS architecture

The majority of the works that have already addressed Microgrid architectures with MAS, use a basic structure, based on two or three hierarchical layers (Jimeno et al., 2011; Jian et al., 2009; Logenthiran et al., 2010a; Sujil et al., 2014; Xiao et al., 2010; Zhou et al., 2010).
For example, Roche et al. (2010) describes a general system with three layers:

- A bottom layer constituted by a set of agents that control generators, loads or storage. These agents have various degrees of intelligence giving them the ability to react in real time.
- An intermediate layer with area or zonal operators, in charge of coordinating the components of the bottom layer (generators and loads).
- A top layer corresponding to a distribution network or Microgrid operator, that coordinates several areas by optimizing global operations.

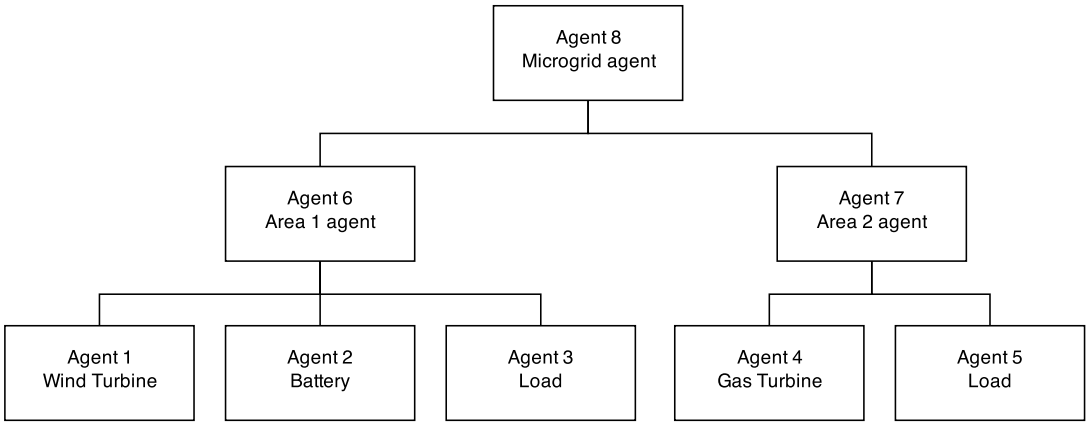


Figure 2.3: Multi-agent common architecture, adapted from (Roche et al., 2010).

Figure 2.3 shows a generic scheme of this three-level layered tree.

This structure is suitable to be associated with the real electric circuit installation. In such scenarios a distinct agent could monitor/control each distribution boards, specially in structures bigger than home grids (e.g. hotels, factories or buildings). The maximum depth of these circuits extend to a maximum of two switchboard levels after the main general distribution board. For larger installations, the number of switchboards increase horizontally (Schneider Electric, 2013). In either case, agents are not required to be logically associated with every single electric switchboard.

Another architecture that is based in three layered tree architecture is the agent-based Building Energy Management (BEM) system, inspired on the multi-zone nature of the buildings (Guo et al., 2013). The interest in this architecture results from the use of a functional approach. The design of agents is thus planned according with the functions of the building and their interactions.

Another interesting architecture called PowerMatcher was conceived to simulate the electronic market implemented in a distributed manner via a tree-structure of so-called SD-Matchers (Kok et al., 2005). The PowerMatcher is a market-based control concept for supply and demand matching (SDM) in electricity networks with a high share of distributed generation. In the PowerMatcher method each device is represented by a control agent, which tries to operate the process associated with the de-

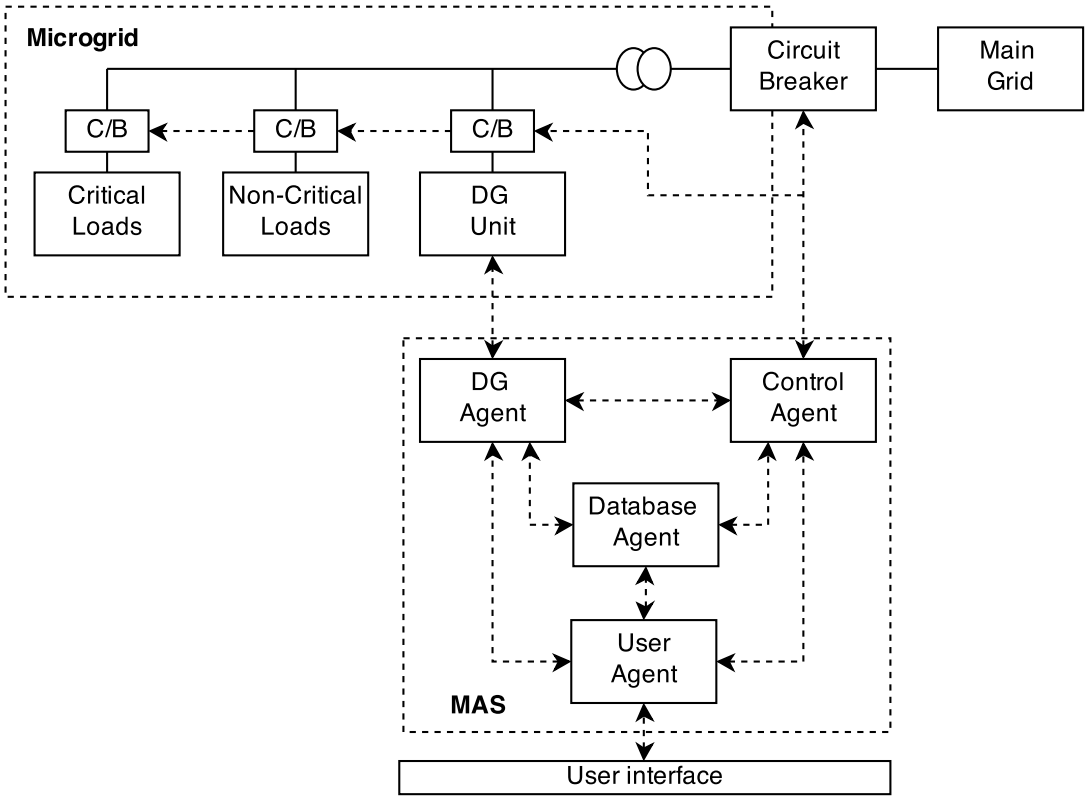


Figure 2.4: A conceptual model of a MAS architecture, adapted from (Kulasekera et al., 2011).

vice in an economical optimal way, balancing biddings.

In (Kulasekera et al., 2011; Pipattanasomporn et al., 2009) a conceptual model of MAS Microgrid architecture is presented. The architecture is designed to control a general Microgrid setup, usually referenced with Intelligent Distributed Autonomous Power Systems (IDAPS) architectures. The model is based on a single-layered distributed control architecture which comprises three major agents, charged of sensing and controlling the components of the Microgrid.

Figure 2.4 illustrates the MAS architecture conceptual model for a single layered architecture.

However such single layered architectures lack capacity to develop a distributed control model. Thus, as stated in (Kulasekera et al., 2011), the proper delegation of multiple objectives to a secondary layer of agents is required to improve the robustness.

In terms of security, a Microgrid Cyber Security Reference Architecture was defined

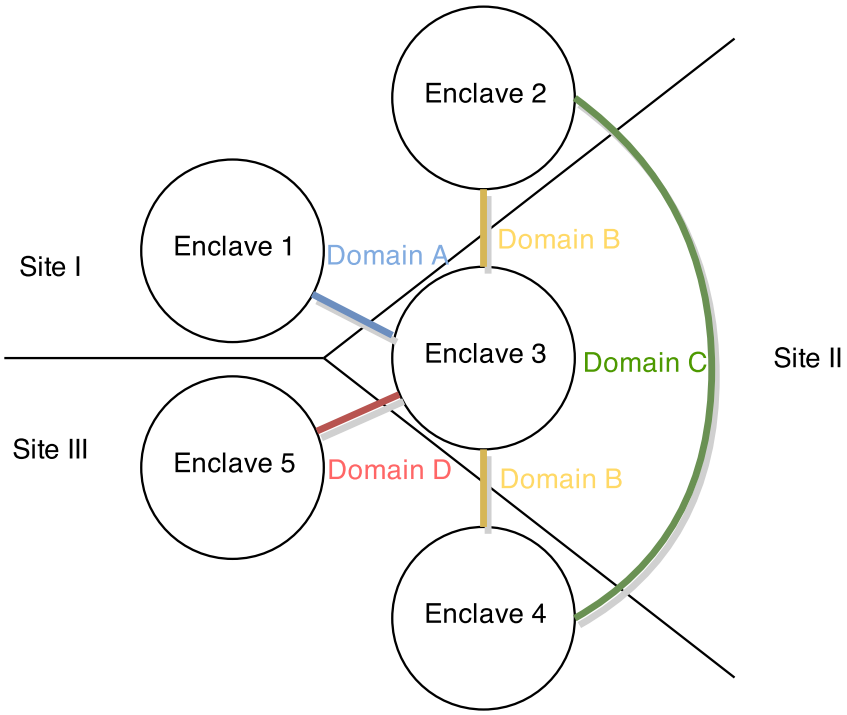


Figure 2.5: Example of the segmentation of a network into enclaves by functional domains, adapted from (Veitch et al., 2013).

by the Sandia National Laboratories, in a technical report created for the United States Department of Energy (Veitch et al., 2013). Their design approach addresses the security concerns, presenting the segmentation of the Microgrid control system network into enclaves, grouping enclaves into functional domains, and describing actor communication using data exchange attributes.

Figure 2.5 sketches an example of the segmentation of a network into enclaves and functional domains.

2.3.2 MAS Specifications, Standards and Interoperability

In the field communication standards and interoperability, the Foundation for Intelligent Physical Agents (FIPA) is a non-profit organization aimed at producing standards for the interoperation of heterogeneous software agents (Luck et al., 2003). Since 2005, FIPA is formally accepted as a standards committee of the IEEE Computer Society for MAS (McArthur et al., 2007a).

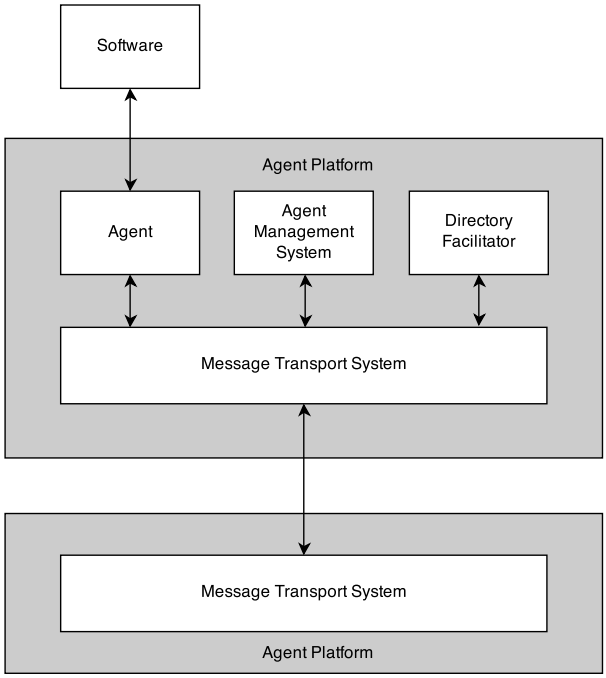


Figure 2.6: FIPA Agent Management Reference Model, adapted from (FIPA00002, Supersedes, 2000).

FIPA aims to define specifications and standards that can be used to support interoperability between agent-based systems developed by the different companies and organizations (McArthur et al., 2007a; FIPA00002, Supersedes, 2000). The usage of standards facilitates the integration between previously separate systems.
The main contributions from FIPA include the definition of the Agent Management Reference Model, the Agent Management Services (namely the Directory Facilitator (DF) and the Agent Management System (AMS) services), and the Message Transport Service (MTS). There is also a non standard agent communication ontologie proposal.

Agent Reference Model

In order to customize the agent model, FIPA has created a standard Agent Management Reference Model, which is illustrated in Figure 2.6.
Every Agent Platform comprises two utility agents, the AMS, which is mandatory, and the DF agent, which is optional. For more details see (FIPA00002, Supersedes,

Message field	Description
performative	Type of communicative act
sender	Participant in communication
receiver	Participant in communication
reply-to	Participant in communication
content	Content of message
language	Content of language
encoding	Encoding of content
ontology	Ontology used
protocol	Protocol for conversation
conversation-id	ID for conversation control
reply-with	Conversation control parameter
in-reply-to	Conversation control parameter
reply-by	Conversation control parameter

Table 2.1: FIPA-ACL message fields, adapted from (McArthur et al., 2007b)

2000).

Agent Communication Languages (ACL)

Agent communications represent a challenging concern for standardization. The reason for such a statement is justified by a variety of possible terminologies that can be proposed according to different conceivable scenarios. A FIPA Agent Communication Language (FIPA-ACL) message contains 13 fields, displayed in Table 2.1.

The only compulsory field in the message is the performative field that defines the type of communicative act. With the use of performative, FIPA-ACL ensures that the recipients of the message will understand the meaning of a message in the same way as the sender does, removing ambiguity about the content (McArthur et al., 2007a). FIPA also specifies 22 performative acts that define the type of message content and the flow of messages expected by each the agent during the performative communication act.

Content Languages and Ontologies

FIPA has proposed four different content languages for standards: (1) FIPA-Semantic Language (FIPA-SL); (2) Knowledge Interchange Format (KIF); (3) Resource Definition Framework (RDF); and (4) Constraint Choice Language (CCL) (McArthur et al., 2007a).

The content of a message comprises two parts: language and ontology. The ontologies are a supplementary part of the language that represent the common vocabulary. While the content language signifies the grammar of the content. Ontology relates to the semantics. Unless agents employ a common ontology they will not be able to understand the received messages.

Thus, FIPA has created the FIPA-ACL with its roots in speech act theory and incorporating many aspects of Knowledge Query and Manipulation Language (KQML), with the aim to provide a reference for communications standardization.

2.3.3 MAS Platforms

The existent open-source platforms for the design and implementation of MAS - like JANUS, JADE, ZEUS or MadKit Roche et al. (2010) - were not taken into consideration in this work, because those frameworks where not developed in Python. The elected programming language for this work. Instead of developing a solution from scratch, it was decided to search different frameworks, that could support demanding applications. We have opted to use SPADE (Smart Python Multi-Agent Development Environment), a platform among others that was developed in Python that in addition follows the FIPA standard proposal for MAS (Aranda et al., 2006). SPADE communications are built around the XMPP/JABBER, a protocol commonly used in chat messaging. Other protocols capable of supporting message transport, such as the HTTP protocol, are also supported. Figure 2.7 presents the general representation of the SPADE framework.

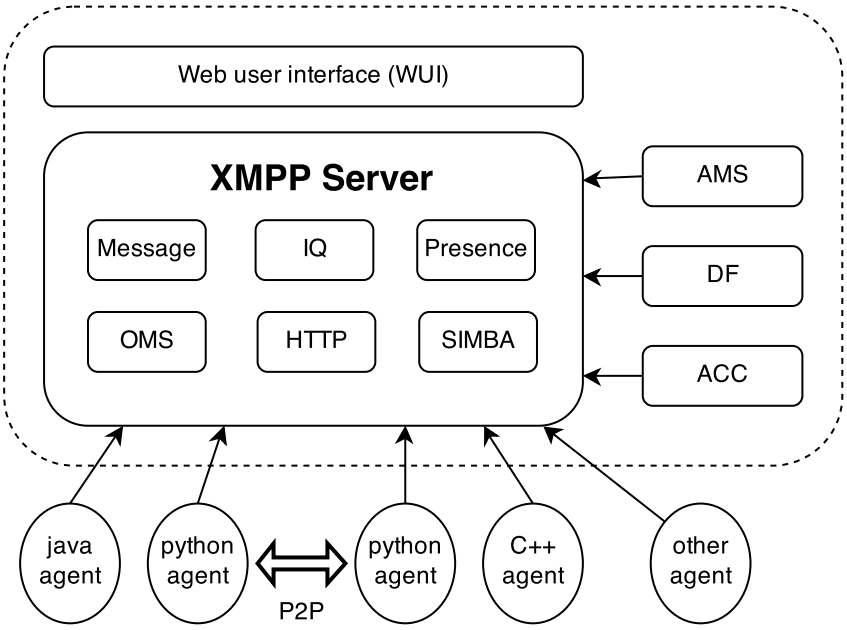


Figure 2.7: SPADE platform model, adapted from (Palanca, 2014).

Among the elements of the SPADE platform which are important to emphasize, we find the XML router/XMPP server, the Directory Facilitator (DF), the Agent Management System (AMS) and the the Agent Communication Channel (ACC).
In the following, the formal definitions of each element is presented (Gregori et al., 2006):

- XML router (XMPP server) - it is a standard XMPP server that routes all the messages from its sender to the specified receiver with no user intervention. This XML Router acts as the Message Transport System (MTS).

- Directory Facilitator (DF) - it is a component that provides a service directory to register and query services offered by the agents that are registered at the platform.

- Agent Management System (AMS) - is the component that implements the basic management services for the agents.

- Agent Communication Channel (ACC) - manages all the communication within the platform. It receives the FIPA-ACL messages that arrive to the platform and

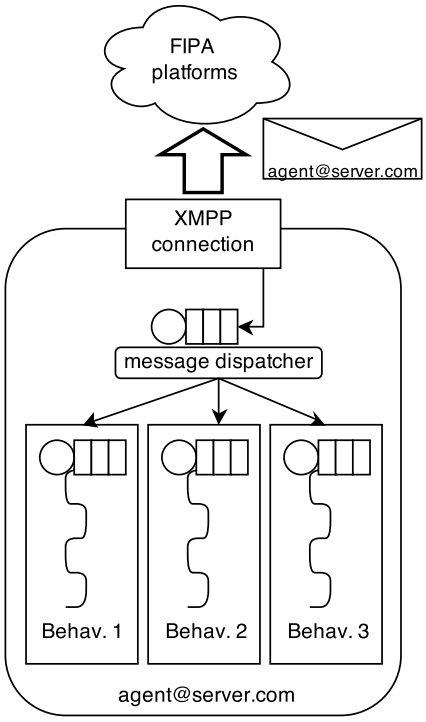


Figure 2.8: SPADE agent model, adapted from (Aranda et al., 2006).

it redirects them to the correct destination element. This destination can be either an agent or another component

Besides the platform model, SPADE also has its own Agent Model, illustrated in Figure 2.8.

The Agent Model is composed by a connection mechanism to the platform, a message dispatcher, and a set of different behaviours that the dispatcher attributes to the messages.

Elements staked out from the agent model are the identifier called Jabber ID (JID) (which is essential) and a password validation to establish the connection with the platform that is optional and recommended.

An agent can run several behaviours simultaneously. A behaviour is a process that an agent can execute using repeating patterns. SPADE provides some predefined behaviour types: Cyclic, One-Shot, Periodic, Time-Out, and Finite State Machine behaviours. Those behaviour types help to implement different behaviours that an agent can perform. Every agent can have as many behaviours as desired. When a message

arrives to the agent, the message dispatcher redirects it to the correct behaviour queue (Gregori et al., 2006).

A great advantage of this framework resides in the interoperability. The Message Transport System (MTP) is able to communicate with any kind of FIPA compliant platform that supports the XMPP Message Transport Protocol. A JADE plug-in that provides a MTP, implementing the XMPP protocol was also developed (Gregori et al., 2006).

Another advantage is the ability to run SPADE across multiple platforms. Several architectures and operating systems are possible such as Windows, Linux, MacOS, Windows Mobile, PalmOS, SymbianOS for mobile phones, etc.

The SPADE framework provides some built-in security mechanisms that help to maintain the system's integrity. For instance it supports a security password to log in to the XML Router and the connection to the XML Router which can be encoded with a symmetric cryptographic algorithm using SSL (Gregori et al., 2006).

2.4 Power Dispatch Algorithms

In this scenario, the main objective of economic dispatch of electric power is to schedule loads in order to use the maximum of the power being produced, with the minimum operational cost, while satisfying system constraints, and having into consideration tariffs and distributed generation. To perform this task, meta-heuristic algorithms are the most popular solutions (Zambonelli et al., 2003). However none of these solutions was until now capable of differentiating itself as an ideal method. In this section we further explain the current context of Microgrid energy management. This includes the analysis of architectures, platforms, standards and specifications accepted as the current state-of-the-art.

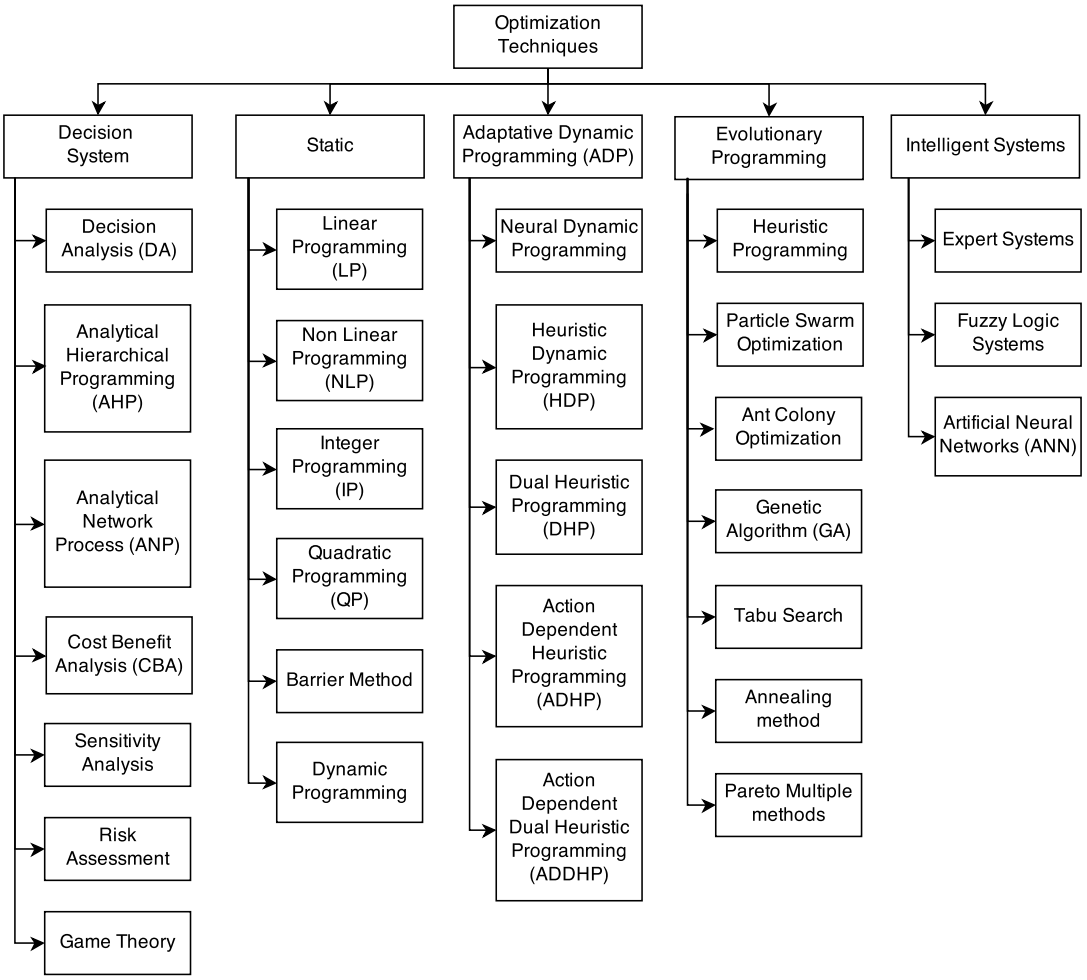


Figure 2.9: Categorization of the optimization techniques, adapted from (Momoh, 2012).

2.4.1 Available Power Dispatch Algorithm

A wide variety of optimization techniques were already applied to solve different challenges placed by smart grids.

Figure 2.9 presents a classification of the optimization solutions, in five groups, namely: Decision Systems, Static Methods, Adaptive Dynamic Programming (ADP), Evolutionary Programming and Intelligent Systems (Momoh, 2012). In some cases, these techniques could also be combined to form hybrid solutions.

Among these solutions, the Evolutionary Programming family of methods is currently widely used to solve a large range of problems. In particular, the Genetic Algorithms (GA) are highly relevant for its industrial applications, because they are ca-

pable of handling problems with non-linear constraints, multiple objectives, and dynamic components properties that frequently appear in real-world problems (Roeva et al., 2013). In the Microgrid's scope they have been applied on profit-based optimal generation scheduling (Razali and Hashim, 2010), on operation planning of Multi Smart Microgrids (MSMG) (Nagasaka et al., 2012) or on Multiobjective Environmental/Economic Power Dispatch (Mohamed and Koivo, 2008). Their applicability may extend beyond these three examples due to the number of challenges that may arise with the integration of plug-in electric vehicles or aid operations of electricity markets, among others. In many of these cases, GAs are implemented with a multi-objective function.

Given its importance, in the next section we will make an overview of the aforementioned GA algorithm.

2.4.2 Genetic Algorithm

Genetic algorithms are a stochastic, parallel algorithm inspired by genetics, evolution theories of natural selection and survival of the fittest, being used for general purposes. It is an iterative procedure acting on a population of chromosomes. Each chromosome encodes a candidate solution to the problem (Logenthiran and Srinivasan, 2009).

The fitness is a measure of accountability associated with each chromosome, important on the performance of the algorithm. In general, among the population of chromosomes, or individuals, the chromosome with the lowest fitness values represent the best solution for a minimization problem. Otherwise in a maximization problem the best solution would be associated with a bigger fitness value.

The fitness function can use a penalty to penalize potential solutions which violate constraints. The objective function returns a fitness value determining the ability of the solution to survive and to produce offspring.

New generations of solutions are obtained through processes of selection, crossover

and mutation. During the evolutionary process, new generations should result in increasingly better solutions and evolve toward an optimal solution. Using elitism replacement, it is possible to preserve a determined number of best individuals from a previous algorithm population.

The process is described in the flow chart presented in Figure 2.10 and a more detailed explanation, applied to the work presented in this thesis, will be presented in Section 3.3.2.

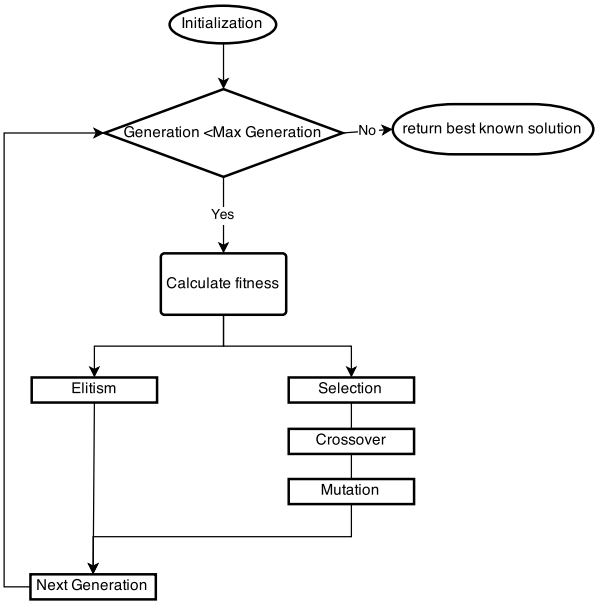


Figure 2.10: Genetic algorithm flow chart, adapted from (Logenthiran and Srinivasan, 2009).

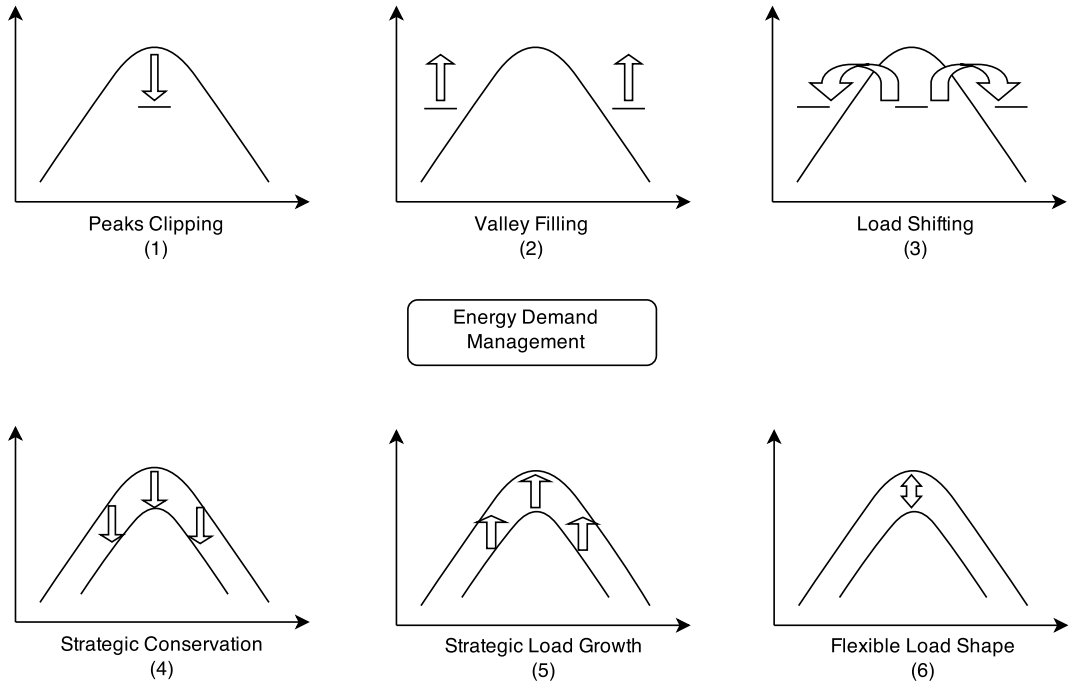


Figure 2.11: Energy demand management techniques, adapted from (Gellings and Smith, 1989).

2.5 DSM Techniques

The challenge that our system aims to manage is caracterized by typical Demand Side Management (DSM) techniques. DSM consists in the supervision, control and change of the energy demand on the consumers side. This management involves the identification and implementation of a series of initiatives designed to encourage a more efficient use of energy through financial bonuses or penalties (different electricity tariff prices depending on the schedule), management of load profiles or energy conservation, among others (Gellings and Smith, 1989).

The most relevant DSM techniques aim to reshape the aggregated load curves by changing consumption profiles. As shown in Figure 2.11, there are six common DSM techniques: Peaks Clipping, Valley Filling, Load Shifting, Strategic Conservation, Strategic Load Growth and Flexible Load Shape.

Although they promote energy efficiency, not all of these measures necessarily imply a reduction in the energy consumption on the consumer side, but they encourage a

better use of energy by avoiding peak loads during the hours of highest consumption and by promoting the use of that power in off-peak hours, leading to lower load variations. This way, they improve system stability and also avoid "blackouts" and unnecessary investments in increasing production and network capacity.
For example, load shifting techniques such as Peak Clipping and Valleys filling are implied in our management system, as a consequence of a system guided by different tariff prices according to different periods of time, as we will see later.
Globally flexible Load Shape is the set of all measures represented in Figure 2.11(1)-(5). This technique requires a total control of facility equipment, according to the parameters that are considered, such as tariffs, energy production, energy storage, network stability, desired load diagram, user comfort, energy saving and energy efficiency. Adjustable Load Curve measure is actually the more approximate measure that represents what our system does.
This will entail an automatic management by the system on all devices on the network, depending on the instructions given by the user or the supplier to promote an effective management and sustainable use of energy. The load diagram will vary over time according to the strategies used in order to try to meet the objectives that were imposed on the system, approaching the actual consumption to the desired consumption.
In a smart grid utility, DSM strategies need to manage a large number of controllable loads of several types. Moreover, loads can have profiles which spread over a few hours. Thus, the strategies should be able to deal with all possible control durations of a variety of controllable loads (Logenthiran et al., 2012). Although non-controllable loads are in acquaintance, the system as to deal with them as an adverse effect.

3

A Resource Management Protocol for the Management of a Microgrid

In this chapter we will present a new resource management protocol for the management of Microgrids. We will start by presenting the considered Microgrid architecture, followed by the description of mechanisms designed to control the Microgrid network communications.

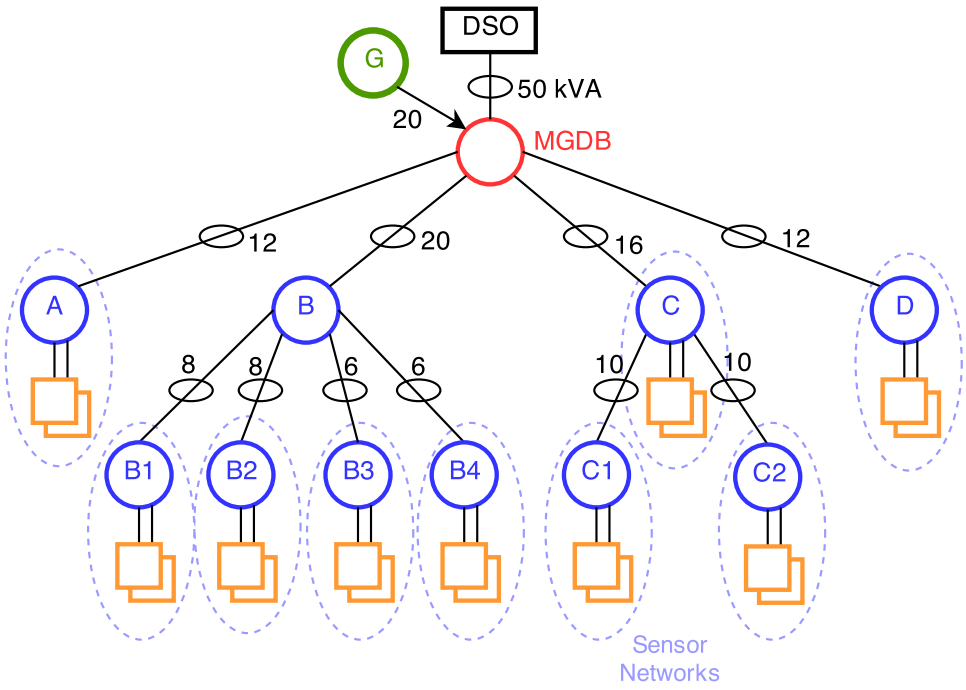


Figure 3.1: Example of a Microgrid architecture with a tree structure comprising several distribution boards and having a renewable generator, in node G.

3.1 Microgrid Topology Architecture

Figure 3.1 presents a typical structure of a low voltage Microgrid (Schneider Electric, 2013), common among industrial and business facilities. These structures are composed by a hierarchy of Distribution Boards (DB), where the Main General Distribution Board (MGDB) interconnects the external DSO circuits to several internal workshop circuits represented as A, B/Bx, C/Cx and D in Figure 3.1. Workshop DBs can be divided into intermediate DBs if they obligatorily feed other lower level DBs, and possibly, electrical loads (e.g., A, B, C and D in Figure 3.1) or leaf DB if they only feed loads (e.g., B1, B2, B3, B4, C1 and C2 in Figure 3.1).

At the lower levels we find electrical loads, represented in Figure 3.1 by the orange boxes. They can be controlled in terms of one or more of the following parameters: from when to start, up to when they are allowed to finish, or the maximum power to be drawn from the electrical grid. Some of them can also be controlled through indirect parameters, as for an example HVAC (heating, ventilation, and air conditioning)

set point temperatures. Finally, some loads are not controllable and/or individually monitored.

Each intermediate and leaf DB can connect dozens of circuits, aggregating hundreds of loads. Furthermore, simultaneity (or diversity) factors, k_s, are applied at each DB level, considering that not all equipments run at the same time. Usually, the simultaneity factor values range from 0.1 to 1.0, depending on the type of loads that are connected to a certain circuit. They enable the computation of the expected resulting aggregated load, which is drawn from higher levels boards. This procedure is repeated in higher DBs, leading to an expected total demand for the full installation (exemplified as 50 kVA in Figure 3.1). The aggregated power of these installations can easily reach hundreds of kVA in industrial installations, distributed over dozens of DBs.

In general, simultaneity factors result from practice and consider that the working periods of the equipment are typically spread over time. However, they are not computed considering that many devices could work at the same time, as it may happen if a period of lower tariffs is combined with a greedy automatic load shifting. Thus if scheduling is applied to loads, some measures as the one proposed in the next sections should be taken to avoid overloads.

3.2 Microgrid Communications Architecture

Given an architecture as the one presented in Figure 3.1, the communication structure that controls and monitors electric devices should derive from the electrical structure. Thus, in such control system, we consider that a Monitoring and Control Device (MCD) should be placed at each DB. The set of MCDs will form a distributed Energy Management System (EMS) of the whole installation.

At each distribution board, MCDs measure the current, voltage, active and reactive power consumed from the upward circuit, while communicating through wireless

and/or wired sensor networks with electrical equipment. Sensors devices are also used to measure environmental data (e.g., temperature, movement, and light intensity). MCD devices are thus in charge of Machine-to-Machine communication while reflecting Human-to-Machine interactions. Based on these inputs they define when terminal devices should work. These load scheduling decisions should result from optimization algorithms that take into consideration:

1. The forecasted power curves of installed renewable sources in the yet-to-come minutes/hours;
2. The power consumption curve of each equipment/load;
3. The future minute/hourly based tariffs charged by the DSO;
4. The local and global power constraints imposed by the electrical installation; and
5. The human requirements and comfort levels.

Given the computation capabilities available in many electronic devices, MCDs are currently capable of running optimization algorithms and communicating with each other for the management of distributed resources, which are shared by the whole Microgrid. While this distributed architecture is capable of parallel computing, it also places several challenges in terms of coordination between control devices and scalability.

In order to address these issues, in the following section we consider that optimization algorithms for load scheduling run in a distributed fashion at MCDs, making local decisions that reflect a global equilibrium of the system. Given these considerations, we will define and evaluate a communication mechanism that can be used to manage these electrical devices.

3.2.1 Introducing the Distributed Resource Reservation Protocol

The problem of distributed resource reservation has been addressed previously in computer networks. The Resource Reservation Protocol (RSVP) in particular, specified in IETF RFC2205 (Zhang et al., 1997) and updated since then with several features, was used to support a distributed Quality of Service (QoS) resource reservation procedure among several integrated services routers (Wroclawski, 1997).

RSVP considers two fundamental message types: PATH and RESV. In IP Multicast trees, the PATH message travels downstream along the multicast routes with information about the traffic that the sender application expects to generate and storing the path state with the QoS control capabilities of routers along the path. RESV messages are originated in leaf nodes and travel upstream, being used to request an appropriate resource reservation from the desired QoS. As RESV messages move from receivers to senders, reservation parameters are merged at intermediate nodes.

While the RSVP protocol can't be applied directly to the resource reservation problem, a similar concept may be used to implement a distributed mechanism for load management as we will propose in the next section.

3.2.2 Microgrid Resource Management Protocol

Distinctly from the RSVP protocol, that only reserves flows for a subsequent time period, in the optimization mechanism that we are considering such requests should also address in the "far" future time intervals. This means that resource request messages must carry a structure with vector of n power requests, where each index refers to a time interval (e.g., for the 5 minutes interval between 10:15 and 10:20).

The proposed Microgrid Resource Management Protocol considers two communication phases (as shown in Figure 3.2), which are similar with the ones that were defined for RSVP. For each of these phases one message type is used: a Resource Information (RI) message, or a Resource Allocation (RA) message.

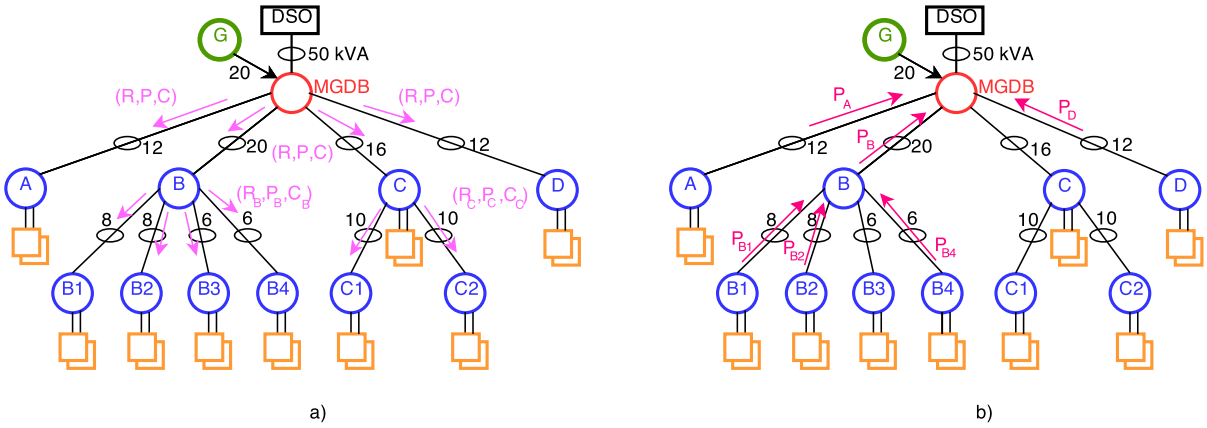


Figure 3.2: Two stages of the Microgrid Resource Management Protocol: a) Resource Information (RI) messages travel down the tree carrying information about the resources that are commonly distributed, b) Resource Allocation (RA) messages inform upper nodes about the forecasted power consumption of each aggregator node.

In the first stage, the MCD, at the top of the tree, multicasts RI messages. Each of these messages contains three vectors, represented by (R, P, C), where:

- R is a vector which informs lower MCDs about the forecasted power that is expected to be generated by renewable sources;
- P is a vector which translates the ratio of maximum upward power that lower MCDs can allocate;
- C contains the energy cost associated with each time interval.

Each of the time intervals of the C vector starts by reflecting the tariff of the DSO. However, as explained later, the associated values will be adjusted to avoid cyclic overloading in adjacent time periods, penalizing the intervals where overloads occur. As these RI messages traverse down the tree (i.e., from the top to leaf MCDs), P and C vectors may be changed by intermediate MCDs, in order to reflect their own capabilities and states. Thus, when these RI messages reach a MCD leaf, the (R, P, C) vectors reflect the capability of the whole grid, being used as input in the optimization algorithm to decide: when loads should start working, when they should finish and/or what is the power level they are allowed to request (Eduardo et al., 2013).

Leaf MCDs, after running the optimization algorithm, generate an aggregate load vector, which is sent upstream using a Resource Allocation (RA) message. Intermediate MCDs, after receiving RA messages from lower MCDs, behave like leaf MCD, i.e. they run optimization algorithms to decide when loads should start working, when they should finish and/or what is the power level they are allowed to request. However, while they may be allowed to perform time shifting of their own loads (depending on the user's restrictions), they are typically not allowed to shift aggregated loads that they receive from lower level MCDs.

If at some time instant(s), the aggregated load surpasses the maximum allowed upward power of a DB, the MCD must act, since it is not possible to assure the requested power. This may happen if several loads of different downward aggregators are scheduled to work at the same time. At this point, intermediate MCD aggregators should increase the cost of the energy associated with the overload periods and explicitly instruct lower level MCDs to reduce the power they are requesting for the time intervals where overloads happened. In both cases upper level MCDs will inform lower level MCDs about the required reschedule of their loads using a subsequent RI message, changing the associated power and cost vectors of (R, P, C), which will lead lower MCDs to make the necessary adjustments.

Each time a new load scheduling is requested, RA message is sent upwards, which triggers the exchange of RA and RI messages. This process stops when the top level MCD verifies that after several repetitions the cost does not improve. It then stops sending RA messages. Given this explanation, in the following we will describe this resource management mechanism in more detail.

3.2.3 Optimization Mechanism at MCDs

The task of MCDs leaves is to run the optimization algorithms that minimize the cost of electric consumption of various loads, shifting them in time or adjusting the power consumed, taking into consideration:

- The electricity tariffs;
- The power generated from local renewable sources;
- The time constraints imposed by the user for each device; and
- The Microgrid electric structure and constraints.

In order to do this, after receiving an RI message with (R, P, C) vectors, in our case the MCDs leaves run a GA which can be replaceable for any another kind of optimizer capable of doing schedulling based on an objective function. The optimizer targets the minimization of the objective function given by

$$F = \frac{1}{Q} \times \sum_{t \in T} \alpha(t) \times C(t) + \beta(t), \tag{3.1}$$

where:

- Q translates the quality assessment of the scheduling solution seen from a user perspective;
- T represents a set of time intervals;
- $C(t)$ translates the cost of energy for interval t (obtained from the RI message);
- $\alpha(t)$ is the difference in power between available renewable forecast and the power loads aggregate at instant t (Equation (3.2)); and
- $\beta(t)$ is a penalty restriction for overloads at instant t (Equation (3.3));

the objective function is called severall times returning several fitness (F) values. During GA evolution, which characterize each of the population chromossomes. When the stopping criteria is met, which in our case is the number of generations, the GA algorithm returns the best chromossome resultant from the GA inner work. This chromossome is a vector with epoch values corresponding to the start of operation solutions for every machine load in the the network.

In the case of our problem, the penalties that influence fitness are related to the placement of the solution, considering the horizontal and vertical layout of the load curves aggregate. A load curve that starts before the requested time or surpasses the end of operation request is penalized. The aggregate of load curves that vertically surpass the power limit for the circuit will also be penalized.
Equation (3.1) has only explicit penalties related to vertical misplacements, α and β, since:

$$\alpha(t) = \begin{cases} (P_{MCD}(t) - R'(t)) \times \Delta(t) & \text{if } P_{MCD}(t) > R'(t) \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

and

$$\beta(t) = \begin{cases} \infty & \text{if } P_{MCD}(t) > P_{max} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where:

- $P_{MCD}(t)$ translates the sum of loads power scheduled to work at the time interval t;
- P_{max} translates the maximum upward power limit of the DB of the MCD;
- t translates the time period associated with each R, P or C vector entrance;
- $R'(t)$ the remaining renewable power forecast, at instant t;
- $\Delta(t)$ is the pre-defined time interval step, in seconds; and

In our case, R' is obtained from the R vector received in the RI message using equation:

$$R'(t) = \begin{cases} R(t) & \text{if } \overline{P}(t) = 0 \text{ and } R(t) \geq 0 \\ \overline{P}(t) + R(t) & \text{if } P(t) > 0 \text{ and } R(t) \geq 0, \\ \overline{P}(t) \times (1 + R(t)) & \text{if R(t) < 0.} \end{cases} \tag{3.4}$$

where:

- $\overline{P}(t)$ represents the last requested aggregate; and
- $R(t)$ translates an estimation of the power generated by renewable sources at time interval t.

Equation (3.4) propagates in RI multicast messages triggered from the root, updating the information in all nodes about the remaining renewable forecast.
Function $\alpha(t)$, Equation (3.2), penalizes the aggregate that do not take full advantage of the remaining power from renewable sources. Surpassing the remaining renewable, not taking advantage of the cost zero possibility is penalized, otherwise it isn't. Function $\beta(t)$, Equation (3.3), penalizes abruptly a vertical aggregate that potentially provokes an overload, surpassing the power limit established by the local MCD. This is very important in preventing potential shedding of electrical circuits. Using these equations, the GA procedure defines when loads should be scheduled to start. However, this can only be made after a Microgrid level verification of the solution. In order to obtain it, leaf MCDs send an RA message to an upper level MCD containing the aggregated load vector, P_{MCD}.

3.2.4 Load Aggregation

After receiving RA messages from lower level MCDs, an upper layer MCD sums up the lower level P_{MCD} load vectors, generating an aggregated vector of requested power, P_r. If $P_r(t)$ is higher than the maximum power of the upward circuit (i.e., P_{max}) at some time interval t, then the values of P and C vectors stored in the MCD will be updated, for all the time intervals t where overloads happened, according to equations:

$$P(t) = \frac{P_{max}}{P_r(t)} \tag{3.5}$$

and

$$C(t) = C_0(t) + \left(\frac{1}{1+e^{k-n}}\right) \times \Delta T \tag{3.6}$$

where:

- $C(t)$ represents the new value of the energy cost at time interval t;
- C_0 is the vector with cost values obtained from the DSO;
- k is a constant used to adjust the responsiveness to repeated overloads;
- n represents the number of overloads that happened for time t; and
- ΔT is the difference of cost between two tariffs.

Equation (3.6) adds memory to the cost vector with the aim of reducing fibrillation, which happens when several loads continuously and in parallel oscillate around a small set of time intervals. ΔT and k can assume different values according to the level of the MCD.
After changing the power and cost vectors and before running its own optimization algorithms, $P_r(t)$ is upper limited to P_{max}, for all time instants t, where overloads occurred. Using the resulting power margin, the GA is used to decide where loads should work, setting the aggregated power vector P_{MCD}, of the intermediate MCD, which is sent to the upper layer MCD, through a subsequent RA message.

3.2.5 The (R, P, C) Computation

The top level MCD, will act like an intermediate node when RA messages arrive to it, with the exception that it will not generate a new RA message. Instead, after summing up the P_{MCD} load vectors received from lower MCDs and obtaining an aggregated requested power vector P_r, the top level MCD will change the (R, P, C) vectors to reflect the capability of the whole grid, before sending it down in a subsequent RI message.

Regarding the P and C vectors, they will be updated according to the procedure explained in Equations (3.5) and (3.6), only if and when overloads are expected to happen. For all the time instants t where overloads are not predicted to occur (i.e., $P_r(t) < P_{max}$) no information will be conveyed in the P vector of the RI message. This means that the P vector will not be used to perform a First-Come-First-Serve reservation procedure, which would tend to be unfair with the most recent requests. For those time intervals t where overloads occur, $P(t)$ will equally force a percentage of reduction in all power requests from lower MCDs (given by equation (3.5)).

Finally, the R vector will be obtained using:

$$R(t) = \begin{cases} P_G(t) - P_r(t) & \text{if } P_G(t) > P_r(t). \\ \frac{P_G(t) - P_r(t)}{P_r(t)} & \text{if } P_G(t) \leq P_r(t). \end{cases} \tag{3.7}$$

where P_G translates the forecasted generation vector of a renewable source. For those values where $R(t)$ is positive, it will convey the forecasted generated power that is still not being used by scheduled loads. However, when $P_r(t)$ surpasses $P_G(t)$, $R(t)$ will be negative and it will carry the ratio of power that all nodes are requesting beyond the forecasted $P_G(t)$. This value will be used by MCDs to estimate the ratio of power that is not being paid, as expressed by Equations (3.2) and (3.4).

As these RI messages go down the tree, P and C vectors may be changed by intermediate MCDs, according to their own stored state or capability. In terms of P, values that are sent down in a newly generated RI message are the lowest among the ones received in the RI message and the ones stored in the node. As for the C cost vector, the MCD will send the highest value among received and stores values.

The next section will outline the simulation platform.

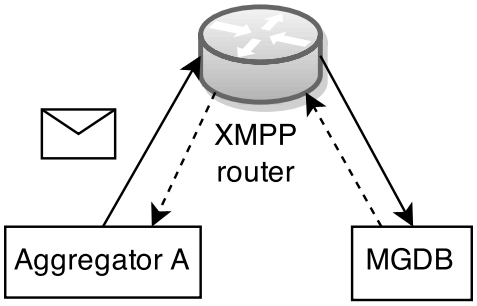


Figure 3.3: Generic representation of the communications between MCDs.

3.3 Protocol Implementation

To develop a simulation that demonstrates the applicability of our load management protocol capabilities for each MCD node to support the communications and the optimization algorithm part were considered as follows.

3.3.1 General Communications Implementation

Each object in our simulator (aggregators, generators and consumer appliances) constitutes an agent within a MAS. The simulator was implemented using the SPADE framework (see Section 2.3.3) (Gregori et al., 2006). SPADE was built around the XMPP/Jabber communication framework, being developed in Python. It has shown to be a particularly useful system in the implementation of MAS.
The simulator has been implemented as a distributed system where each of the previous defined objects (e.g., MCDs and electrical consumers/loads) are connected in a tree structure like the one described in Figure 3.1. Each agent implements the corresponding capabilities and behaviours as are the communication or the optimization actions.
Built around the capabilities provided by SPADE, every communication between agents passes through a XMPP router, where the messages are registered and dispatched.
Figure 3.3 represents a simple communication tunnel between a leaf MCD and a

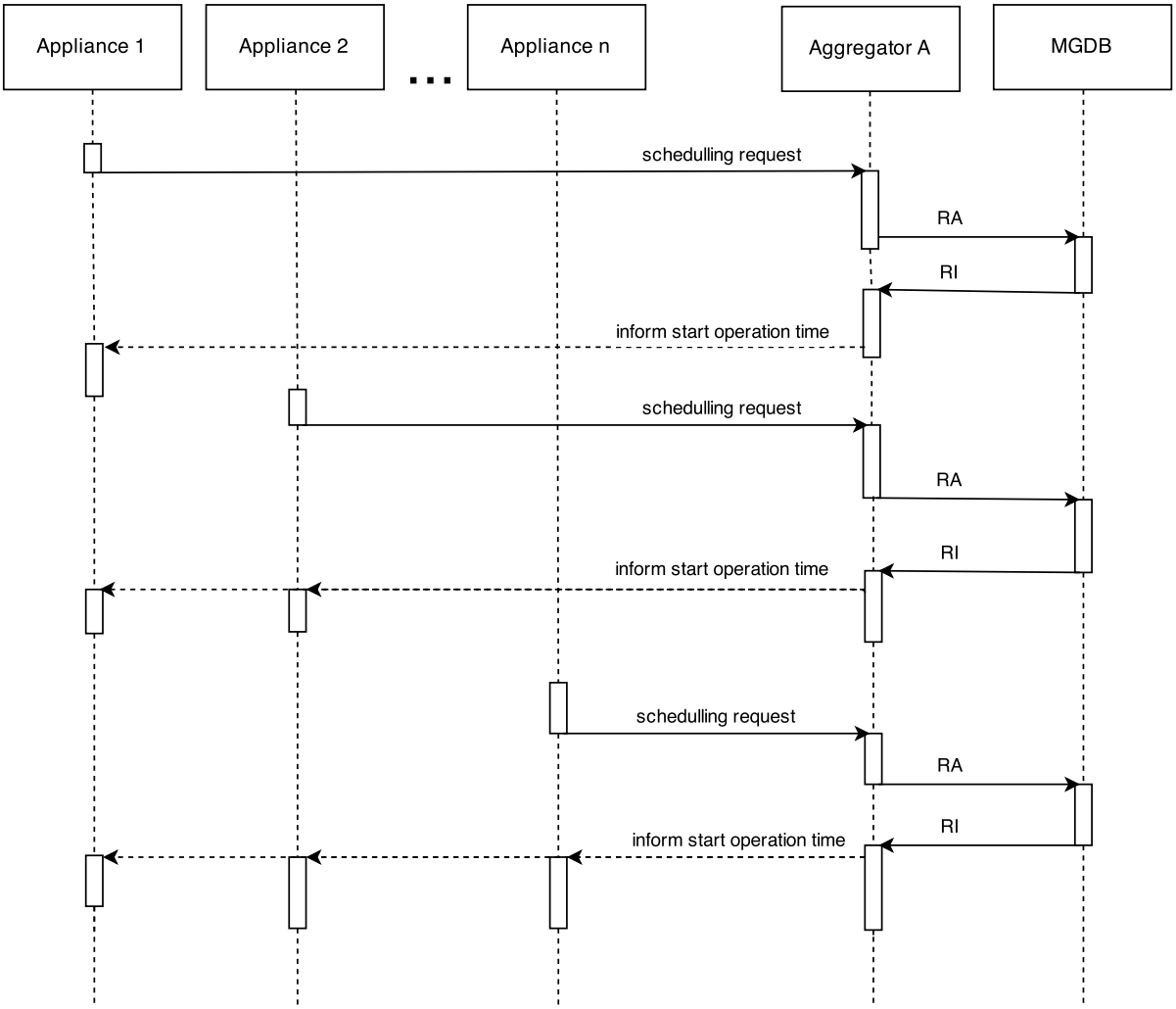


Figure 3.4: Communication diagram representing one layered tree structure, using one MCD node at the top ("MGDB"), and an unique layer of Aggregators immediately bellow. In this case only the communications in the branch which leds to the "Aggregator A" are represented. Every Appliance is subscribed to the subnetwork managed by the "Aggregator A".

higher level MCD. This tunneling is valid for any message exchange between generic agents under this system. This means that every message travelling between appliances and MCDs will first pass through the XMPP router (using its IP address), which register and handles routing.

Figure 3.4 presents a message diagram with several appliances communicating with a lower level MCD (or aggregator) which in turn communicates with a higher level MCD. Every aggregator at the lower level is passively waiting until a triggering

message from an appliance arrives and requests a decision about the time when it should start operating. After computing these requests this aggregator forwards a RA (resource allocation) message to the higher level aggregator containing the power curve resulting from the sum of all loads that have been scheduled to work. This upper level aggregator monitors all the aggregators in the layer bellow it. In the communication diagram shown in Figure 3.4 only two levels of aggregators are considered, and thus the RA message ceases at the upper level aggregator/MCD. The MCD top layer represents the core of the Microgrid, being able of monitoring all the aggregated loads and validating the decisions made from lower level aggregators. The criteria used to validate (or not) a solution depends on the compliance (or not) of the maximum power limit allowed for the distribution board where the aggregator is placed.
At the top level aggregator, RI messages travel down the tree, containing the resources that are still available. When a RI message arrives at the bottom layer aggregator, it runs its optimization algorithm and if any modifications were made it informs the appliances about the starting times for their operation.
For our communication states we have defined a simple ontology that consists in 6 keyword labels, that identify the intent of any message by its subject:

- *programme* announces a new request to be programmed. This subject triggers a new role of communications exchange to decide when the machine should start operating. With this subject the GA computation is activated to decide the placement of the load curves;
- *RA* informs the aggregator to handle resources allocation;
- *RI* informs the aggregator to handle resources information updates; and
- *generation_update* informs the aggregator about current energy production forecast from a generator;

The aggregators and MCDs have a decision menu procedure to handle any subject

that arrives to its message receiver. The message receiver is an event behaviour implemented as a subclass of the SPADE object (`SPADE.object.Behaviour`), which awakes when a new message arrives and stores the message subject ID and text body within a queue.

Also a SPADE periodic behaviour, designated as Message Dispatcher, is implemented as a subclass of a SPADE object which checks the message subject on queue first position every n seconds and decides which method to apply according the first subject stored in queue. Algorithm 1 presents the decision menu from message dispatcher to respond to any different message subject.

Algorithm 1: *Decision menu*

```
   Input: message queue
1  if message queue is not empty then
2      Get message subject ID;
3      Get information (Array of start operating requests, Array of load curves, costs
       vector, etc) from message body content;
4      switch subject ID do
5          case "generation_update"
6              Update R(t) vector with power from distributed generation;
7          case "programme"
8              Call GA to compute a new solution;
9          case "RA"
10             Attend RA message (see Algorithm 2);
11         case "RI"
12             Attend RI message (see Algorithm 3);
```

Among the subject ID's, two of them need to be explained in more detail, namely: the RA and the RI messages.

Following the procedures already presented in the previous sections, algorithms 2 and 3 describe the RA and RI message handling, respectively.

Algorithm 2: *RA message procedure*

Input: array of start operating requests, array of load curve arrays and (R,P,C) arrays.

Generate aggregate load curve array from the sum of instants of load curve arrays;
if *type of node is intermediate node* **then**
    Send RA message to uplink node, within aggregate load curve plus start operating time, and within (R,P,C) information updated according to Equations (3.5), (3.6) and (3.7);
if *type of node is root node* **then**
    Broadcast RI message to all down level aggregators, within (R,P,C) information updated according to Equations (3.5), (3.6) and (3.7);

Algorithm 3: *RI message procedure*

Input: (R,P,C) arrays

Receive (R, P, C) arrays and update $R'(t)$ according to equation (3.4);
if *type of node is intermediate node* **then**
    Forwards (R, P, C) information downlink;
if *type of node is leaf node* **then**
    if *P(t) array all set to zeros* **then**
        Stop the message exchange;
    else
        Call GA to obtain new start operating solutions;
        Generate load curve aggregate array;
        Send RA message upstream;

3.3.2 MCD Supplementary Optimization Algorithm Implementation

Regarding the optimization procedure, the PyEvolve framework (Perone, 2009), which implements the main GA algorithm, was used. The sample genome chromosomes were encoded as a list of integers (epoch times) where each element encodes the beginning (initial time) of a consumer machine program, e.g., $[t_1, t_2, ..., t_n]$. This means that the first appliance will start at t_1, the second appliance will start at t_2, and so on. Genetic Algorithms work combining selection, recombination and mutation operators. The selection pressure drives the population toward better solutions while recombination uses genes of selected parents to produce offspring that will form the next generation. Mutation is used to escape from local minima. Next we will present the operators used in this work.

GA mutators. The GA was configured with three mutation operators. The first one is the (1) Swap mutator which exchanges genes in a chromossome (see Figure 3.5, top). The swap of the genes in a chromossome (programmed loads starting times) varies the chromosome without loosing the "best placements" to move the loads. The second mutator is the (2) Integer Gaussian mutator which uses a random integer number influenced from a Gaussian distribution with deviation and mean parameters pre-configured, to slightly move the charges around their current position (Figure 3.5, middle) where ϵ is a sample from a Integer Gaussian distribution with mean $\mu = 0$ and standard deviation σ, $N(\mu, \sigma)$. This mutator also as the effect of not “loosing the best spots” to place the loads, since in general only slight changes are made to the initial time of the programs. In addition to the previous mutators, it was defined a (3) “Pseudo-Greedy” mutator that uses a roulette to influence the genome by giving better probabilities of moving charges to the lower cost tariff periods or to the predicted cost zero intervals due to renewable power forecast (see Figure 3.5 bottom, where t'_i is a value pseudo-randomly chosen so that

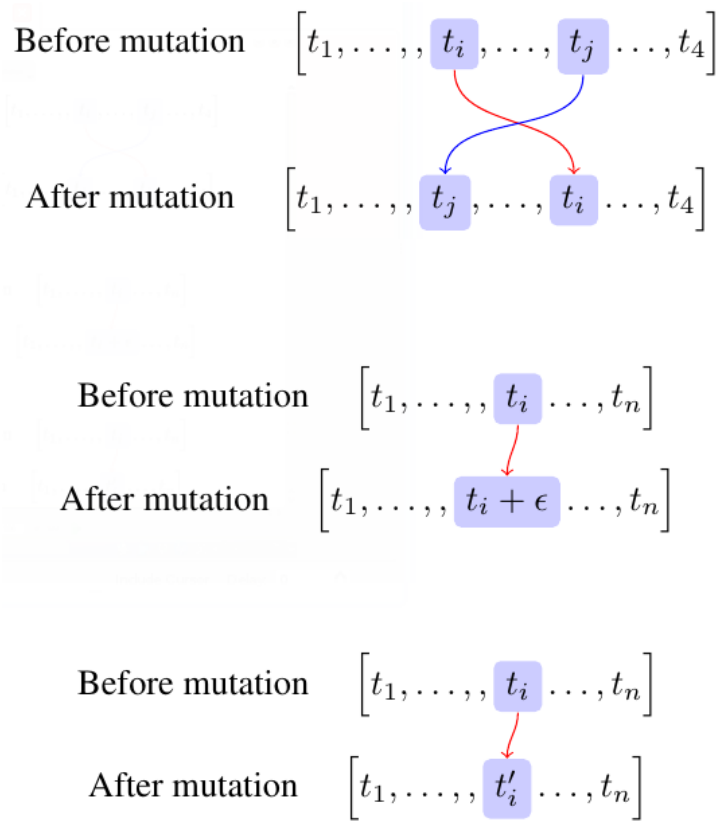


Figure 3.5: Genetic algorithm mutation operators: Swap mutator (top); the Integer Gaussian mutator (middle), ϵ is a sample from a Gaussian distribution with mean μ and standard deviation σ, $N(\mu, \sigma)$; and the "Pseudo-Greedy" mutator (bottom), t'_i is a pseudo-randomly chosen so that null or lower cost tariff periods have larger probability of being chosen.

zero or lower cost tariff periods have larger probability of being chosen). This "Pseudo-Greedy" mutator is more disruptive than previous ones since the offspring chromosomes can be significantly different from the parents.

GA Crossover. A single point crossover was used. The single point crossover gets two parent chromosomes and defines a cutting point. Data beyond that point in either chromosome is swapped between the two parent, producing the offspring (see Figure 3.6).

GA selection. The selection method chosen was the roulette wheel. In this case the parents are selected according to their fitness (see Equation (3.1)), i.e., chromosomes with better fitness get a larger chance of being selected. As the operator's name suggests, the process can be figured as a roulette wheel where the slices are bigger or

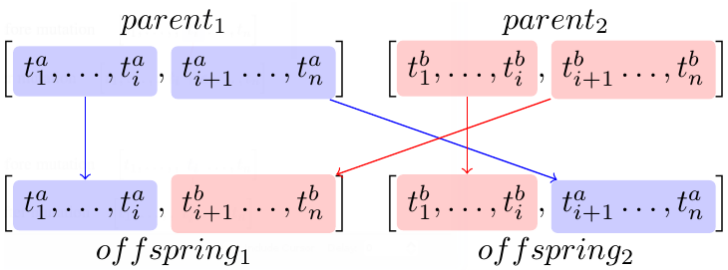


Figure 3.6: Genetic algorithm - single-point crossover.

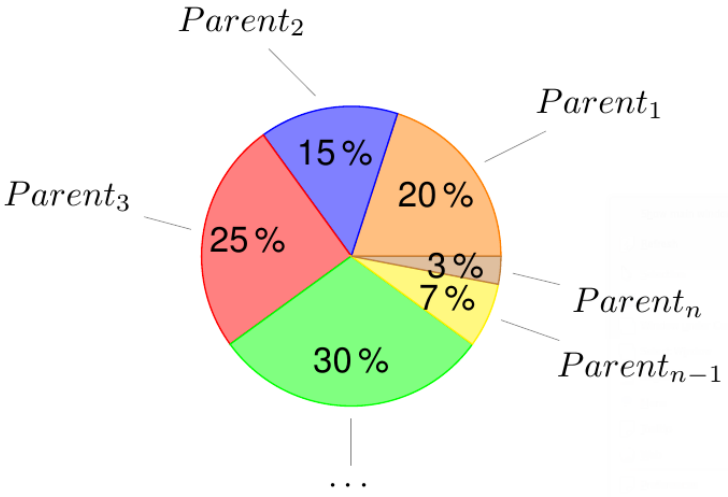


Figure 3.7: Genetic algorithm - roulette wheel.

smaller accordingly to the corresponding chromosome fitness function. Figure 3.7 shows a distribution of roulette slices, where $Parent_1$ has 20% chance of being selected, $Parent_2$ has 15% chance of being selected, etc.

Algorithm 4, describes the handling of programming requests from consumers. In more detail, when a "*programme*" message ID from an appliance or RA message from an aggregator imposes a reschedule, the GA is initialized and run. The GA's results are then collected to generate the load curve aggregated vector which is then send to the top level node.

The logic of how the GA evolves is described by the fluxogram in Figure 2.9.

Algorithm 4: *Procedure to handle a programme request*

Input: Start request, end request, load curve vector, appliance operation states.
Output: chromosome/best individual with start operating requests.

```
1 Initialize GA engine;
2 for current generation < maximum number of generations do
3 |   Evolve GA one step;
4 |   Elitism replacement, 2 individuals from previous population;
5 Collect GA best solution;
6 Generate load curve aggregate vector;
7 Send aggregate vector of load curves and start operating requests, identified as a RA
  message;
```

4

Simulation Platform and Results

In this chapter we will evaluate the optimization mechanism, proposed in this thesis, in a scenario that comprises 96 loads, spread over several distribution boards. We start by tuning the Genetic Algorithm to be able of reducing its computation time, before simulating the whole architecture.

4.1 Scenario Description

In the following scenario it is assumed that we are going to make a load scheduling in a search space of forecasted 24 hours (although it can be adapted to a wider one). We have also considered a power generation curve obtained from a solar

Tariff name	Period	Price
Void tariff (T0)	00:00 to 10:30 21:00 to 00:00	0.0955 €/kWh
Full tariff (T1)	13:00 to 19:30	0.1642 €/kWh
Peak tariff (T2)	10:30 to 13:00 19:30 to 21:00	0.2066 €/kWh

Table 4.1: Three-hourly tariff rates.

photovoltaic plant with a peak production of about 25 kW and a tariff with 3 price periods presented in Table 4.1.

In addition, we consider the possibility of selling the remaining energy from renewable sources to the grid, based on the Table 4.2. However selling prices do not weight optimization algorithm decisions. These table prices are accounted to compare the profit from non-optimized versus optimized solutions later in conclusions.

At 7:00 a.m., the lower level MCDs gradually start requesting the scheduling of the loads, representing a total demand of 263.5 kWh to be spread along the day, which corresponds to the scheduling of 96 loads, as shown in section 4.3.

Each of those loads belong to a set with six load profiles, numbered from C1 to C6, Figure 4.1. These load profiles representing a consumption load curve profile from an appliance.

The distribution of the 96 loads from the aforementioned scenario is shown by Table 4.3.

4.2 Tuning the GA

Implementing a distributed load scheduling mechanism in Microgrids using IoT devices places several challenges. One of these challenges is computational capability. In the first tests, performed a Beaglebone Black (AM335x 1GHz ARM® Cortex-A8 processor), the GA algorithm took nearly 9 minutes to compute a good

Period	**Price**
00:00 to 10:30	0.0955 €/kWh
00:00:00 to 01:00:00	0.05992 €/kWh
01:00:00 to 02:00:00	0.05330 €/kWh
02:00:00 to 03:00:00	0.050 €/kWh
03:00:00 to 04:00:00	0.04861 €/kWh
04:00:00 to 05:00:00	0.04573 €/kWh
05:00:00 to 06:00:00	0.04500 €/kWh
06:00:00 to 07:00:00	0.04500 €/kWh
07:00:00 to 08:00:00	0.04500 €/kWh
08:00:00 to 09:00:00	0.03940 €/kWh
09:00:00 to 10:00:00	0.04400 €/kWh
10:00:00 to 11:00:00	0.04400 €/kWh
11:00:00 to 12:00:00	0.04400 €/kWh
12:00:00 to 13:00:00	0.04469 €/kWh
13:00:00 to 14:00:00	0.04400 €/kWh
14:00:00 to 15:00:00	0.04400 €/kWh
15:00:00 to 16:00:00	0.04240 €/kWh
16:00:00 to 17:00:00	0.04000 €/kWh
17:00:00 to 18:00:00	0.03712 €/kWh
18:00:00 to 19:00:00	0.04400 €/kWh
19:00:00 to 20:00:00	0.04400 €/kWh
20:00:00 to 21:00:00	0.04500 €/kWh
21:00:00 to 22:00:00	0.04573 €/kWh
22:00:00 to 23:00:00	0.04500 €/kWh
23:00:00 to 00:00:00	0.04573 €/kWh

Table 4.2: Energy selling table, adapted from OMIE (2014).

Load / Aggregator	C1	C2	C3	C4	C5	C6
A	4	4	4	5	2	-
B1	2	2	2	-	1	3
B2	2	2	2	-	-	3
B3	-	2	2	-	-	3
B4	2	-	2	-	-	3
C1	3	3	2	-	4	-
C2	3	3	3	-	4	-
D	4	4	4	5	2	-

Table 4.3: Number of load requests by aggregator node.

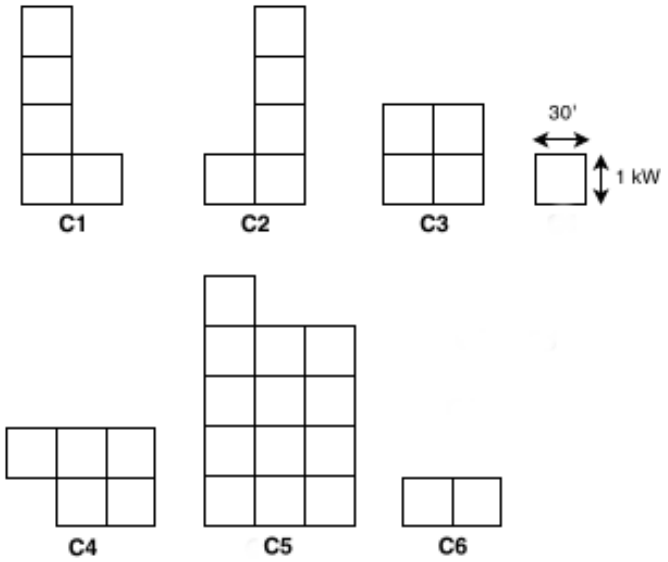


Figure 4.1: Set of load profiles used in the simulation tests. Each singular block represents a 1kW consumption during 30 minutes.

solution. This obviously was considered inappropriate, prompting for the necessary tuning of the GA algorithm, in order to converge to a good solution in a much shorter period of time.

Setting GA parameters is far from being a trivial task (Harik and Lobo, 1999). Also it is not correct to define any tuning procedure as a standards procedure. In fact, there have been conflicting advices about the best settings for these GA control parameters (Haines et al., 2012). Because the dynamics of the GA are so complex, it is difficult to evaluate the effect of all parameters simultaneously, and thus, many research studies have analysed the effect of one or two parameters separatedly (Harik and Lobo, 1999).

Initial tests made in our platform have shown that one important parameters affecting convergence was the population size which is one of the important parameters to consider in GA computation. Various results about the appropriate population size can be found in the literature. Researchers usually argue that a small population size could guide the algorithm to poor solutions and a large population size would expend more computation time in finding a solution (Roeva et al., 2013).

For each set of parameters, running them for a long number of generations trying to take conclusion on the convergence speed and solutions quality.

Thus in the following tests we have selected to adjust the population size, while setting the mutation probability to 0.05 and crossover rate to 0.9.

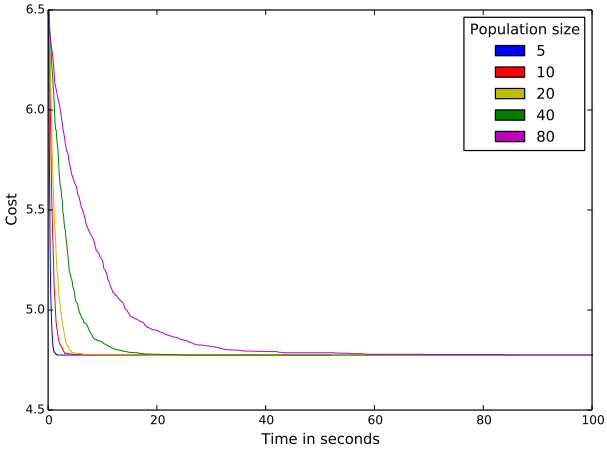
We have considered two request behaviours: simultaneous requests versus gradual requests. In simultaneous requests, we have considered that all scheduling requests, arriving from appliances, are made at the same time, while in gradual requests we have made them arrive periodically (i.e., distributed in time). As mentioned the length of the simulation (number of iterations) was set with large values, namely it was adjusted to 2000 generations for simultaneous requests, and 7000 generations for gradual arrival of requests.

In the next sections, we present the results from the tuning procedure considering different number of loads and population sizes, using an Intel(R) Core(TM) i7-4770 @3.4 GHz architecture device.

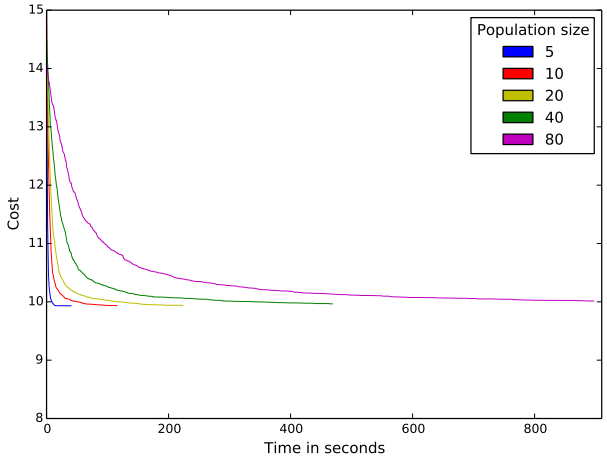
4.2.1 Tuning the Population Size for Simultaneous Requests, without using Distributed Generation

Figure 4.2 shows the results obtained after 30 trials for each set of parameters. For instance, Figure 4.2(a) presents the curves for the simultaneous scheduling of 25 loads. As can be seen, the cost converges in all cases to nearly 4.75. The blue curve, representing the population size of 5 individuals have shown to converge quicker, followed by the populations of 10, 20, 40 and 80. Similar behaviours were verified in 4.2(b) and 4.2(c), respectively for the simultaneous schedulling of 50 loads and 100 loads. In Figure 4.2(b), the cost converged to a value near to 10, for all the population sizes, and in Figure 4.2(c) the cost converged to 18, approximately.

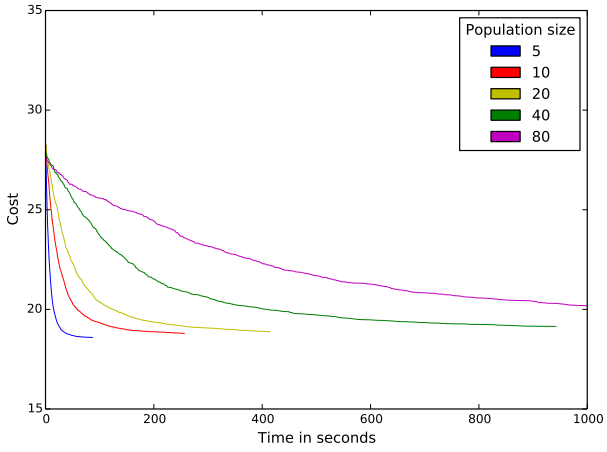
In general the results in Figure 4.2, show that the best population size was 5 individuals. Since the time required to schedule the loads increases substantially



(a) 25 loads



(b) 50 loads



(c) 100 loads

Figure 4.2: Cost evolution according to different population sizes and considering the simultaneous arrival of (a), (b) and (c) scheduling requests, without distributed generation.

with the population size while the costs are very similar.

4.2.2 Tuning the Population Size for Gradual Requests, without using Distributed Generation

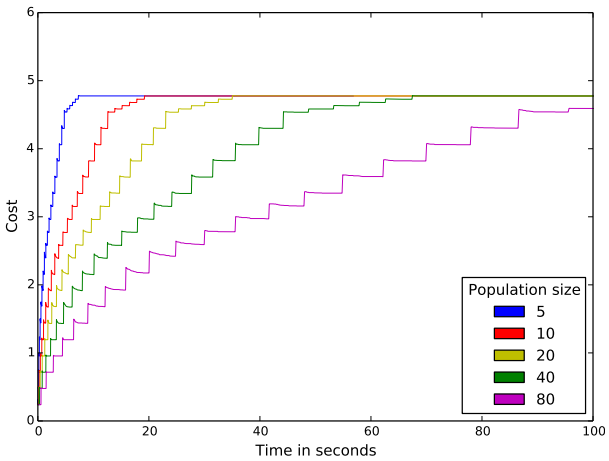
Figure 4.3 shows the cost evolution of the algorithm when 25, 50 and 100 loads gradually request their scheduling. As previously, the curve that converges more rapidly corresponds to the population of 5 individuals, while converging to similar costs.

4.2.3 Tuning the Population Size for Simultaneous Requests, using Distributed Generation

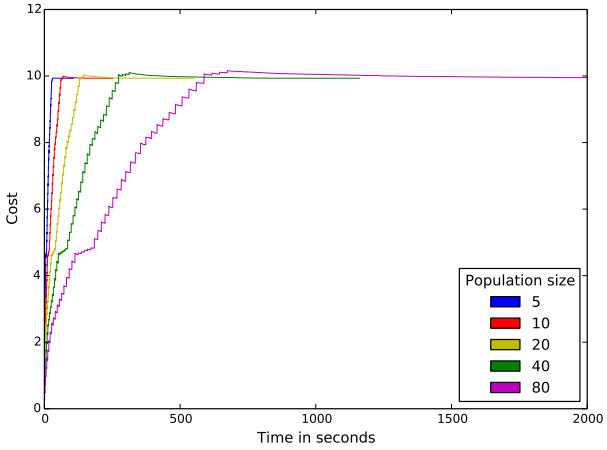
The tests presented in the previous sections did not consider that power coming from renewable energy sources was available at the MCD. Thus the scheduling of loads only considered the electrical tariffs. In the following tests we add a forecasted power curve, produced from a photovoltaic energy source, and repeated the tests.

Figure 4.4(a) presents the curves for simultaneous requests of a 25 loads scheduling. As we can see, the cost converged in all cases to 0.0. As in 4.2(a) the blue curve, representing the population size of 5 individuals have shown to converge quicker, followed by the populations of 10, 20, 40 and 80, sorted by convergence times. Similar behaviours were verified in 4.4(b) and 4.4(c), respectively for the requests of 50 loads and 100 loads.

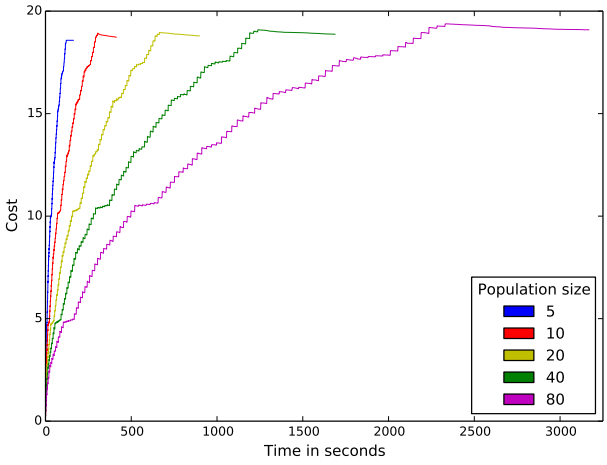
When comparing these results with the ones of Figure 4.2, it can be verified that the convergence times increase with the introduction of the renewable power source.



(a) 25 loads

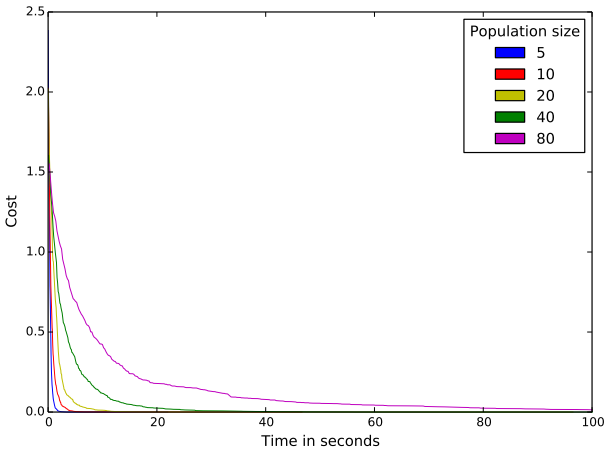


(b) 50 loads

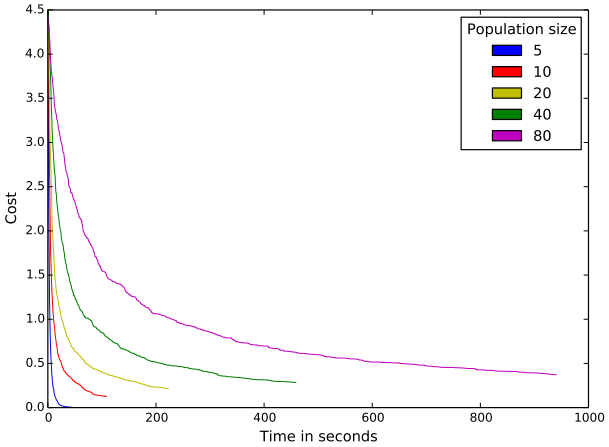


(c) 100 loads

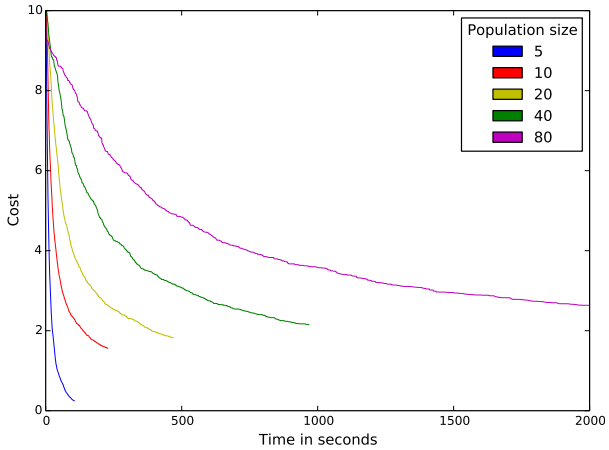
Figure 4.3: Costs evolution according to different population sizes for gradual scheduling requests of (a), (b) and (c) loads, without using distributed generation.



(a) 25 loads

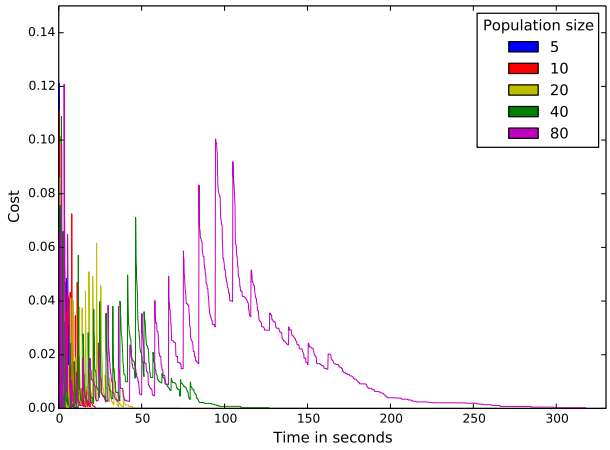


(b) 50 loads

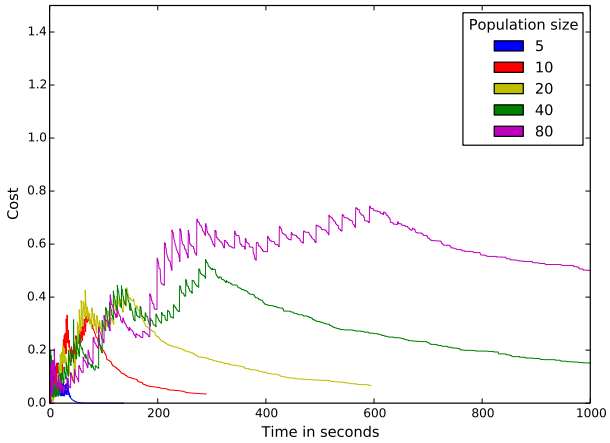


(c) 100 loads

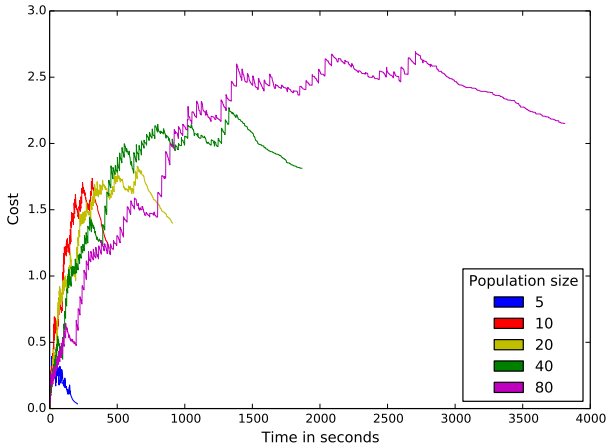
Figure 4.4: Costs evolution according to different population sizes and considering the simultaneous arrival of 25, 50 and 100 scheduling requests, with distributed generation.



(a) 25 loads



(b) 50 loads



(c) 100 loads

Figure 4.5: Cost evolution according to different population sizes for gradual scheduling requests of 25, 50 and 100 loads, with distributed generation.

4.2.4 Tuning the Population Size for Gradual Requests, Using Distributed Generation

Figure 4.5 shows the cost evolution when a renewable power source forecast is added to the system and when 25, 50 and 100 loads gradually request their

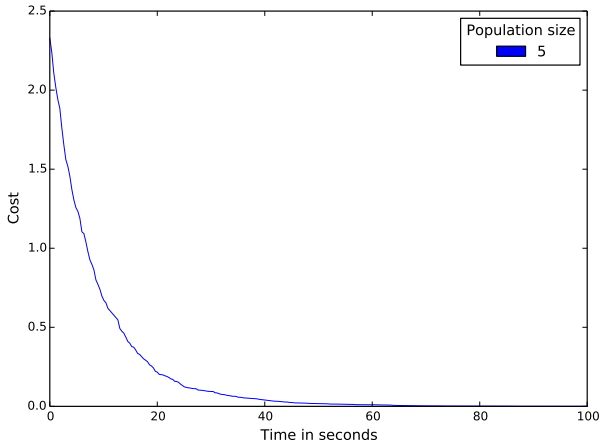
scheduling. As in Figure 4.3 these results show that the curve converging more rapidly corresponds to a population of 5 individuals, followed by 10, 20, 40, 80, sorted by increasing order of convergence time. These results confirm the tendency of faster results for smaller populations.

4.2.5 Discussion of the Tuning Results

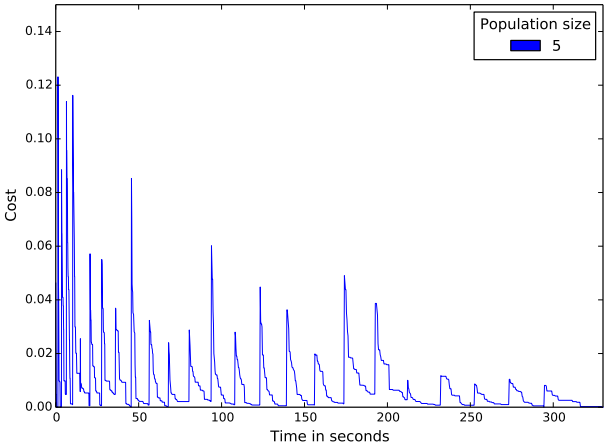
Each curves result from a mean value computed after 30 trials, performed in the associated scenario. The Analysis of Figures 4.2, 4.3, 4.4, 4.5, allow us to conclude that for the proposed scenarios the best population size is the one with 5 individuals, which equals the result achieved by Carroll (1996) and Alvarez (2002). In fact, in all tests this proved to be the best population size with lower time convergences to achieve a minimum cost values. Those values were accepted as close to an optimum solution, without a concrete proof, altough we have used a number of generations big enough to believe that solutions were not trapped in suboptimal solutions.
This results are quite important since we are looking at a schedulling solution which should run in near real time on a beaglebone or similar equipment.

4.2.6 Testing Speed of Population Size 5 in a Beaglebone Device, using Distributed Generation

In order to understand how long it would take to run the same algorithm using an beaglebone black (AM335x 1GHz ARM® Cortex-A8 processor), in this section we are going to analyse the convergence times considering the best set of parameters achieved in previous section (i.e. a population size of 5 individuals). The results are presented in Figure 4.6, which comparing 25 simultaneous requests with the same amount of gradual requests.
In Figure 4.6(a) scheduling 25 loads at the same time, the algorithm takes on average time of 37.81 seconds to find a solution which average cost of 0.051, and 107.03



(a) 25 loads, with all loads sending requests at the same time.



(b) 25 loads, with all loads sending requests gradually.

Figure 4.6: Results with population size 5 in beaglebone device.

seconds average to reach a completely cost zero solution. While the time taken to obtain these results is still too high, it is also true that in a real scenario load requests are typically sparse in time. This later scenario was evaluated in Figure 4.6(b).

In Figure 4.6(b) the cost was allways lower than 0.12, converging to a cost of 0.0 in nearly 12 seconds for each time a new load is added.

Obtaining optimal solutions in a range lower than one minute, using an apparatus of limited computational resources shows a good indicators for the practical feasibility of this system, taking into consideration that it shouldn't be larger the number of loads associated to each MCD where the beaglebone is to be placed.

4.3 Simulation Test results

Figure 4.7 presents typical scheduling solutions obtained by the algorithm for the 96 loads. As can be observed, the algorithm is able to schedule most of the loads to the phase where generation was available, while avoiding more expensive placements.

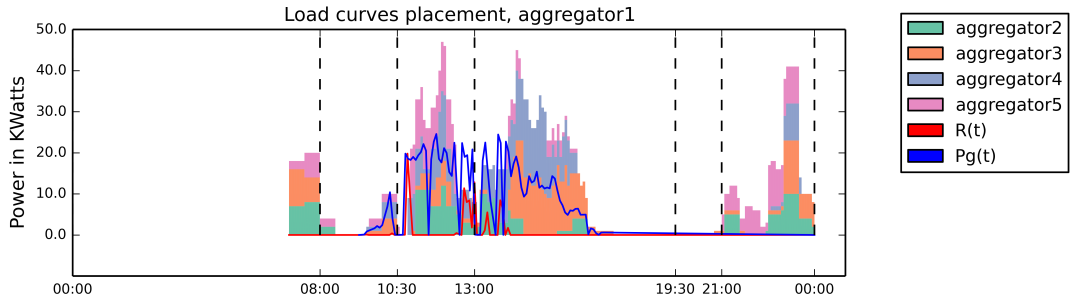


Figure 4.7: Load placement resulting from the distributed scheduling algorithm considering 96 load requests after 7 a.m..

Using the same parameters, systematic tests repeating 30 executions were performed comparing the proposed algorithm with a scenario without load scheduling, i.e. with random placement of the loads. The associated average and standard deviation results are shown in Table 4.4 collected from the optimized results and not optimized.

The results from these tests show that on average the proposed distributed load scheduling mechanism was able to achieve a reduction of 13.8% in the electricity costs compared to the random requests scenario.

The profits within optimization mechanism result on average 0.33 cents, while non-optimized consumption result on average 0.98 cents, taking into account prices from Table 4.2.

	Electricity cost	
Evaluation Parameter	**Optimized**	**Non-Optimized**
Average	20.56	23.86
Standard deviation	1.0	0.61
	Eletricity sale profit	
Evaluation Parameter	**Optimized**	**Non-Optimized**
Average	0.33	0.98
Standard deviation	0.14	0.23

Table 4.4: Cost-Profit results of the performed systematic tests.

5

Conclusions and Future Work

This thesis presents and evaluates a new Microgrid Resource Management Protocol. The proposed protocol includes the following features: (1) considers aspects of the electrical grid protection such as preventing the shedding of the electrical circuits caused by overloads; (2) takes into account the use of energy from renewable sources; (3) targets the economic cost minimization according to a set of tariffs defined by a DSO and (4) already meets some essential requirements for the integrity and management of a microgrid.

The mentioned protocol is already being implemented in real equipment. Using a lower resourced device suitable for the IoT purpose of small devices, such as a beglebone black (AM335x 1GHz ARM® Cortex-A8 processor) and using an effort of

25 loads requesting at the same time, the algorithm was able to dispatch a cost zero solution in an average of 107.03 seconds average, considering power from renewable sources. However a considerably good solution of 0.05 is achieved in an average of 37.81 seconds. This response rate is already acceptable for the purpose of a single aggregator. In a real home scenario it is not expected that 25 loads would request schedullings at the same time. If that was the case, measures could be taken, like increasing the computational capacities. Also the protocol was successfully implemented using a tree of three beaglebone devices, although no systematic test results are shown in this current work.

Results from extensive tests presented in Chapter 4 show that significant electricity cost reductions can be achieved using this methodology, when compared to non-optimized requests.

The advantage of selling energy to the eletricity provider is higher when the consumption pattern is not optimized, as the energy comes from the remaining energy from renewable sources and the optimization tries to fill the spaces on the edge of the distributed generation load profile where the cost is zero. However the profit of non-optimized consumption doesn't compensate the average cost reduction achieved by optimizing the consumption patterns.

Altogether we are sure that our solution is unique and it could be applied in an electrical grid. In the future we aim to successfully apply our protocol for a testbed in a real scenario, which is a work in progress. Also as future work, improvements or the replacement of the algorithm is a possible hypothesis, since the protocol is not algorithm dependent.

5.1 Publications and Contributions Related to the Thesis

The research work presented in this thesis contributed to four publications, namely three conferences and one patent pending request. The publications are annexed to this work by its chronological order.

6
Bibliography

Advisory, C. (2011). IEEE Standard for Ubiquitous Green Community Control Network Protocol. *IEEE Std 1888-2011*, pages 1–65.

Alvarez, G. (2002). Can we make genetic algorithms work in high-dimensionality problems. *SEP–112*, pages 195–212.

Aranda, G., Palanca, J., Espinosa, A., Terrasa, A., and García-Fornes, A. (2006). Towards developing multi-agent systems in ADA. In *Reliable Software Technologies–ADA-Europe 2006*, pages 131–142. Springer.

Brucoli, D. M. and Kevin O'Halloran, A. (2014). Microgrids: reliable power while integrating renewable generation. `http://www.engineersjournal.ie/microgrids-reliable-power-while-integrating-renewable-generation/`. Accessed 23th of September, 2014.

Bullis, K. (2012). How power outages in india may one day be avoided. `http://www.technologyreview.com/news/428666/how-power-outages-in-india-may-one-day-be-avoided/`.

Calderwood, S., Liu, W., Hong, J., and Loughlin, M. (2013). An architecture of a multi-agent system for scada.

Carroll, D. L. (1996). Genetic algorithms and optimizing chemical oxygen-iodine lasers. *Developments in theoretical and applied mechanics*, 18(3):411–424.

Chowdhury, S. and Crossley, P. (2009). *Microgrids and active distribution networks*. The Institution of Engineering and Technology.

Christensen, L. R. and Greene, W. H. (1976). Economies of scale in US electric power generation. *The Journal of Political Economy*, pages 655–676.

Davidson, E. M., McArthur, S. D., McDonald, J. R., Cumming, T., and Watt, I. (2006). Applying multi-agent system technology in practice: automated management and analysis of scada and digital fault recorder data. *Power Systems, IEEE Transactions on*, 21(2):559–567.

Dimeas, A. and Hatziargyriou, N. (2004). A multi-agent system for microgrids. In *Methods and applications of artificial intelligence*, pages 447–455. Springer.

Eduardo, J., Cardoso, P., and Monteiro, J. (2013). Gestão de cargas numa micro grid utilizando algoritmos genéticos. *13ª Conferência sobre Redes de Computadores, Leiria*, pages pp. 13–18.

Feroze, H. (2009). *Multi-agent systems in microgrids: Design and implementation*. PhD thesis, Virginia Polytechnic Institute and State University.

FIPA00002, Supersedes (2000). Fipa agent management specification. *Change*.

Gellings, C. W. and Smith, W. M. (1989). Integrating demand-side management into utility planning. *IEEE*, 77(6):908–918.

Gregori, M. E., Cámara, J. P., and Bada, G. A. (2006). A jabber-based multi-agent system platform. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 1282–1284. ACM.

Guo, Y., Wu, J., and Long, C. (2013). Agent-based multi-time-scale plug load energy management in commercial building. In *Control and Automation (ICCA), 2013 10th IEEE International Conference*, pages 1884–1889. IEEE.

Haines, A., Mills, K., and Filliben, J. (2012). Determining relative importance and best settings for genetic algorithm control parameters. *NIST Pub*, 912472:1–22.

Harik, G. R. and Lobo, F. G. (1999). A parameter-less genetic algorithm. In *GECCO*, volume 99, pages 258–267.

Indexmundi (2011). Electricity production by source. `http://www.indexmundi.com/facts/visualizations/electricity-production/`. Accessed in August 5th, 2014.

Jian, Z., Qian, A., Chuanwen, J., Xingang, W., Zhanghua, Z., and Chenghong, G. (2009). The application of multi agent system in microgrid coordination control. In *Sustainable Power Generation and Supply, 2009. SUPERGEN'09. International Conference on*, pages 1–6. IEEE.

Jimeno, J., Anduaga, J., Oyarzabal, J., and de Muro, A. G. (2011). Architecture of a microgrid energy management system. *European Transactions on Electrical Power*, 21(2):1142–1158.

Kok, J., Warmer, C., and Kamphuis, I. (2005). Powermatcher: multiagent control in the electricity infrastructure. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 75–82. ACM.

Kulasekera, A., Gopura, R., Hemapala, K., and Perera, N. (2011). A review on multi-agent systems in microgrid applications. In *Innovative Smart Grid Technologies-India (ISGT India), 2011 IEEE PES*, pages 173–177. IEEE.

Kumar, A. (2013). Using microgrids in india to prevent grid collapses. `http://sauryaenertech.com/microgrids-in-india/`.

Laboratories, S. N. (2012). Scada history. `http://energy.sandia.gov/?page_id=6972`.

Legrand (2009). *Electrical energy supply, Power guide*. Legrand, book 03 edition.

Liu, C.-C., Jung, J., Heydt, G. T., Vittal, V., and Phadke, A. G. (2000). The strategic power infrastructure defense (spid) system. a conceptual design. *Control Systems, IEEE*, 20(4):40–52.

Logenthiran, T. and Srinivasan, D. (2009). Short term generation scheduling of a microgrid. In *TENCON 2009-2009 IEEE Region 10 Conference*, pages 1–6. IEEE.

Logenthiran, T., Srinivasan, D., Khambadkone, A., and Aung, H. (2010a). Multi-agent system (mas) for short-term generation scheduling of a microgrid. In *Sustainable Energy Technologies (ICSET), 2010 IEEE International Conference on*, pages 1–6. IEEE.

Logenthiran, T., Srinivasan, D., Khambadkone, A., and Aung, H. (2010b). Scalable multi-agent system (mas) for operation of a microgrid in islanded mode. In *Power Electronics, Drives and Energy Systems (PEDES) & 2010 Power India, 2010 Joint International Conference on*, pages 1–6. IEEE.

Logenthiran, T., Srinivasan, D., and Shun, T. Z. (2012). Demand side management in smart grid using heuristic optimization. *Smart Grid, IEEE Transactions on*, 3(3):1244–1252.

Luck, M., McBurney, P., and Preist, C. (2003). *Agent technology: Enabling next generation computing (a roadmap for agent based computing)*. AgentLink/University of Southampton.

Luck, M., McBurney, P., Shehory, O., and Willmott, S. (2005). Agent technology: computing as interaction (a roadmap for agent based computing).

McArthur, S. D., Davidson, E. M., Catterson, V. M., Dimeas, A. L., Hatziargyriou, N. D., Ponci, F., and Funabashi, T. (2007a). Multi-agent systems for power engineering applications—part i: Concepts, approaches, and technical challenges. *Power Systems, IEEE Transactions on*, 22(4):1743–1752.

McArthur, S. D., Davidson, E. M., Catterson, V. M., Dimeas, A. L., Hatziargyriou, N. D., Ponci, F., and Funabashi, T. (2007b). Multi-agent systems for power engineering applications—part ii: technologies, standards, and tools for building multi-agent systems. *Power Systems, IEEE Transactions on*, 22(4):1753–1759.

Mohamed, F. and Koivo, H. (2008). Multiobjective genetic algorithms for online management problem of microgrid. *International Review of Electrical Engineering*, 3(1).

Momoh, J. (2012). *Smart grid: fundamentals of design and analysis*, volume 63. John Wiley & Sons.

Monteiro, J., Cardoso, P., Serra, R., and Fernandes, L. (2014). Evaluation of the human factor in the scheduling of smart appliances in smart grids. In Stephanidis, C. and Antona, M., editors, *Universal Access in Human-Computer Interaction. Aging and Assistive Environments*, volume 8515 of *Lecture Notes in Computer Science*, pages 537–548. Springer International Publishing.

Morgan Hill, C. (2013). Openadr 2.0 profile specification, a profile. `http://www.openadr.org/specification`. Accessed 15th of August 2014.

Nagasaka, K., Ando, K., Xu, Y., Takamori, H., Wang, J., Mitsuta, A., Saito, O., and Go, E. (2012). A research on operation planning of multi smart micro grid. In *Advanced Mechatronic Systems (ICAMechS), 2012 International Conference on*, pages 351–356. IEEE.

Nagata, T. and Sasaki, H. (2002). A multi-agent approach to power system restoration. *Power Systems, IEEE Transactions on*, 17(2):457–462.

Nishioka, A. S. (2014). Power system stabilization. In *Future Renewable Energy and Smart Grid Technologies*. CIGRE TNC Technical Seminar, Hitachi, Ltd.

OASIS Energy Interoperation Technical Committee OASIS (2012). Oasis energy interoperation version 1.0. OASIS archives.

OMIE (2014). Omi-polo español s.a. (omie),. `http://www.omie.es/files/flash/ResultadosMercado.swf`. Accessed in April 1st, 2014.

Palanca, J. (2014). spade2, smart python agent development environment. `https://code.google.com/p/spade2/`. Acessed in 18th of August 2014.

Perone, C. (2009). Pyevolve framework. `http://pyevolve.sourceforge.net/`.

Phung, D. (1987). Theory and evidence for using the economy-of-scale law in power plant economics. Technical report, PAI Corp., Oak Ridge, TN (USA).

Piette, M. A. (2009). Open automated demand response communications specification (version 1.0). *Lawrence Berkeley National Laboratory*.

Pipattanasomporn, M., Feroze, H., and Rahman, S. (2009). Multi-agent systems in a distributed smart grid: Design and implementation. In *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*, pages 1–8. IEEE.

Praça, I., Ramos, C., Vale, Z., and Cordeiro, M. (2003). Mascem: a multiagent system that simulates competitive electricity markets. *Intelligent Systems, IEEE*, 18(6):54–60.

Razali, N. M. and Hashim, A. (2010). Profit-based optimal generation scheduling of a microgrid. In *Power Engineering and Optimization Conference (PEOCO), 2010 4th International*, pages 232–237. IEEE.

Roche, R., Blunier, B., Miraoui, A., Hilaire, V., and Koukam, A. (2010). Multi-agent systems for grid energy management: A short review. In *IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society*, pages 3341–3346. IEEE.

Roeva, O., Fidanova, S., and Paprzycki, M. (2013). Influence of the population size on the genetic algorithm performance in case of cultivation process modelling. In *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on*, pages 371–376. IEEE.

Romero, J. J. (2012). Blackouts illuminate india's power problems. *Spectrum, IEEE*, 49(10):11–12.

Schneider Electric (2013). *Electrical installation guide 2013. According to IEC international standards*.

Shahidehpour, M. and Alomoush, M. (2002). Restructured electric power systems: Operation, trading, and volatility [book review]. *Computer Applications in Power, IEEE*, 15(2):60–62.

Shoham, Y. and Leyton-Brown, K. (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

Sujil, A., AGarwal, S. K., and Kumar, R. (2014). Centralized multi-agent implementation for securing critical loads in pv based microgrid. *Journal of Modern Power Systems and Clean Energy*, 2(1):77–86.

Sweeney, J. L. (2002). The california electricity crisis: Lessons for the future. *BRIDGE-WASHINGTON-*, 32(2):23–31.

Sweeney, J. L. (2008). *The California electricity crisis*. Hoover Press.

UGCCNET-CS, W. (2011). Deployments, utility-scale smart meter, plans & proposals, sep. 2011. *Edison Foundation, Institute for Electric Efficiency*.

van der Meulen, J. R. R. (2013). Gartner says the internet of things installed base will grow to 26 billion units by 2020. `http://www.gartner.com/newsroom/id/2636073`. Accessed in 24th of July, 2014.

Vasseur, J.-P. and Dunkels, A. (2010). *Interconnecting smart objects with ip: The next internet*. Morgan Kaufmann.

Veitch, C. K., Henry, J. M., Richardson, B. T., and Hart, D. H. (2013). Microgrid cyber security reference architecture. Technical report, Sandia National Laboratories (SNL-NM), Albuquerque, NM (United States).

Wroclawski, J. (1997). The use of rsvp with ietf integrated services. *Network Working Group*.

Xiao, Z., Li, T., Huang, M., Shi, J., Yang, J., Yu, J., and Wu, W. (2010). Hierarchical mas based control strategy for microgrid. *Energies*, 3(9):1622–1638.

Zambonelli, F., Jennings, N. R., and Wooldridge, M. (2003). Developing multiagent systems: The gaia methodology. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 12(3):317–370.

Zhang, L., Berson, S., Herzog, S., and Jamin, S. (1997). Resource reservation protocol (rsvp)–version 1 functional specification. *Resource*.

Zhou, X., Gu, Y., Ma, Y., Cui, L., and Liu, S. (2010). Hybrid operation control method for micro-grid based on mas. In *Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on*, volume 1, pages 72–75. IEEE.

ZigBee Alliance (2013). *Smart Energy Profile 2 Application Protocol Standard*. Number ZigBee Public Document 13-0200-00. ZigBee Alliance.

A

Articles Accepted for Conferences

A.1 *"Gestão de Cargas numa Micro Grid Utilizando Algoritmos Genéticos", CRC 2013, Leiria (Portugal)*

Gestão de Cargas numa *Micro Grid* Utilizando Algoritmos Genéticos

Jorge Eduardo
Instituto Superior de Engenharia
Universidade do Algarve
Faro, Portugal 8005-139
Email: jmeduardo@ualg.pt

Pedro Cardoso
Instituto Superior de Engenharia
Universidade do Algarve
Faro, Portugal 8005-139
Email: pcardoso@ualg.pt

Jânio M. Monteiro
Instituto Superior de Engenharia
Universidade do Algarve
Faro, Portugal 8005-139
Email: jmmontei@ualg.pt

***Resumo*—O sistema de produção de energia elétrica tem até hoje caraterizado-se por um esquema de geração centralizado, unidirecional e contendo medidas de incentivo e controlo sobre a procura de consumo. Com o recente desenvolvimento de novos equipamentos de geração e controlo verifica-se uma mudança gradual deste modelo para outro denominado de *Smart Grid*, em que a geração de eletricidade distribuída, geralmente de baixa potência, começa a integrar-se de uma forma eficaz com a rede já existente e não apenas com a entrega da eletricidade produzida. Uma das medidas que pode ser implementada para que se promova um ajuste no lado do consumo por forma a que este se nivele a uma potência produzida que varia em função de fatores ambientais, baseia-se na implementação de tarifários variáveis. Mas para que tal venha a ser possível é necessário desenvolver uma série de equipamentos que consigam comunicar entre si e com a rede de distribuição, utilizando protocolos normalizados, gerindo os consumos de uma forma otimizada em função dos tarifários previstos e refletindo as preferências dos utilizadores. Neste âmbito, o objetivo deste artigo é o de definir e avaliar uma solução de otimização aplicada à gestão de cargas em *Smart Grids* baseada em algoritmos genéticos, que em função de restrições impostas pelo utilizador defina o escalonamento de funcionamento de diversas cargas.**

Palavras-chaves: Smart Grids, Home Grids, Micro Grids, Algoritmos Genéticos.

I. Introdução

Nas redes de produção e distribuição de energia elétrica encontramos atualmente um dinamismo bastante diferente daquele que encontrávamos há uns anos atrás, resultante em grande parte da crescente introdução de fontes de energia renováveis. Ao contrário da distribuição elétrica clássica, na produção obtida a partir de energias renováveis, os fatores ambientais de que elas dependem causam variações na geração difíceis de controlar, que por sua vez, são a causa de ineficiências e desadaptações de diversa ordem.

Diversas soluções têm sido apontadas neste âmbito. Algumas propostas apontam para que se promova um ajuste no lado do consumo por forma a que este se nivele à potência produzida. Outras soluções apontam para que se armazene a energia produzida em excesso em determinado momento, recorrendo entre outros, às baterias dos automóveis elétricos. Há ainda soluções que optam por tarifar a produção para autoconsumo privado, de sistemas ligados à rede energética, numa tentativa de limitar a generalização destes equipamentos e reduzir os riscos de desestabilização, tal como aconteceu recentemente em Espanha [1].

Se por um lado a produção desregulada pode ser vista como um risco, o ajuste do consumo em momentos de pico é visto como algo desejável por parte dos operadores de eletricidade. Assim, a primeira versão do protocolo Open Automated Demand Response (openADR) [2] surgiu como uma das medidas que poderia minimizar os apagões verificados na crise energética vivida no ano 2000, no estado da Califórnia.

Em termos de tarifários de eletricidade, existe uma clara desadaptação entre os preços dos tarifários dos utilizadores e os custos marginais de produção [3]. Uma solução que poderia resolver este problema passa pela introdução de tarifários dinâmicos, que estão neste momento a ser testados e implementados em várias regiões dos Estados Unidos da América, a par da instalação de 27 milhões de contadores inteligentes já efetuada [4].

Assim, pretende-se comunicar aos equipamentos e pessoas o tarifário em vigor para que estas decidam em consonância. Neste contexto, as tecnologias associadas às *Smart Grids* surgem como a solução que a médio/longo prazo permitirão a gestão eficiente das redes energéticas equipadas de geração renovável, através de equipamentos que combinem as redes de distribuição elétrica, as redes de comunicação, as redes de sensores e as tecnologias de informação, criando uma plataforma de configuração fácil e acesso ubíquo.

Neste sentido, o objetivo de criação de um sistema de gestão de energia é o de implementar um conjunto de equipamentos que comuniquem entre si, chamados objetos inteligentes [5], que atuando num sistema de controlo otimizado suportado no conceito da *Internet of Things* (IoT), permitam um melhor aproveitamento da energia produzida pelas energias renováveis e a melhoria da gestão energética de edifícios.

No âmbito dos protocolos que podem ser utilizados para gestão energética salientam-se as recentes especificações das arquiteturas protocolares *Smart Energy Profile* – versão 2 (SEP 2.0) [6], IEEE 1888 [7] e do protocolo OpenADR 2.0 [8]. Qualquer arquitetura protocolar que venha a ser implementada não pode por isso ser criada sem ter em conta essas soluções. A um nível mais baixo salientam-se também o conjunto de protocolos IPv6 e 6LoWPAN que cada vez mais assumem importância em redes com elevado número de objetos inteligentes

capazes de comunicar entre si, sobre redes com ou sem fios.

Por fim, suportando-se nas tecnologias de informação, o sistema deverá fazer do utilizador o elemento chave da solução final. Para tal, deverá ser criada uma interface que lhe permita de uma forma fácil aceder, configurar e modificar sempre que seja necessário as suas preferências. Por exemplo cabe ao utilizador especificar que a temperatura da água quente não deve baixar dos 30 graus, ou que o programa da máquina de lavar a louça deve terminar antes das 19:00. Ao utilizador cabe a definição das restrições a que o sistema deverá responder, sendo o primeiro liberto da gestão do sistema. Essa gestão deverá ser feita utilizando algoritmos de otimização que, tendo em conta os tarifários e comunicando com os diversos objetos inteligentes de uma *Home/Micro Grid* decidam que cargas devem entrar em funcionamento e em que alturas o devem fazer. É este o âmbito deste artigo.

O resto do artigo tem a seguinte estrutura. A secção II apresenta o conjunto de protocolos de comunicação atualmente em definição para Smart Grids. A secção III apresenta os algoritmos de otimização considerados neste artigo, nomeadamente os algoritmos genéticos. A secção IV faz a descrição e formulação do problema do controlo de cargas. A secção V descreve os testes e ensaios efetuados. Por fim, a secção VI conclui o artigo apontando possíveis desenvolvimentos futuros.

II. Protocolos de Comunicação em Smart Grids

A possibilidade de os utilizadores gerirem os seus consumos de energia em função da produção é uma característica crítica das *Smart Grids* sendo a base de inovação, novos produtos e serviços. Por forma a suportar esta capacidade, a comunicação entre os diversos dispositivos, tais como contadores, eletrodomésticos, veículos elétricos, sistemas de gestão de energia e recursos energéticos distribuídos (incluindo energias renováveis e armazenamento) deve ocorrer utilizando procedimentos abertos, seguros e normalizados. Neste âmbito, foram recentemente definidos vários protocolos.

Um desses protocolos, o *Smart Energy Profile* resulta da colaboração entre *low-power ZigBee*, o Wi-Fi e a *Home Plug power-line technologies*, construindo uma arquitetura de gestão de energia para *Micro Grids*, suportada em redes IP. Assim, ao contrário do SEP 1.0 que considerava apenas o ZigBee como forma de interligação entre objetos inteligentes, nesta última versão as comunicações já incluem ligações Wi-Fi e PLC. O *Smart Energy Profile* foi desenhado para implementar uma arquitetura REST, tendo como base as ações do GET, HEAD, PUT, POST e DELETE, complementadas com um mecanismos de subscrição leve. Apesar do SEP se poder suportar em qualquer protocolo que implemente uma comunicação REST (por exemplo o *Constrained Application Protocol* (CoAP) [9]) o HTTP é considerado a base para a interoperabilidade das aplicações.

Em março de 2011, o *Institute of Electrical and Electronics Engineers* (IEEE), a maior associação profissional do Mundo anunciou a aprovação e publicação da norma *Standard for Ubiquitous Green Community Control Network Protocol* (IEEE 1888™), no âmbito da *Ubiquitous Green Community Control Network Protocol* (UGCCNet). Com origem na China, a norma IEEE 1888 assume-se como uma norma global no âmbito da IoT que tem como objetivo a eficiência energética, através da gestão das energias renováveis (dita *green energy*), através da comunicação utilizando protocolos Internet e as Tecnologias de Informação e Comunicação.

Como fundamento da especificação das normas, está a intenção de criar um novo sistema de controlo fácil de utilizar, largamente divulgado e inteligente, incluindo: a fácil monitorização do consumo, análise dos desperdícios energéticos e sugestões de melhoria, controlo automático de índices de conforto (CI), ajuste produção-consumo, entre outros.

Várias sub-normas IEEE 1888 estão atualmente em definição, identificadas como IEEE 1888.x, com x a variar entre 1 e 4.

O protocolo de aplicação *Open Automated Demand Response* (OpenADR) versão 2.0 [8], é uma evolução e extensão da primeira versão desenvolvida pela *Demand Response Research Center* no Lawrence Berkeley National Laboratory. OpenADR 2.0 é suportado pela aliança industrial OpenADR, tendo sido desenvolvida como parte da norma *OASIS Energy Interoperatio*n 1.0 publicada em Fevereiro de 2012.

Uma característica que diferencia o OpenADR de outras arquiteturas que implementam o *Demand Response* (DR) automatizado é a de que os pedidos não contêm informação que especifique os dispositivos ou operações que devem ser alteradas ou paradas. O OpenADR apenas notifica os dispositivos para que estes reduzam o consumo – quer utilizando pedidos específicos ou através de um aumento no preço da eletricidade. As respostas dos equipamentos dependem sempre das preferências diretas ou indiretas do utilizador. Para além disso, as mensagens OpenADR permitem aos utilizadores individualmente responderem aos pedidos de *Demand Response* com um sinal de "opt out", o que aumenta ainda mais a flexibilidade dada às opções dos utilizadores. O resultado é um sistema que promove a *DR* automatizada, enquanto maximiza a flexibilidade local em resposta aos pedidos.

O núcleo da norma OpenADR 2.0 é constituído por um conjunto de modelos de dados e padrões de troca que definem os sinais de DR e as interfaces entre: mercados de energia (no sentido de suportarem uma informação de preço dinâmico), *Independent System Operators* (ISO), *Distributed Energy Resources* (DER) e consumidores de energia (industriais/residenciais).

Se por um lado os protocolos que permitem a comunicação entre os diversos equipamentos de uma *Micro Grid* estão a ser definidos, há ainda que definir os procedimentos de controlo de cargas otimizados que permitam o ajuste DR. Esse é o âmbito das próximas secções.

III. Algoritmos de Otimização

Em matemática, o termo otimização, refere-se ao estudo de problemas em que se busca minimizar ou maximizar uma função através da escolha sistemática dos valores de variáveis reais ou inteiras dentro de um conjunto viável.

Para a resolução destes problemas, além de possíveis métodos específicos, existem métodos genéricos de

otimização, usualmente designados de meta-heurísticas, que podem aproximar as soluções dos problema que se consigam modelar de acordo com estruturas de dados adequadas. Exemplos são os Algoritmos Genéticos (AG) [10], os algoritmos baseados em colónias de formigas (*Ant colony Optimization*) [11] ou os algoritmos *Particle Swarm Optimization* [12].

Em particular neste artigo dedicaremos a nossa atenção aos AGs. O processo evolutivo dos AG começa por uma população (ou geração) inicial, formada por um conjunto de indivíduos que representam soluções do problema a otimizar. Em geral, a população inicial é definida aleatoriamente, usando heurísticas, ou soluções conhecidas. As gerações seguintes são obtidas iterando sobre os passos que se seguem até se verificar o critério de paragem especificado. Começa-se por avaliar os indivíduos da geração de forma a definir a sua "qualidade" dentro da população. No passo seguinte alguns dos indivíduos são selecionados de acordo com as suas aptidões, i.e., quanto melhores forem os indivíduos mais chances terão de serem escolhidos para o efetuarem o cruzamento. Na fase do cruzamento são criados novos indivíduos, os filhos, que combinam as caraterísticas dos pais. A alguns desses novos indivíduos são aplicadas mutações de modo a evitar a estagnação da população e consequente convergência prematura. Os indivíduos obtidos são agora colocados numa nova população com tamanho igual ao da população original. O processo pode ser resumido de acordo com o Algoritmo 1.

Algoritmo 1 *Algoritmo genético*

1: Gerar a população inicial.
2: **Enquanto** critério de paragem não for satisfeito **faça**
3: Avaliar cada indivíduo da população.
4: Seleccionar os indivíduos mais aptos.
5: Criar novos indivíduos aplicando os operadores: cruzamento e mutação.
6: Armazenar os novos indivíduos numa nova população.
7: **Fim Enquanto**
8: Devolve a melhor solução encontrada

Na seção seguinte iremos apresentar uma formulação para problema da gestão eficiente de energia e discutir a solução implementada.

IV. Descrição e formulação do Problema

A. Formulação do Problema

O objetivo de um sistema de controlo de cargas é o de minimizar os custos do consumo de eletricidade de diversos aparelhos deslocando-os no tempo (ou ajustando a potência consumida). O sistema deverá procurar minimizar o custo em função do tarifário em vigor. Para além disso, o sistema deve impedir em cada momento que o consumo exceda a potência contratada.

Deste modo são vários os objetivos, dos quais podemos destacar: minimizar o custo dos consumos de diversos equipamentos; não exceder a potência contratada; evitar exceder em consumo a potência produzida por fontes renováveis, quando existentes; utilizar tanto quanto possível as tarifas económicas; ter em conta as restrições temporais impostas pelo utilizador para cada um dos equipamentos.

B. Descrição do Simulador

O sistema que se pretende implementar pode ser visto como um sistema em que cada objeto (contadores, geradores e consumidores) é um agente dentro do Sistema Multi-Agente (SMA)[13]. Nesse sentido, o simulador foi implementado como um sistema semidistribuído, em que o processo de otimização é controlado por um agente central que comunica e decide os períodos de funcionamento dos outros agentes usando um algoritmo genético. Como benefício, o sistema distribuído pode fornecer redundância, flexibilidade e adaptabilidade. No nosso caso, trata-se de um sistema bastante flexível quando consideramos a adição ou remoção de novos objetos produtores ou consumidores, que é feita de forma transparente. Salientamos que foi ponderado um processo de otimização distribuído, usando processos de negociação de períodos de funcionamento entre agentes, mas que nesta fase foi deixado para trabalho futuro. Utilizando o sistema híbrido que esboçamos, consegue-se tirar partido das características dos sistemas centralizados que permitem exercer um controlo mais rígido sobre os objetos, pois exige que todas as comunicações e decisões passem por um único objeto centralizador, não perdendo a flexibilidade associada à volatilidade dos restantes agentes.

Como referido, neste trabalho foi implementado um sistema semidistribuído em que todos os agentes consumidores respondem ao sincronismo de um agente central, o gestor de cargas. Este agente coordena a actividade dos restantes e contabiliza os custos. Cada um dos outros agentes está encarregue de guardar o seu histórico de utilização e informar o centralizador das suas potências.

Relativamente ao simulador este foi implementado sobre a plataforma SPADE (*Smart Python Multi-Agent Development Environment*) [14], [15]. O SPADE foi constuído à volta da framework de comunicação XMPP/Jabber. Desenvolvido em Python, trata-se de um sistema particularmente útil na implementação de SMA, suportando os conceitos de agente e servidores, a comunicação entre agentes desenvolvidos em múltiplas linguagens de programação, processamento de comportamentos e o protocolo de comunicação extensível baseado em XML. O SPADE segue ainda as especificações FIPA para SMA [16].

Quanta à implementação do simulador foi considerada a sequência de passos que a seguir descrevemos. O processo começa com o gestor de cargas (GC) a enviar uma mensagem em *broadcast* para todos os equipamentos de consumo, com a requisição dos pedidos de funcionamento dos equipamentos, horário de início pretendido, limite para terminar o programa, além do seu registo de consumo previstos ao longo do tempo. Para efeitos de simulação, foi estipulado que esta requisição é enviada às 7h00. O GC aguarda pelas respostas dos agentes e após as ter recebido com os dados requisitados, efetua o escalonamento dos horários de início de funcionamento das máquinas. Como já referimos, neste escalonamento é aplicado um AG. A informação do escalonamento (horários de ínicio) é depois enviada a cada um dos consumidores.

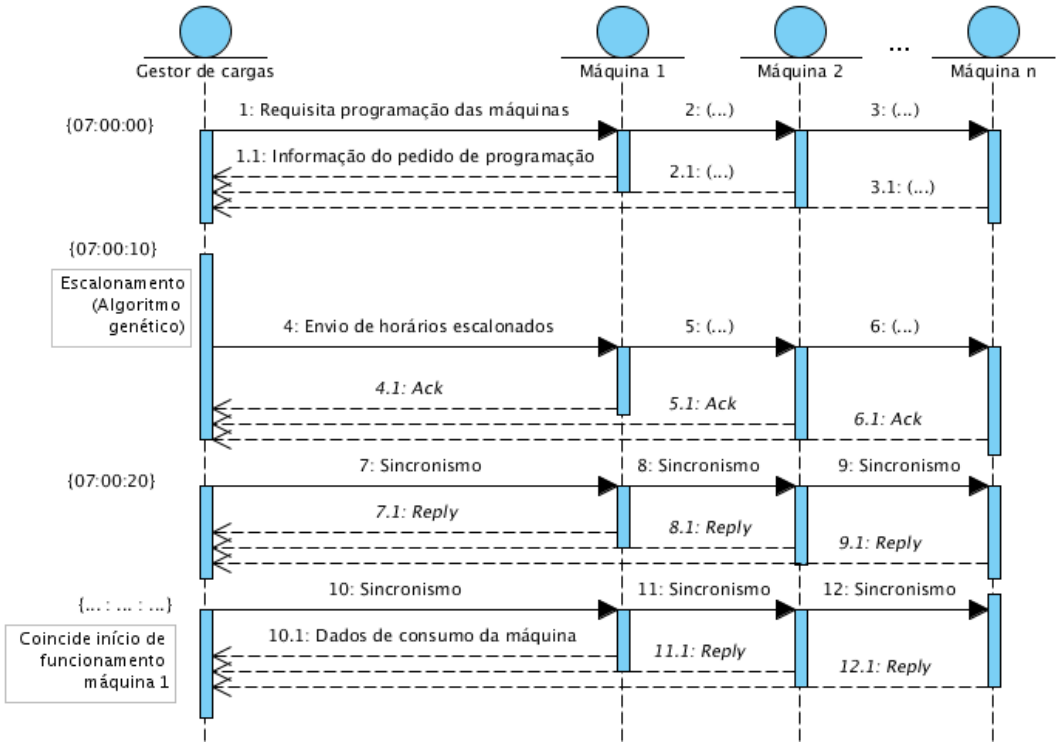


Figura 1. Diagrama da sequência de mensagens entre o gestor de cargas e os consumidores.

Cada equipamento consumidor guarda o horário de início de funcionamento e envia mensagem de *Acknowledgement* para o gestor de cargas. A partir daí o gestor de cargas passa a enviar apenas mensagem de sincronismo. Quando o equipamento consumidor atinge o horário de início de funcionamento passa a enviar os seus dados de consumo até terminar o período de operação.

Relativamente ao AG foi usado a *framework* PyEvolve [17] com os indivíduos codificados como listas de inteiros correspondentes ao início do programa da máquina consumidora. A configuração usada inclui o método de seleção uniforme, o operador de cruzamento de ponto único e os operadores de mutação *swap* e Gaussiano inteiro.

Na Figura 1 está resumida sequência que acabámos de descrever.

V. Testes e Ensaios

Por forma a confirmar a fiabilidade do simulador desenvolvido fez-se uma análise gradual de testes com o objetivo de verificar o procedimento de cálculo do custo e o correcto posicionamento das cargas.

Admitindo uma potência contratada de 3.3 kVA, foi considerado um valor unitário para o factor de potência (PF), resultando na potência ativa de 3.3 kW.

Admitindo que numa habitação ou empresa existirão cargas que não são controláveis, o sistema de controlo de carga irá ajustar a soma das potências das cargas controláveis para que esta não ultrapasse uma percentagem da potência contratada (por exemplo 0.80), evitando assim o deslatre dos circuitos quando as cargas não controláveis aumentem de potência.

Para efeitos de testes foi considerado um tarifário com três tarifas:

- P1 (“horas de vazio”) – 0.1 euros por kWh entre as 22h00 e as 22h30;
- P2 (“horas de cheia”) – 0.1486 euros por kWh entre as 22h30 e as 10h30 (do dia seguinte); e
- P3 (“horas de ponta”) – 0.1865 euros por kWh entre as 10h30 e as 22h00.

Tendo em conta este tarifário e admitindo cargas de potência constante durante períodos de 1 hora, a expectativa é que o início de funcionamento das cargas fique próximo do início do período das “horas vazias”. Assim, por exemplo, admitindo 3 cargas de 1kW com uma duração de 1 hora, os tarifários anteriores e uma potência máxima controlável de 2.64 kW (admitindo uma rácio entre potência controlada e potência contratada de 0.8), espera-se que duas cargas tenham início aproximado às 22h00 e que uma das cargas seja obrigada a deslocar-se para o tarifário P2.

Consideremos outro exemplo, admitindo quatro cargas em que uma das cargas tem 2.5 kW constantes durante uma hora e as outras três cargas consomem 1 kW durante uma hora, a carga de 2.5 kW deverá ocupar o horário da tarifa mais baixa e as restantes três cargas ficarão na tarifa intermédia, pois é essa a solução que minimiza o custo.

De seguida apresentamos resultados experimentais considerando um cenário de cargas constantes ao longo do seu

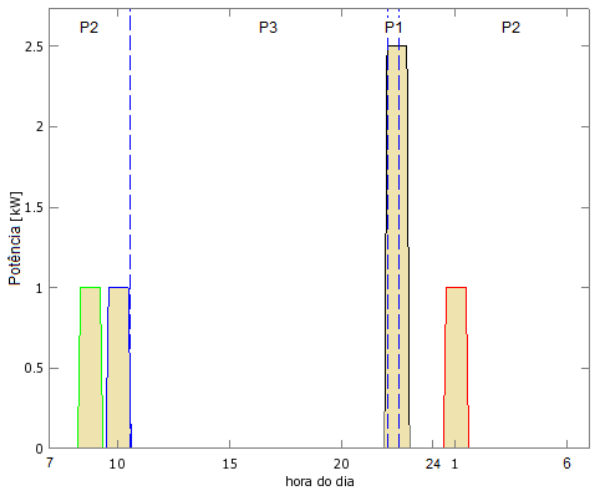


Figura 2. Consumo das cargas de acordo com o escalonamento e tarifários para 4 equipamentos

período de funcionamento e posteriormente os ensaios considerarão curvas de carga de equipamentos reais.

A. Resultados para uma máquina 2.5 kW e três máquinas de 1 kW com três tarifários

Nestes testes foram quantificados os custos resultantes do escalonamento temporal dos diversos equipamentos, comparando a introdução do algoritmo genérico com uma seleção aleatória da hora de entrada em funcionamento das máquinas. Consideraram-se quatro equipamentos: um consumindo uma potência de 2.5 kW durante 1 hora e três consumindo uma potência de 1 kW durante 1 hora. Para limitar as opções dos testes, a tarifa mais baixa apenas esteve disponível entre as 22:00 e as 22:30 (de acordo com o tarifário apresentado na secção anterior). Os equipamentos foram definidos para funcionar numa janela temporal de 24h, entre as 7h do primeiro dia e as 7h do dia seguinte.

As simulações foram repetidas vinte vezes, tendo-se posteriormente calculado a média e desvio padrão de custos para ambas as situações (escalonamento aleatório e escalonamento com o AG). A média de custos diária para a entrada em funcionamento aleatória dos equipamentos foi de 0.9177 Euros, com um desvio padrão de 0.0691 Euros. Já no caso da introdução do AG alcançou-se um custo médio de 0.7561 Euros com desvio padrão igual a 0.0026. Estes resultados traduzem uma redução média de custos de 17.6%.

A Figura 2 apresenta um dos resultados do algoritmo genético. Como se pode constatar, a carga que consome maior potência ficou escalonada para iniciar o funcionamento às 22:00 e as outras cargas ficaram escalonadas para funcionar no período do tarifário com custo intermédio (P2), o que traduz o custo mais baixo possível.

B. Resultados para quatro equipamentos (reais) com três tarifas

Nos testes seguintes foram consideradas curvas de potência consumida de diversos equipamentos existentes em residências, obtidas utilizando o Analisador de Qualidade de Energia Fluke 435. Entre os registos disponíveis, não foram considerados controláveis cargas como arcas frigoríficas, iluminação ou aparelhos AVAC. Os quatro equipamentos considerados controláveis foram: a máquina de lavar a roupa, a máquina de secar roupa, o termoacumulador e bomba de piscina. Os resultados destes ensaios não têm uma análise tão balizada, em que se perceba de antemão qual é o resultado final do posicionamento de cada uma das cargas de consumo porque estas poderão ser "encaixadas" nas tarifas mais baixas sem ultrapassar os 80% da potência contratada. O algoritmo deverá garantir que o início de funcionamento das máquinas não conduza a que a soma das potências ultrapasse os 80% da potência contratada. Na Figura 3 apresenta-se a potência consumida ao longo do tempo por: a) Bomba de Piscina; b) Máquina de Lavar Roupa; c) Máquina de Secar Roupa; d) Termoacumulador.

Também neste caso o simulador foi executado vinte vezes. Das simulações obteve-se uma média de 0,5028 Euros (desvio padrão igual 0.0309 Euros) para a entrada em funcionamento dos equipamentos num instante aleatório. Já no caso da introdução do AG alcançou-se um custo médio de 0.4232 Euros com desvio padrão igual a 0.003. Estes resultados traduzem uma redução média de custos de 16%, confirmando os resultados obtidos previamente.

VI. Conclusões e Trabalho futuro

Neste artigo foi apresentado um método para efetuar o escalonamento de cargas ao longo de um dia com o objetivo de minimizar os custos associados aos consumos. Essa minimização tem em conta os tarifários conhecidos à priori. O processo foi simulado na plataforma multi-agentes SPADE, tendo sido apresentado os valores obtidos para dois cenários. Conclui-se que no cenário de teste o AG obteve uma melhoria de cerca de 17%, enquanto que no cenário com leituras de máquinas reais essa melhoria foi de cerca de 16%.

Como trabalho futuro, pretende-se considerar a produção de energia através de fontes renováveis, otimizando de acordo com as estimativas de produção ao longo do dia, aplicar cargas não controláveis, estudar processos de otimização distribuídos incluindo processos de negociação entre agentes.

Agradecimentos

Este trabalho foi parcialmente suportado pelo projeto Manage the Intelligence QREN I&DT, n.o 30260.

Referências

[1] "Propuesta de real decreto por el que se establece la regulación de las condiciones administrativas, técnicas y economicas de las modalidades de suministro de energía eléctrica con autoconsumo y de producción com autoconsumo," Online: http://www.suelosolar.es/newsolares/newsol.asp?id=8479, Julho 2013.

[2] M. A. Piette, G. Ghatikar, S. Kiliccote, E. Koch, D. Hennage, P. Palensky, and C. McParland, *Open automated demand response communications specification (Version 1.0)*, California Energy Commission, PIER. CEC-500-2009-063, 2009.

[3] P. L. Joskow and C. D. Wolfram, "Dynamic pricing of electricity," *The American Economic Review*, vol. 102, no. 3, pp. 381–385, 2012.

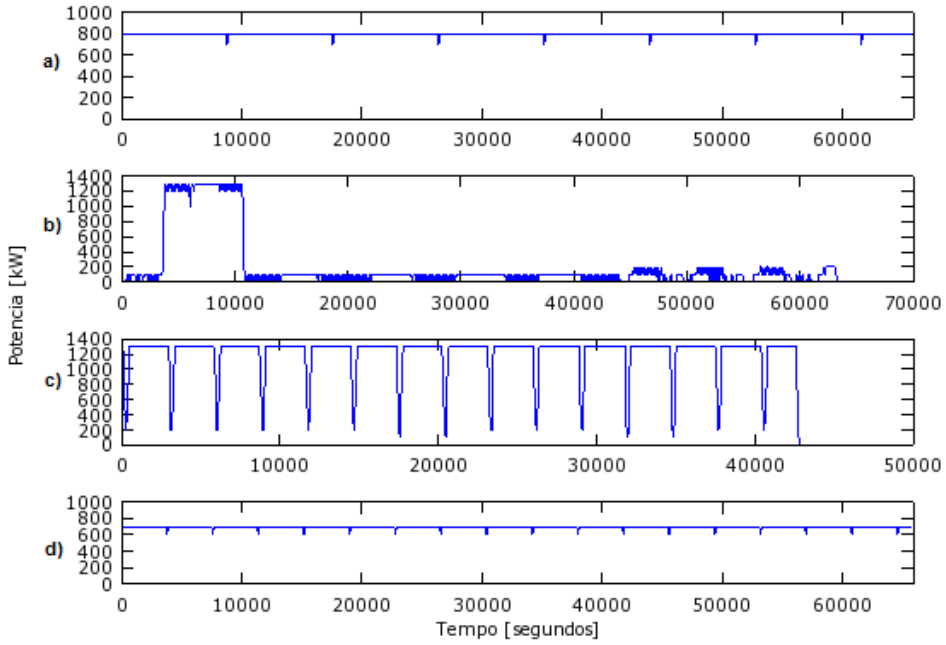


Figura 3. Potência consumida ao longo do tempo pelos equipamentos: a) Bomba de Piscina; b) Máquina de Lavar Roupa; c) Máquina de Secar Roupa; d) Termoacumulador

[4] *Utility-Scale Smart Meter Deployments, Plans & Proposals*, Institute for Electric Efficiency, Setembro 2011.

[5] J.-P. Vasseur and A. Dunkels, *Interconnecting smart objects with ip: The next internet*. Morgan Kaufmann, 2010.

[6] *Smart Energy Profile 2, Application Protocol Standard*, Zigbee public document 13-0200-00 ed., ZigBee Alliance, abril 2013.

[7] *IEEE Standard for Ubiquitous Green Community Control Network Protocol*, IEEE Std 1888 Std., abril 2011. [Online]. Available: http://standards.ieee.org/findstds/standard/1888-2011.html

[8] *OpenADR 2.0. Profile Specification. A Profile*, OpenADR Alliance, 2012.

[9] Z. Shelby, K. Hartke, and C. Bormann, *Constrained Application Protocol (CoAP)*, Draft: draft-ietf-core-coap-18, Internet Engineering Task Force (IETF) Std., junho 2013.

[10] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*. Springer Berlin, 2010, vol. 2.

[11] M. Dorigo and T. Stützle, *Ant Colony Optimization*. MIT Press, 2004.

[12] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[13] Y. Shoham and K. Leyton-Brown, *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2009.

[14] E. Argente, J. Palanca, G. Aranda, V. Julian, V. Botti, A. Garcia-Fornes, and A. Espinosa, "Supporting agent organizations," in *Multi-Agent Systems and Applications V*. Springer, 2007, pp. 236–245.

[15] M. E. Gregori, J. P. Cámara, and G. A. Bada, "A jabber-based multi-agent system platform," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. ACM, 2006, pp. 1282–1284.

[16] "Fipa specifications," online: http://www.fipa.org/, 2013.

[17] "Pyevolve," online: http://pyevolve.sourceforge.net/, 2013.

A.2 *"A Distributed Load Scheduling Mechanism for Micro Grids", IEEE SmartGridComm 2014, Venice (Italy)*

A Distributed Load Scheduling Mechanism for Micro Grids

Jânio Monteiro[1,2], Jorge Eduardo[1], Pedro J. S. Cardoso[1,3], Jorge Semião[1,2]
[1] ISE, University of Algarve, Faro, Portugal
[2] INESC-ID, Lisbon, Portugal
[3] LARSys, University of Algarve, Faro, Portugal
Emails: {jmmontei, jmeduardo, pcardoso, jsemiao}@ualg.pt

***Abstract*— Several protocols have recently been defined for smart grids that enable the communication between electric devices and energy management systems. While these protocols and architectures can already be applied in different fields of micro grids, it is still not clear how the distributed resources and constraints of such electrical grids can be managed in an optimum way. In order to achieve a reduction in electricity costs and maximizing investments made in renewable sources, an optimization mechanism should be used to perform load scheduling, considering different variables such as forecasted power generation curve from renewable sources, different tariffs' rates, electric circuit constraints, user restrictions and correspondent comfort levels. Given these considerations, this work defines and evaluates a distributed micro grid resource management architecture and protocol which is able to optimize load scheduling while considering all the mentioned restrictions and parameters. The proposed architecture was implemented on a multi-agent simulator and the performed tests show that significant reductions in electricity cost can be achieved using this methodology.**

I. INTRODUCTION

When comparing current electrical grids with the ones that we had a few years ago, a very different dynamism is verified which results from the increasing introduction of renewable energy sources. Those renewable power sources are sometimes characterized as Intermittent Resources (IRs), as they depend on environmental factors that make them significantly vary over time, and difficult to predict with accuracy. This may in turn cause inefficiencies and mismatches of various kinds in the necessary equilibrium between production and consumption.

In order to reduce these mismatches several solutions can be considered. Some proposals opt for promoting an adjustment in the consumption side using dynamic tariff rates (so called Demand Side Management) using dynamic tariff rates, so that the consumption may adapt to the power being produced. In this field, Distribution System Operators (DSOs) typically buy electricity in markets that already define their prices daily, reflecting the forecasted supply and demand data for the following day (as for instance happens in [1]). These dynamic tariffs are also being applied to DSO customers in various regions of Europe and United States [2], because constant tariff rates do not correlate with the marginal costs of production [4]. Based on these tariffs, either automatically or by human intervention, the working periods of equipment can be changed to take advantage of the lowest price and high production.

In this sense, the goal of creating a system capable of energy management is to implement a set of so-called smart objects [4], supported in the concept of the Internet of Things (IoT), that by communicating with each other and acting based on an optimized control system, allow a better use of the energy produced by renewable energy sources and the improvement of energy management in buildings.

In terms of energy control, several protocols like the Smart Energy Profile - Version 2 (SEP 2.0) [5], IEEE 1888 [6], and the OpenADR 2.0 [7] protocol architectures have already been defined. However, while these protocols and architectures can already be applied to Micro Grids, a mechanism is necessary to enable the management and control of the distributed resources that are typically available in such grids.

One of such resources is electrical power. In fact, while until now load scheduling has been performed non-automatically, the introduction of automatic management systems in medium to large scale installations can cause demand hikes at low price periods, causing a disruption of supply, due to overloading. Thus, a Micro Grid energy management system should take into consideration electrical circuit constraints [8], while reducing electricity costs and maximizing investments made in renewable sources equipment. That mechanism should implement load scheduling, resulting from optimization algorithms that reflect user comfort levels and restrictions [9]. It should also consider the forecasted renewable power generation and the different rate tariffs from the DSOs.

Given these considerations, this paper introduces a new Micro Grid energy management system which, considering a tree based electrical grid [8], defines a communication and control structure composed by agents. To test our proposal, a simulator based on a Multi Agent System [10] was implemented and the experimental tests show that electricity cost reductions can be achieved once the management system is used.

The remainder of the paper has the following structure. Section II analyses a Micro Grid structure and set of protocols developed for communication and control in such electrical grids. Section III, proposes a Resource Management Protocol

for the distributed management of Micro Grids. Section IV described the simulation platform and results obtained using the proposed protocol. Finally, section V concludes the paper.

II. Micro Grid Structure and Protocols

A. Communication Protocols in Smart Grids

The user's ability to manage their energy consumption according to the production is a critical feature of Smart Grids, and a base for innovation, new products and services. In order to support this capability, the communication between different devices such as meters, appliances, electric vehicles, energy management systems and distributed energy resources (including renewable energy and storage) must occur using secure, standard and open procedures. In this context, several protocols have been recently defined.

One of these protocols, the Smart Energy Profile [5] results from the collaboration between the low-power ZigBee, Wi-Fi and HomePlug power-line technologies, building a power management architecture for Micro Grids, supported on IP networks.

In March 2011, the Institute of Electrical and Electronics Engineers (IEEE) announced the approval and publication of the Standard for Ubiquitous Green Community Control Network Protocol (IEEE 1888 TM) [6] within the Ubiquitous Green Community Control Network Protocol (UGCCNet). Originating in China, the IEEE 1888 standard defines itself as a global standard within the IoT, which aims at energy efficiency through the management of renewable energy, through communication using Internet protocols and Information and Communication technologies.

Another communication protocol for Smart Grids is the Open Automated Demand Response (OpenADR) version 2.0 [7]. The OpenADR is an evolution and extension of the first version, developed by the Demand Response Research Center at Lawrence Berkeley National Laboratory. It is supported by the OpenADR industrial alliance, having been developed as part of the standard OASIS Energy Interoperation 1.0, published in February 2012 [11].

If on one hand the protocols that allow communication between different devices of a Micro Grid are being developed, a control procedure is still needed to support an optimized load scheduling when managing distributed resources. This is the purpose of the forthcoming sections.

B. Micro-Grid Architecture

Fig. 1 presents a typical structure of a low voltage Micro Grid [8], common among industrial and business facilities. These structures are comprised of a hierarchy of Distribution Boards (DB), where the Main General Distribution Board (MGDB) interconnects the external DSO circuits to several internal workshop circuits (represented as A, B/B_x, C/C_x and D, in Fig. 1). Workshop DBs can be divided into intermediate DBs if they obligatorily feed other lower level DBs, and possibly, electrical loads (e.g., A, B, C and D in Fig 1) or leaf DB if they only feed loads (e.g., B_1, B_2, B_3, B_4, C_1 and C_2 in Fig 1).

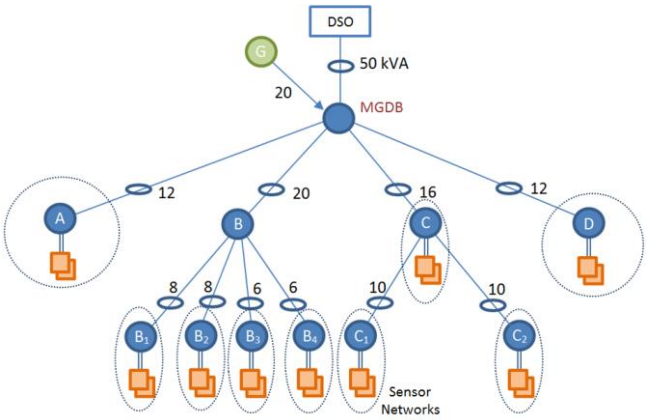


Fig. 1. Example of micro-grid architecture with a tree structure comprising several Distribution Boards and having a renewable generator, in node G.

At the lower levels we find electrical loads (represented by orange boxes in Fig. 1). They can be controlled in terms of one or more of the following parameters: when to start, when they should finish, or the maximum power to be drawn from the electrical grid. Some of them can also be controlled through indirect parameters, like for instance HVAC (heating, ventilation, and air conditioning) set point temperatures. Finally, some loads are not controllable and/or individually monitored.

Each intermediate and leaf DB can connect tens of circuits, aggregating hundreds of loads. Furthermore, simultaneity (or diversity) factors, *ks*, are applied at each DB level, considering that not all equipment runs at the same time. Usually, the simultaneity factor values range from 0.1 to 1.0, depending on the type of loads that are connected to a certain circuit. They enable the computation of the expected resulting aggregated load, which is drawn from higher levels boards. This procedure is repeated in higher DBs, leading to an expected total demand for the full installation (exemplified as 50 kVA in Fig. 1). The aggregated power of these installations can easily reach cents of kVA in industrial installations, distributed over tens of DBs.

Simultaneity factors result from practice and considering that working periods of equipment are typically spread over time. However, they were not computed considering that many devices could work at the same time, as it may happen if a period of lower tariffs is combined with a greedy automatic load shifting. Thus if scheduling is applied to loads, some measures as the one proposed in the next sections should be taken to avoid overloads.

C. Communication Architecture

Given the architecture presented in Fig. 1, the communication structure that controls and monitors electric devices should derive from the electrical structure. Thus, in such control system, we consider that a Monitoring and Control Device (MCD) should be placed at each DB. The set of MCDs will form a distributed Energy Management System (EMS) of the whole installation.

At each distribution board, MCDs measure the current, voltage, active and reactive power consumed from the upward circuit, while communicating through wireless and/or wired Sensor Networks (SN) with electrical equipment. Sensors devices are also used to measure ambient data (e.g., temperature, movement, and light intensity).

MCD devices are thus in charge of Machine-to-Machine communication while reflecting Human-to-Machine interactions. Based on these inputs they define when terminal devices should work. These load scheduling decisions should result from optimization algorithms that take into consideration: (1) the forecasted power curves of installed renewable sources in the yet-to-come minutes/hours; (2) the power consumption curve of each equipment/load; (3) the future minute/hourly based tariffs charged by the DSO; (4) the local and global power constraints imposed by the electrical installation; and (5) human requirements and comfort levels.

Given the computation capabilities available in many electronic devices, MCDs are currently capable of running optimization algorithms and communicating with each other for the management of distributed resources, which are shared by the whole micro grid. While this distributed architecture is capable of parallel computing, it also places several challenges in terms of coordination between control devices and scalability.

In order to address these issues, in the following we consider that optimization algorithms for load scheduling run in a distributed fashion at MCDs, making local decisions that reflect a global equilibrium of the system. Given these considerations, we will define and evaluate a communication mechanism that can be used to manage these electrical devices.

III. A Resource Management Protocol for Micro Grid Management

A. *Introducing Distributed Resource Reservation*

The problem of distributed resource reservation has been addressed previously in computer networks. The Resource Reservation Protocol (RSVP) in particular, specified in IETF RFC2205 [12] and updated since then with several features, was used to support a distributed Quality of Service (QoS) resource reservation procedure among several Integrated Services routers [13].

RSVP considers two fundamental message types: PATH and RESV. In IP Multicast trees, the PATH message travels downstream along the multicast routes with information about the traffic that the sender application expects to generate and storing path state with the QoS control capabilities of routers along the path. RESV messages are originated in leaf nodes and travel upstream, being used to request an appropriate resource reservation from the desired QoS. As RESV messages move from receivers to senders, reservation parameters are merged at intermediate nodes.

While the RSVP protocol cannot be applied directly to the resource reservation problem described in sections II.B and II.C, a similar concept may be used to implement a distributed mechanism for load management.

B. *Micro Grid Resource Management Protocol*

Differently from the RSVP protocol, that only reserves flows for a subsequent time period, in the optimization mechanism that we are considering, such requests should also address future time intervals. This means that resource request messages must carry a vector of *n* power requests, where each index refers to a time interval (for instance for the 5 minutes interval between 10:15 and 10:20). In this case, index 0 refers to present time and subsequent indexes refer to future time intervals.

The proposed Micro Grid Resource Management Protocol considers two communication phases (as shown in Fig. 2), which are similar with the ones that were defined for RSVP. For each of these phases one message type is used: a Resource Information (RI) message, and a Resource Allocation (RA) message.

In the first stage, the MCD at the top of the tree multicasts RI messages. Each of these messages contains three vectors, represented by (*R*, *P*, *C*): the *R* vector informs lower MCDs about the forecasted power that is expected to be generated by renewable sources; *P* vector translates the ratio of maximum upward power that lower MCDs can allocate; and *C* contains the energy cost (per kWh) associated with each time interval. Each of the time intervals of the *C* vector starts by reflecting the tariff of the DSO. However, as explained later, the associated values will be adjusted to avoid cyclic overloading in adjacent time periods, penalizing the intervals where these overloads occur.

As these RI messages traverse down the tree (i.e., from the top to leaf MCDs), *P* and *C* vectors may be changed by intermediate MCDs, in order to reflect their own capabilities and state. Thus, when these RI messages reach a leaf MCD, the (*R*, *P*, *C*) vectors reflect the capability of the whole grid, being used as input in the optimization algorithm to decide: when loads should start working, when they should finish and/or what is the power level they are allowed to request [14].

Leaf MCDs, after running the optimization algorithm, generate an aggregate load vector, which is sent upstream using a Resource Allocation (RA) message. Intermediate MCDs, after receiving RA messages from lower MCDs, behave like leaf MCD, i.e. they run optimization algorithms to decide when loads should start working, when they should finish and/or what is the power level they are allowed to request. However, while they may be allowed to perform time shifting of their own loads (depending on the user's restrictions), they are typically not allowed to shift aggregated loads that they receive from lower level MCDs.

If at some time instant(s), the aggregated load surpasses the maximum allowed upward power of a DB, the MCD must act, since it is not possible to assure the requested power. This may happen if several loads of different downward aggregators are scheduled to work at the same time. At this point, intermediate MCD aggregators should increase the cost of the energy associated with the overload periods and explicitly instruct lower level MCDs to reduce the power they are requesting for the time intervals where overloads happened. In both cases upper level MCDs will inform lower level MCDs about the

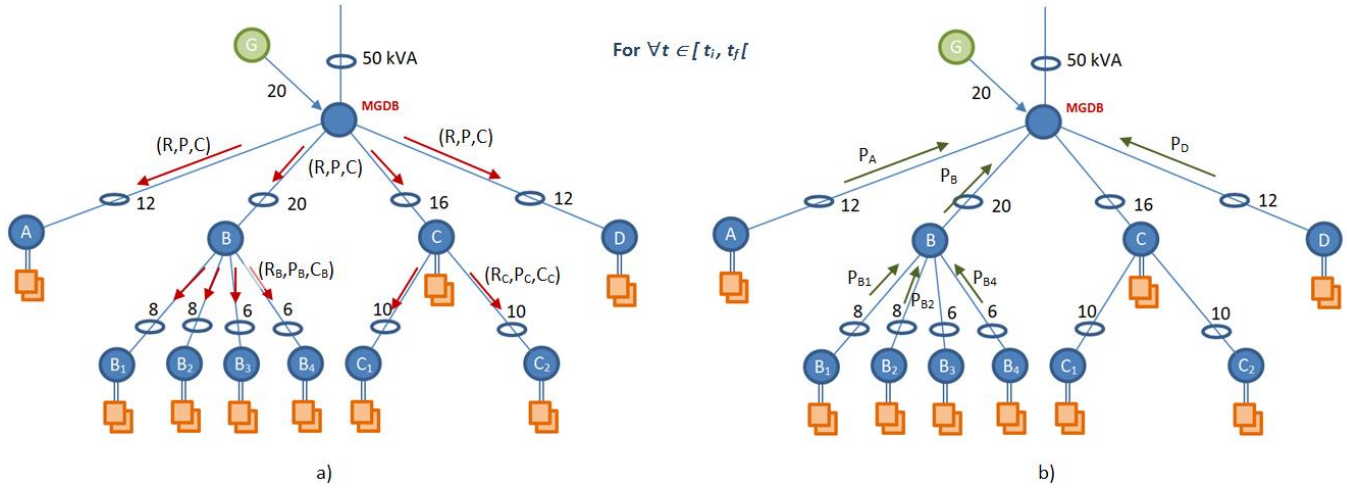


Fig. 2. Two stages of the Micro Grid Resource Management Protocol: a) Resource Information (RI) messages travel down the tree carrying information about the resources that are commonly distributed, b) Resource Allocation (RA) messages inform upper nodes about the forecasted power consumption of each aggregator node.

required reschedule of their loads using a subsequent RI message, changing the associated power and cost vectors of (*R*, *P*, *C*), which will lead lower MCDs to make the necessary adjustments.

Each time a new load scheduling is requested, an RI message is sent upwards, which triggers the exchange of RA and RI messages. This process stops when the top level MCD verifies that after several repetitions the cost does not improve. It then stops sending RA messages.

Given this brief explanation, in the following we will describe this resource management mechanism in more detail.

C. Optimization Mechanism at MCDs

The task of leaf MCDs is to run the optimization algorithms that minimize the cost of electric consumption of various loads, shifting them in time or adjusting the power consumed, taking into consideration: (1) the electricity tariffs, (2) the power generated from local renewable sources, (3) the time constraints imposed by the user for each device and (4) the micro grid electric structure and constraints.

In order to do this, after receiving an RI message with (*R*, *P*, *C*) vectors, leaf MCDs run a Genetic Algorithm (GA) targeting the minimization of the objective function given by:

$$F = \frac{1}{Q}.\sum_{t \in T} \alpha(t).C(t) + \beta(t) \ , \qquad (1)$$

where *Q* translates the quality assessment of the scheduling solution seen from a user perspective, *T* represents a set of time intervals, *C(t)* translates the cost of energy for interval *t* (obtained from the RI message),

$$\alpha(t) = \begin{cases} (P_{MCD}(t) - R'(t)).\Delta_t & if \ P_{MCD}(t) > R'(t) \\ 0 & otherwise \end{cases} \qquad (2)$$

and

$$\beta(t) = \begin{cases} \infty & if \ P_{MCD}(t) > P_{\max} \\ 0 & otherwise \end{cases} . \qquad (3)$$

In equations (2) and (3), $P_{MCD}(t)$ translates the sum of load's power scheduled to work at the time interval *t*; P_{max} translates the maximum upward power limit of the DB of the MCD; Δ_t translates the time period associated with each *R*, *P* or *C* vector entrance; and *R'* is obtained from the *R* vector received in the RI message using equation:

$$R'(t) = \begin{cases} R(t) & if \ \overline{P}(t) = 0 \ \& \ R(t) \geq 0 \\ \overline{P}(t) + R(t) & if \ \overline{P}(t) > 0 \ \& \ R(t) \geq 0 \\ \overline{P}(t).(1 + R(t)) & if \ R(t) < 0 \end{cases} , \qquad (4)$$

where $\overline{P}(t)$ represents the last requested $P_{MCD}(t)$ vector, in the iterative optimization process. In this sense, *R*' translates an estimation of the power generated by renewable sources at time interval *t*.

Using these equations, the Genetic Algorithm procedure defines when loads should be scheduled to start. These decisions can be conveyed to loads using one of the protocols described in section II.A. However, this can only be made after a micro grid level verification of the solution. In order to obtain it, leaf MCDs send an RA message to an upper level MCD containing the aggregated load vector, $P_{MCD}(t)$.

D. Load Aggregation

After receiving RA messages from lower level MCDs, an upper layer MCD sums up the lower level $P_{MCD}(t)$ load vectors, generating an aggregated vector of requested power P_r.

If $P_r(t)$ is higher than the maximum power of the upward circuit (i.e., P_{max}) at some time interval(s) *t*, then the values of

P and *C* vectors stored in the MCD will be updated, for all the time intervals *t* where overloads happened, according to equations:

$$P(t) = \frac{P_{\max}}{P_r(t)} \quad (5)$$

and

$$C(t) = C_0(t) + \left(\frac{1}{1+e^{(k-n)}} \right).\Delta T \quad (6)$$

where $C(t)$ represents the new value of the energy cost at time interval t , $C_0(t)$ is the vector with cost values obtained from the DSO, ΔT represents a difference between tariffs, k is a constant used to adjust the responsiveness to repeated overloads and n represents the number of overloads that happened for time t. Equation (6) adds memory to the cost vector with the aim of reducing fibrillation, which happens when several loads continuously and in parallel oscillate around a small set of time intervals. ΔT and k can assume different values according to the level of the MCD.

After changing the power and cost vectors and before running its own optimization algorithms, $P_r(t)$ is upper limited to P_{max}, for all time instants where overloads occurred. Using the resulting power margin, the genetic algorithm is used to decide where loads should work, setting the aggregated power vector P_{MCD}, of the intermediate MCD, which is sent to the upper layer MCD, through a subsequent RA message.

E. The (R, P, C) computation

When RA messages arrive to the top level MCD, it will act like an intermediate node, with the exception that it will not generate a new RA message. Instead, after summing up the P_{MCD} load vectors received from lower MCDs and obtaining an aggregated requested power vector P_r, it will change the (R, P, C) vectors to reflect the capability of the whole grid, before sending it down in a subsequent RI message.

Regarding the *P* and *C* vectors, they will be updated according to the procedure explained in equations (5) and (6), only if and when overloads are expected to happen. For all the time instants *t* where overloads are not predicted to occur (i.e., $P_r(t)<P_{max}$) no information will be conveyed in the *P* vector of the RI message. This means that the *P* vector will not be used to perform a First-Come-First-Serve reservation procedure, which would tend to be unfair with the most recent requests. For those time intervals *t* where overloads occur, *P(t)* will equally force a percentage of reduction in all power requests from lower MCDs (given by equation (5)).

Finally, the *R* vector will be obtained using:

$$R(t) = \begin{cases} P_G(t) - P_r(t) & \text{if } P_G(t) > P_r(t) \\ \dfrac{P_G(t) - P_r(t)}{P_r(t)} & \text{if } P_G(t) \le P_r(t) \end{cases}, \quad (7)$$

where P_G translates the forecasted generation vector of a renewable source. For those values where $R(t)$ is positive, it will convey the forecasted generated power that is still not being used by scheduled loads. However, when $P_r(t)$ surpasses $P_G(t)$, $R(t)$ will be negative and it will carry the ratio of power that all nodes are requesting beyond the forecasted $P_G(t)$. This value will be used by MCDs to estimate the ratio of power that is not being paid, as expressed by equations (2) and (4).

As these RI messages go down the tree, *P* and *C* vectors may be changed by intermediate MCDs, according to their own stored state or capability. In terms of *P*, values that are sent down in a newly generated RI message are the lowest among the ones received in the RI message and the ones stored in the node. As for the *C* cost vector, the MCD will send the highest value among received and stores values.

The next section will outline the simulation platform and some experimental test.

IV. Simulation Platform and Results

A. Simulator's framework

The simulator was specified to implement a system where each object (counters, generators and consumers) is an agent within a Multi-Agent System (MAS) [9]. Taking into consideration the specifications, our simulator was implemented using SPADE (Smart Python Multi-Agent Development Environment) [15][16]. SPADE was built around the XMPP/Jabber communication framework and is developed in Python, showing to be a particularly useful system in the implementation of MAS. Its usefulness comes from its support to: (1) the concepts of agent and servers, (2) the implemented communications between agents, (3) the possibility to develop agents in multiple programming languages (4) the processing of agent behaviors and (5) the extensible communication protocol based on XML. Furthermore, SPADE follows the FIPA specifications for MAS [17].

In this sense, the simulator has been implemented as a distributed system where each of the previous defined objects (e.g., MCDs and electrical consumers/load) are connected in a tree structure like the ones described in the previous sections. Each agent implements the corresponding capacities and behaviors as the communication or the optimization actions.

B. Simulation Tests

Given the architecture defined in Fig. 1, we have implemented a set of simulations for the scheduling of 143 loads, while considering a tariff with 3 price periods (T_0=0.0955 €, T_1=0.1642 € and T_2=0.2066 €) and a power generation curve obtained from a solar photovoltaic plant with a peak production of 25 kW.

At 7:00 a.m., the lower level MCDs gradually start requesting the scheduling of the loads, representing a total demand of 275 kWh. ΔT in equation (2) was set to *5* minutes, while *k=5* and *n=2* in equation (6).

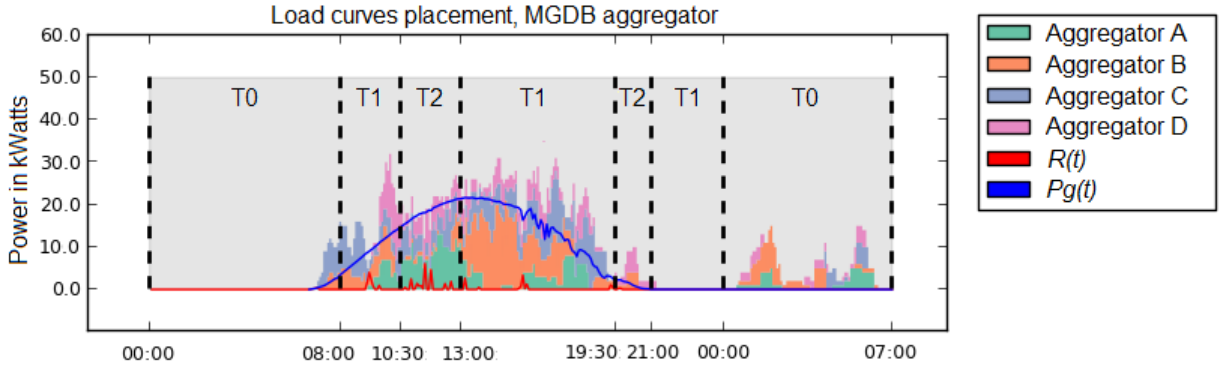


Fig. 3. Load placement resulting from the distributed scheduling algorithm considering 143 load requests after 7 a.m.

Whenever the MCD receives a scheduling request, it will execute a Genetic Algorithm with a population of 128 individuals, 40 generations, an 1D integer list representation (schedule hours are converted into and from an integer value representation), the roulette wheel crossover operator (with 0.9 crossover probability), and a fitness function which takes into consideration the data received through the described communication process (tariff prices and generation), penalizing scheduled overloads. Furthermore, with 0.1 mutation probability, the Genetic Algorithm uses the swap mutator and a special mutator which moves the charges to the lowest tariffs intervals.

Fig. 3 presents one of the scheduling solutions obtained by the algorithm, together with the tariff periods and generation curve. As can be observed, the algorithm is able to schedule most of the loads to the phase where generation was available, while avoiding more costly tariffs.

Using the same parameters, systematic tests with 20 executions were performed comparing the proposed algorithm with a scenario without load scheduling. The associated average and standard deviation results are shown in Table I.

TABLE I. COST RESULTS OF THE PERFORMED SYSTEMATIC TESTS

Evaluation Parameter	**Electricity Cost (€)**	
	Without Load Scheduling	***With Load Scheduling***
Average	17.01	7.98
95% Confidence Interval	17.01 ± 0.67	7.98 ± 0.98

The results of these tests demonstrate that on average the proposed distributed load scheduling mechanism was able to achieve a reduction of 53.1% in the electricity costs, when compared with non-optimized load distribution.

V. CONCLUSIONS

This paper presented and evaluated a new Micro Grid Resource Management Protocol. The proposed system is based on a distributed computational environment where each agent communicates with others to achieve an optimal scheduling for the electrical loads. Among other features, behind the optimization is a Genetic Algorithm which locally optimizes the referred schedule, taking into account the tariff prices, the loads from other objects and the generated power from renewal energy sources. The results show that significant electricity cost reductions can be achieved using this methodology.

Acknowledgments. This work was partly supported by the Portuguese Foundation for Science and Technology (FCT), project LARSyS PEst-OE/EEI/LA0009/2013 and project MTI QREN I&DT, n. 30260. We would also like to thank the project leader Certigarve [www.certigarve.pt].

REFERENCES

[1] OMI-Polo Español S.A. (OMIE), Available: http://www.omie.es/files/flash/ResultadosMercado.swf. Accessed: April, 1st, 2014.

[2] Utility-Scale Smart Meter Deployments, Plans & Proposals, Institute for Electric Efficiency, Sep. 2011.

[3] P. L. Joskow and C. D. Wolfram, "Dynamic pricing of electricity," The American Economic Review, vol. 102, no. 3, pp. 381–385, 2012.

[4] J.-P. Vasseur and A. Dunkels, Interconnecting smart objects with ip: The next internet. Morgan Kaufmann, 2010.

[5] Smart Energy Profile 2, Application Protocol Standard, Zigbee public document 13-0200-00 ed., ZigBee Alliance, Apr. 2013.

[6] IEEE Standard for Ubiquitous Green Community Control Network Protocol, IEEE Std 1888 Std., Apr. 2011.

[7] OpenADR 2.0. Profile Specification. A Profile, OpenADR Alliance, 2012.

[8] "Electrical installation guide 2013, According to IEC international standards", Schneider Electric, 2013.

[9] J. Monteiro, P.J.S. Cardoso, R. Serra, L. Fernandes, "Evaluation of the Human Factor in the Scheduling of Smart Appliances in Smart Grids", 16th Int.. Conf. on Human-Computer Interaction, Jun. 2014, Greece.

[10] Y. Shoham and K. Leyton-Brown, Multiagent systems: Algorithmic, game-theoretic, and logical foundations. Cambridge Univ. Press, 2009.

[11] "OASIS Energy interoperation version 1.0", Feb. 2012.

[12] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP)", IETF RFC2205, Sept. 1997.

[13] J. Wroclawski, "The Use of RSVP with IETF Integrated Services", IETF RFC2210, Sept. 1997.

[14] Eduardo, J., Cardoso, P., Monteiro, J., "Gestão de Cargas numa Micro Grid Utilizando Algoritmos Genéticos", 13ª Conferência sobre Redes de Computadores (CRC'13), pp. 13-18. Portugal, 14-15 Nov. 2013.

[15] E. Argente, J. Palanca, G. Aranda, V. Julian, V. Botti, A. GarciaFornes, and A. Espinosa, "Supporting agent organizations," in MultiAgent Systems and Applications V. Springer, 2007, pp. 236–245.

[16] M. E. Gregori, J. P. C ´ amara, and G. A. Bada, "A jabber-based multiagent system platform," in Proceedings of the fifth international joint onference on Autonomous agents and multiagent systems. ACM, 2006, pp. 1282–1284.

[17] "Fipa specifications," Online: http://www.fipa.org/, Apr. 2014.

A.3 *"Optimized Microgrid Energy Monitoring and Control System with a Human Machine Interface Based on 3D Gestures", International Journal of Applied Mathematics and Computer Science*

Int. J. Appl. Math. Comput. Sci., , Vol. , No. , –
DOI:

OPTIMIZED MICRO GRID ENERGY MONITORING AND CONTROL SYSTEM WITH A HUMAN MACHINE INTERFACE BASED ON 3D GESTURES

JOÃO RODRIGUES [*,**], PEDRO CARDOSO [*,**], JÂNIO MONTEIRO [*,***], LUÍS SOUSA [*,**], JORGE EDUARDO [*], RICARDO ALVES [*,**]

[*] Instituto Superior de Engenharia, University of the Algarve, Faro, PORTUGAL
e-mail: lcsousa@ualg.pt, jmeduardo@ualg.pt, rmalves@ualg.pt

[**]CIAC and LARSyS (ISR - Lisbon), University of the Algarve, PORTUGAL
e-mail: jrodrig@ualg.pt, pcardoso@ualg.pt

[***]INESC-ID, Lisbon, PORTUGAL
e-mail: jmmontei@ualg.pt

With the integration of Information Technologies and the Internet of Things, it is now possible to communicate with several electrical devices and appliances, implementing a new set of energy management strategies. By being able to perform load-scheduling taking into consideration different variables, such as the forecasted power generation from renewable sources, different tariffs' rates, electrical circuit constraints, user restrictions and correspondent comfort levels, such energy management solutions will constitute an important milestone on the path to the Smart Grid. To achieve it, the integration of several innovative solutions is required, that combine machine-to-machine communications, optimizations algorithms and human-computer/machine interaction using simple and intuitive interfaces. In this paper, we present such a solution. In terms of the optimization algorithm for energy management, a Genetic Algorithm is used to implement the overall scheduling procedure of appliances, taking into consideration the aforementioned variables and constraints. For the human-computer interface, a hands-free 3D gesture recognition solution is used that, when combined with 2D and 3D representations of buildings, objects and menus, supports an intuitive interface between humans and the energy management device, facilitating such interaction in a way that cannot be achieved with other interface paradigms.

Keywords: Energy Optimization, Human-Computer Interaction, Leap Motion, Smart Grids.

1. Introduction

We are witnessing an increase in the energy produced by renewable sources, either motivated by the high cost of fossil fuels or by an increment in environmental restrictions. In this scenario, the traditional view of a distribution grid, that uses centralized generators to provide power to consumers, is being replaced by a Smart Grid solution, where renewable energy sources are being integrated into the grid, following a Distributed Generation (DG) schema (Shen, 2012).

One major drawbacks of some of these sources however, with higher expression in the energy obtained from wind and photovoltaic sources, is that they are many times characterized as Intermittent Resources (IR) since their power varies according to uncontrollable environmental conditions. In such scenario the introduction of demand side management solution is seen as a desirable strategy to control/reduce the necessity in power peak.

By creating a new range of appliances that integrate Information Technologies (IT) and the Internet of Things (IoT) (Guinard, 2011), users have now the possibility to get information, control or adjust every equipment, machine or lamp in their building (e.g., house, hotel or factory) either automatically, semi-automatically or manually. When combined with an optimization solution this not only allows a maximization of investments made in renewable sources, but also a reduction of consumption costs, resulting from the adjustment of the work periods of electrical devices according to energy tariff rates of Distribution System Operators (DSOs).

However load scheduling comes with some challenges. While until now load scheduling has

been performed non-automatically, the introduction of automatic management systems in medium to large scale installations can cause demand hikes at low price periods, leading to overloads and disruption of supply. Thus, Micro Grid energy management systems should take into consideration electrical circuit constraints (Schneider Electric, 2013), while reducing electricity costs and maximizing investments made in renewable sources. One example of such a distributed load scheduling mechanism for micro grids can be found in (Monteiro *et al*., 2014a).

In such system users play a key role. In fact, the flexibility given by them to the load scheduling of appliances plays a decisive factor in its ability to perform cost reduction. It is therefore important for the optimization system to reflect user comfort levels and to interact with them using easy and intuitive interfaces. In a scenario where many home residences, hotels, factories, etc., are moving from dozens to hundreds, or even thousands of electric devices, the usability and ergonomics of the traditional selection menus and sub-menus using a keyboard, or a mouse (or even a touch screen) becomes inappropriate for giving a command, or to query any information from device #X. Numerous examples of possible interactions can be mentioned, such as the request about the consumption of a lamp, or a group of lamps represented in a daily graph, or a cycle changing of a machine (e.g., washing or drying), or the selection of the home scenario that requires less consumption during a week day, or even to check all the consumption statistics of an air-conditioning in a specific location on a hotel.

Nowadays, several types of three-dimensional (3D) sensors, e.g., the widely known Kinect (2014) from Microsoft, can be used to interpret specific human gestures, enabling a completely hands-free control of electronic devices, the manipulation of objects in a virtual world or the interaction with augmented reality applications. These sensors, due to their reduced size and price, can be integrated in several different locations in the buildings, allowing the replacement of the traditional menu interface and peripherals devices. Furthermore, the hands-free interfaces can provide easy navigation on the home residence/hotel/factory plant, with the real localization of the electric devices shown in any television or screen located anywhere on the building rooms. Within those plants, the user could navigate from device to device, individually selecting, turning them on/off or requesting information about them. Another major advantage of these hands-free interfaces is the fact that the interaction can still occur if the user is wearing gloves or has dirty hands (e.g., for instance when cooking or working in a factory).

In this sense, the goal of creating a system capable of energy management is to implement a set of so-called smart objects (Vasseur and Dunkels, 2010), supported in the concept of the Internet of Things, that by communicating with each other and acting based on an optimized control system, allow a better use of the energy produced by renewable energy sources and the reduction of costs (one of such resources is electrical power), while respecting electrical circuit constraints, user restrictions and correspondent comfort levels (Monteiro *et al*., 2014b).

Given these considerations, the main contribution of this paper is the presentation of an energy scheduling optimization algorithm in the context of the energy management, connected with this new Human Computer Interaction (HCI) approach, to support energy monitoring and control applications, where the 3D hands-free interface combine 2D and 3D representations of objects, buildings and configuration options. The system allows humans to interact intuitively with electric devices, in a way that cannot be achieved with other interface paradigms.

The remainder of the paper has the following structure. Section 2 presents the state of the art and a deeper contextualization of the problem in hand. Section 3 addresses the optimization problem and Sec. 4 develops the interaction and the resulting interface. Finally, conclusions and future work are presented in Sec. 5.

2. Contextualization and State of the Art

During the period of 2000 and 2001 the Californian energy crisis took place, marked by a sequence of large-scale blackouts. This crisis was not restrained to California. Also the entire Pacific Northwest and the Southwest were affected by the soaring wholesale prices (Sweeney, 2002). A group of factors contributed to these dual crisis: first electrical and then financial. Other large-scale blackouts happened during the period that goes between 2003 and 2012, in countries such as Germany, France, Belgium, Italy, Austria, Spain, Brazil, Paraguay, Chile, Korea, etc. They have occurred not only in developing countries but also in the less-developed ones (Nishioka, 2014).

In the end of July 2012, for two consecutive days India was affected by massive electricity blackouts. The second outage was the largest in history, leaving more than 600 million people, nearly a tenth of the world's population without electricity (Romero, 2012). Sweeney (2008) analyzed the issue in depth, and concluded that one of the fundamental causes of price volatility in the wholesale electricity markets was the lack of responsiveness of electricity demand to wholesale prices. He proposed a possible solution to the problem, such as making retail prices more responsive to wholesale prices in both in quantitative and speed reaction terms, counting on the consumers interest to reduce electricity purchases when retail electricity prices increase and vice-versa.

Consumers can benefit largely from the green generation of (renewable) energy on-site, not only in

the sense that they should be able to produce energy for self-consumption, but also they can feed the excess of production into the electrical grid. Nevertheless, to achieve the full potential of the production-consumption, the consumers are expected to adjust their demand according to the power that is being produced. The paradigm of the traditional role of consumers is being replaced by a more proactive one.

The role of an Energy Consumption Scheduling (ECS) device is to optimally schedule loads, power consumed by devices and appliances, in order to better harness the energy produced locally or shift them to work at periods of time when the tariff rates are lower, while reflecting user preferences. This also requires that loads, like heating, ventilation and air conditioning (HVAC) and home appliances, should be able to communicate with the ECS device and shift their working periods, or adjust the power they consume, according to the power generated locally from renewable sources, or according to a supplier's tariff, which in turn may change dynamically. In the context of the Energy Management in the Smart Grid, several scientific works have analyzed the importance of an efficient management of home grids by consumers, see e.g., Infield *et al*. (2007), Erol-Kantarci and Mouftah (2010) and Erol-Kantarci and Mouftah (2011).

By creating a range of appliances that integrate IT and IoT (Guinard, 2011), with the ability to communicate between devices, we can respond dynamically to the varying tariffs, and a reduction on CO_2 emissions to the atmosphere is also possible (Erol-Kantarci and Mouftah, 2010), while ensuring at the same time higher returns on investments made in renewable energy sources. In this sense, as described in (Grid, 2011), Smart Appliances are characterized as an important milestone on the path to the Smart Grid.

In the search for a lower cost or tariff, one important feature associated with the ECS is to prevent electrical overloads that may happen when several appliances are scheduled to work at the same time period. The ECS device should reflect a level of priority that should be commensurate with the user's preferences, i.e., the ECS should decide which appliance is expected to work first and which ones should be scheduled to work later. As a matter of fact, while overloads may prevent the user from using the ECS, the quality of the scheduling algorithm from an user perspective will also determine the degree of freedom given by him.

If on one hand the user's ability to manage their energy consumption according to the production is a critical feature of Smart Grids, and a base for innovation, new products and services, a set of new protocols and control mechanisms are required to control and communicate with the consuming objects. In order to support this capability, the communication between different devices such as meters, appliances, electric vehicles, energy management systems and distributed energy resources must occur using secure, standard and open procedures.

In this context, several protocols and architectures have been recently defined, like the Smart Energy Profile (SEP) (ZigBee, 2014), the Standard for Ubiquitous Green Community Control Network Protocol – IEEE 1888 TM (Advisory, 2011) within the Ubiquitous Green Community Control Network Protocol (UGCCNet), or the communication protocol for Smart Grids is the Open Automated Demand Response (OpenADR) version 2.0 (Alliance, 2014).

Whereas protocols that allow communication between different devices of a Micro Grid are being developed, optimization algorithms to minimize costs are also necessary. There is a wide variety of optimization techniques referenced within the scope of Smart Grids. In addition there is the possibility of combining several methods forming hybrid optimization solutions. Metaheuristics such as the Genetic Algorithms (GA), used in this work to solve the scheduling problem, are widely used to solve a large range of problems. GAs are deeply relevant for industrial applications, because they are capable of handling problems with non-linear constraints, multiple objectives, and dynamic components properties, that frequently appear in real problems (Roeva *et al*., 2013).

In summary, GAs are stochastic, parallel algorithms inspired by genetics, evolution theories of natural selection and survival of the fittest. It is an iterative procedure actuating on a population of chromosomes. Each chromosome encodes a candidate solution to the problem. The final fitness, which is a measure of accountability associated with each chromosome, depends on how well the algorithm solves the problem. Among the population of chromosomes, or individuals, the chromosome with the lowest fitness values represent the best solution for a minimization problem. The fitness function is many times supported on the objective function combined with a penalty function, which penalizes potential violations of constraints. The fitness function returns a fitness value determining the ability of the solution to survive and to produce offspring. New generations of solutions are obtained through processes of selection, crossover and mutation. During the evolutionary process, new generations should result in increasingly better solutions and evolve toward an optimal solution. Using elitism replacement, it is possible to preserve a pre-determined number of best individuals from a previous population. This process is depicted in the flowchart presented in Fig. 1

Other algorithms could be used to tackle the problem such as Differential Evolution, Ant Colony Optimization, firefly or Artificial Bee Colony algorithms (Hamid *et al*.,

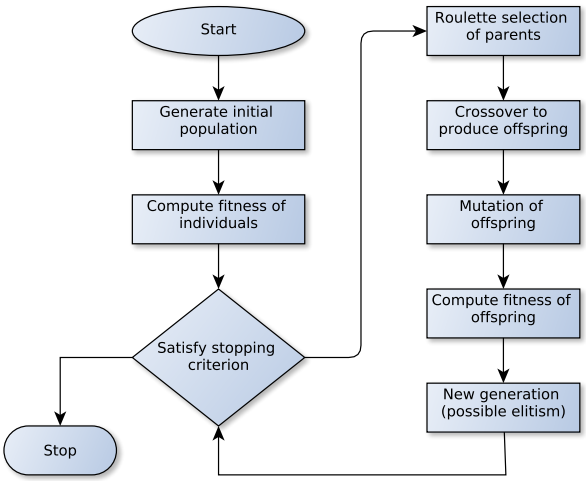


Fig. 1. Genetic algorithm flowchart.

2012; Bhattacharyya *et al*., 2012; Younes *et al*., 2013).

Having the consuming objects (e.g., devices, appliances) connected and the cost of exploration minimized, it is now necessary to provide the management of all these distributed resources by the consumer/user. On other words, a human-computer/machine interface is necessary.

In terms of Human Machine Interface (HMI), several solutions can nowadays be used by ECS devices to obtain and set user preferences. Traditionally, these interfaces have been based in touchscreens, or communicate with users using a graphical user interface (GUI) that can be accessed from personal terminals like PCs, tablets or mobile phones. Nevertheless, all solutions present specific limitations being, at least one common to almost all of them: the impossibility to use thick gloves or dirty hands. Besides, the interaction between humans and machines can be made in many different ways. A different solution is the use of Voice User Interfaces (VUI), that uses speech recognition to control devices and even to the recognition of emotions (Rybka and Janicki, 2013). More recently, multimodal interfaces (Dumas *et al*., 2009) allow humans to interact with machines in a way that cannot be achieved with other interface paradigms.

One of the new paradigms for human computer interaction are the three-dimensional (3D) sensors, such as the Microsoft Kinect (Kinect, 2014), the Asus Xtion (Asus, 2014), the Leap Motion (Leap, 2014b), or the Structure Sensor (Struture, 2014). Those sensors can be used to interpret specific human gestures, enabling a completely hands-free control of electronic devices, the manipulation of objects in a virtual world or the interaction with augmented reality applications. Many of these tracking and gesture recognition sensors have a huge importance in the video-games industries. Hence, with the appropriate software, they have also the capability to detect the user skeleton and/or tracking a single or several users, while replicating with accuracy the hands and the user movements in a 3D mesh.

There is in the literature an enormous amount of applications, where gesture recognition and tracking is referenced, using vision based systems, e.g., for hand gesture orientation (Subramaniam *et al*., 2013), for detecting and recognizing static hand poses and interpret pose sequences in terms of gestures (Kasprzak *et al*., 2012), or hand and gesture detection for interfaces applications (Saleiro *et al*., 2013).

In terns of gesture recognition and tracking using 3D sensors, we can refer for instance, pose and hand control of interactive art installations (Alves *et al*., 2014), applications to help disable or old people (Chung *et al*., 2014), air painting application (Sutton, 2013), education (Figueiredo *et al*., 2014), robotic arm manipulation (Bassily *et al*., 2014), hand gesture recognition in 3D space (Ahn and Jung, 2012), or applications in sign language (Potter *et al*., 2013).

One of the above mentioned sensors, the Leap Motion, is a recent but widely known sensor for hand, fingers and gesture recognition, with a very high accuracy and speed, and it is one of the sensors mentioned in the above publications, that, due to its size, price and specific range of applications, has many potentials of use (Sutton, 2013; Potter *et al*., 2013; Bassily *et al*., 2014). The Leap Motion uses two monochromatic infrared (IR) cameras and three infrared LEDs. For more details see (Spiegelmock, 2013). A smaller observation area and a higher resolution differentiate it from the Kinect and Xtion sensors, which are more suitable for body tracking. To use it, users place their hands in front of the device, not too close, nor too far (in height and width) and execute predefined movements.

In the context of this work, in order to enhance the user's experience, it is used the Leap Motion sensor combined with a 2D and 3D representation of options concerning energy management of home appliances. Currently several solutions can be used to make such 2D/3D representation. One of these solutions is the Unity cross-platform game engine (Unity, 2014). Unity is a game creation system developed by Unity Technologies that includes a game engine and Integrated Development Environment (IDE). Unity is currently used to develop video games for web sites, desktop platforms, consoles, and mobile devices. Simultaneously, Unity is now the default Software Development Kit (SDK) for the Nintendo Wii and has been extended to target more than fifteen platforms (Unity, 2014). This platform is also the ideal one to create the graphics interfaces for our application.

3. Optimization

As stated before, in this work the scheduling of electric devices is done by a Genetic Algorithms. The optimization process presented in this paper is based on a previous work, also supported on GA, where we describe a distributed micro grid resource management architecture and protocol for large micro-grids (Monteiro *et al.*, 2014a). Having several aggregator (or ECS) levels, in Monteiro *et al.* (2014a) we concentrate in the coordination between ECSs and in a protocol that enables their communication. In this work we focus on a single aggregator device (house, factory, etc.) giving a detailed description of the optimization process and parameters.

The architecture was implemented on a multi-agent simulator. The simulator was specified to implement a system where each object (meters, generators and consumers) is an agent within a Multi-Agent System (MAS) (Shoham and Leyton-Brown, 2009).

Taking into consideration the specifications, the simulator was implemented using Smart Python Multi-Agent Development Environment (SPADE) (Argente *et al.*, 2007). SPADE was built around the XMPP/Jabber communication framework and is developed in Python, showing to be a particularly useful system in the implementation of MAS. Its usefulness comes from the support to: (1) the concepts of agent and servers, (2) the implemented communications between agents, (3) the possibility to develop agents in multiple programming languages, (4) the processing of agent behaviors and (5) the extensible communication protocol based on XML. Furthermore, SPADE follows the FIPA specifications for MAS (FIPA, 2014).

Regarding to optimization, the Pyevolve framework (Perone, 2009) which implements the main GA Algorithm was used. The chromosomes were encoded as a list of integers (epoch times) where each element corresponds to the beginning (initial time) of a consumer machine program, e.g., $[t_1, t_2, \ldots, t_n]$. This means that the fist appliance will start at t_1, the second appliance will start at t_2, and so on. Genetic Algorithms work combining selection, recombination and mutation operators. The selection pressure drives the population toward better solutions while recombination uses genes of selected parents to produce offspring that will form the next generation. Mutation is used to escape from local minima.

GA mutators: The GA was configured with three mutation operators, as explained next. The first one is the (1) Swap mutator which exchanges genes in a chromosome (see Fig. 2 top). The swap of the genes (programmed loads starting times) varies the chromosome without "loosing the best spots" to place the loads. The second mutator is the (2) Integer Gaussian mutator which uses a random integer number influenced from a Gaussian distribution with deviation and mean parameters

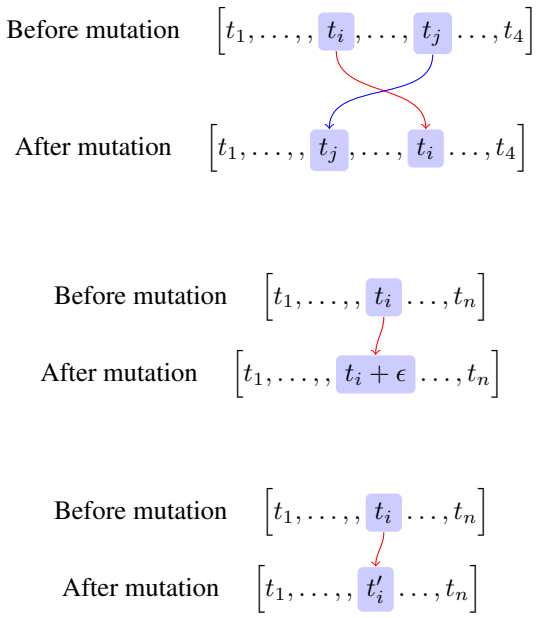


Fig. 2. Genetic algorithm mutation operators: Swap mutator (top); the Integer Gaussian mutator (middle), ϵ is a sample from a Gaussian distribution with mean μ and standard deviation σ, $N(\mu, \sigma)$; and the "Pseudo-Greedy" mutator (bottom), t'_i is a pseudo-randomly chosen so that null or lower cost tariff periods have larger probability of being chosen.

pre-configured, to slightly move the charges around their current position (see Fig. 2 middle row, where ϵ is a sample from a Gaussian distribution with mean $\mu = 0$ and standard deviation σ, $N(\mu, \sigma)$). This mutator also as the effect of not "loosing the best spots" to place the loads, since in general only slight changes are made to the initial time of the programs. In addition to the previous mutators, it was defined a (3) "Pseudo-Greedy" mutator that uses a roulette to influence the genome by giving better probabilities of moving charges to the lower cost tariff periods or to the predicted cost zero intervals due to renewable power forecast (see Fig. 2 bottom, where t'_i is a pseudo-randomly chosen so that zero or lower cost tariff periods have larger probability of being chosen). This "Pseudo-Greedy" mutator is more disruptive than previous ones since the offspring chromosomes can be significantly different from the parents.

GA Crossover: A single point crossover was used. The single point crossover gets two parent chromosomes and defines a cutting points. Data beyond that point in either chromosome is swapped between the two parent, producing the offspring (see Fig. 3).

GA selection: The selection method chosen was the roulette wheel. In this case the parents are selected according to their fitness (see Eq. (1)), i.e., chromosomes

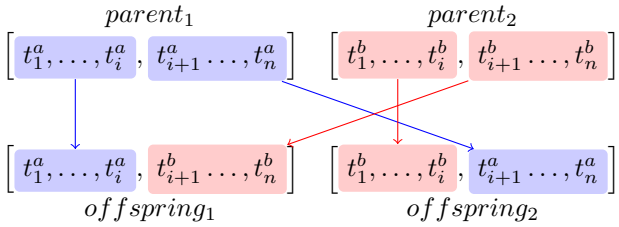


Fig. 3. Genetic Algorithm - single point crossover.

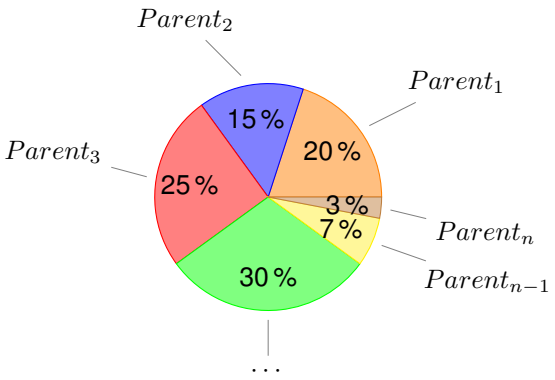


Fig. 4. Genetic Algorithm - roulette wheel.

with better fitness get a larger chance of being selected. As the operator's name suggests, the process can be figured as a roulette wheel where the slices are bigger or smaller accordingly to the corresponding chromosome fitness function. Figure 4 shows a distribution of roulette slices, where $Parent_1$ has 20% chance of being selected, $Parent_2$ has 15% chance of being selected, etc.

The remaining configuration of the GA was set to: (a) population size was equal to 20, the (b) generations number was 100, the (c) mutation rate was 0.05 and the (d) crossover rate was 0.9.

Fitness function: In the current scenario the load management procedure is assumed to be done in a forecasted search space of 24 hours (it could however be adapted to a wider one). A system was also considered where loads are not differentiated by priorities, i.e., all the loads have the same scheduling priority. However, we considered that different load curve profiles are known, as exemplified by Fig. 5, where the power consumptions of four types of appliances are presented, namely: (a) pool pump, (b) dish washer, (c) drying machine, and (d) electric storage water heater.

Load curves are used to compute the fitness function, which was set as

$$F = \frac{1}{Q} \sum_{t \in T} \left(\alpha(t) C(t) + \beta(t) \right), \tag{1}$$

where Q translates the quality assessment of the scheduling solution seen from a user perspective (Monteiro *et al.*, 2014b), T represents a set of time intervals, $C(t)$ translates the cost of energy for interval t,

$$\alpha(t) = \begin{cases} [P(t) - R(t)]\,\Delta_t & \text{if } P(t) > R(t) \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

and

$$\beta(t) = \begin{cases} \infty & \text{if } P(t) > P_{max} \vee \\ & \vee t < t_{min,i} \vee \\ & \vee t + w_i > t_{max,i} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

In equations (2) and (3), $P(t)$ translates the sum of load's power scheduled to work at the time interval t; P_{max} translates the power limit; Δ_t translates the time period discretization window; $t_{min,i}$ is the instant from which the i-appliance can be programmed to work; $t_{max,i}$ is the instant until which the i-appliance has to finish its program; w_i the length of the i-appliance's program; and R translates the forecasted generation vector of renewable sources. On other words, α is the power consumed from the electrical provider during the Δ_t period and the value imposed in Eq. (3) has two objectives: prevention in case of an erroneous prediction of the power generated by renewable resources and ensure that the appliance does its work between $t_{min,i}$ and $t_{max,i}$, i.e., β is a penalty value activated whenever the sum of load's power scheduled to work at the time interval t overcomes the power limit or the users time intervals are not satisfied.

The overall scheduling procedure is a dynamic process, in the sense that an appliance can request a scheduling at any time and can be summarized in the following steps:

Step 1. Start the aggregator (scheduler).

Step 2. The aggregator is put on wait for a scheduling request.

Step 3. An appliance is programmed (usually by a human) and sends a scheduling request to the aggregator.

Step 4. The aggregator takes into consideration the appliance characteristics and configuration (e.g., start after - $t_{min,i}$, end before - $t_{max,i}$, specific program load curve) and uses the GA to find a optimal spot for the appliance to work.

Step 5. The aggregator informs the appliances about their new scheduling (this implies that an appliance can be rescheduled to work on a different period from the one it was previously informed).

Step 6. Return to Step 2.

It is important to notice that after the first scheduling request, to attain a new scheduling for a new appliance

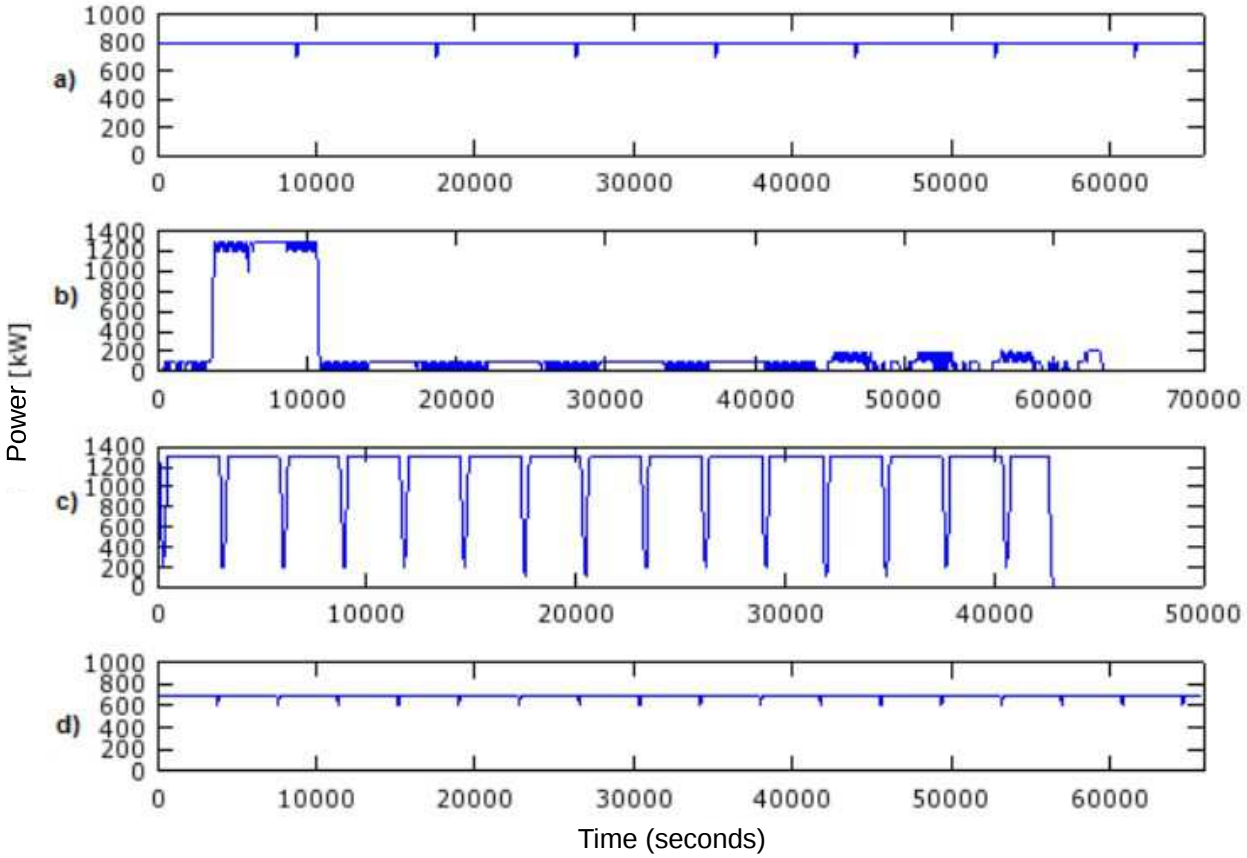


Fig. 5. Examples of the power consumption of four types of appliances: a) pool pump, b) dish washer, c) drying machine and d) electric storage water heater.

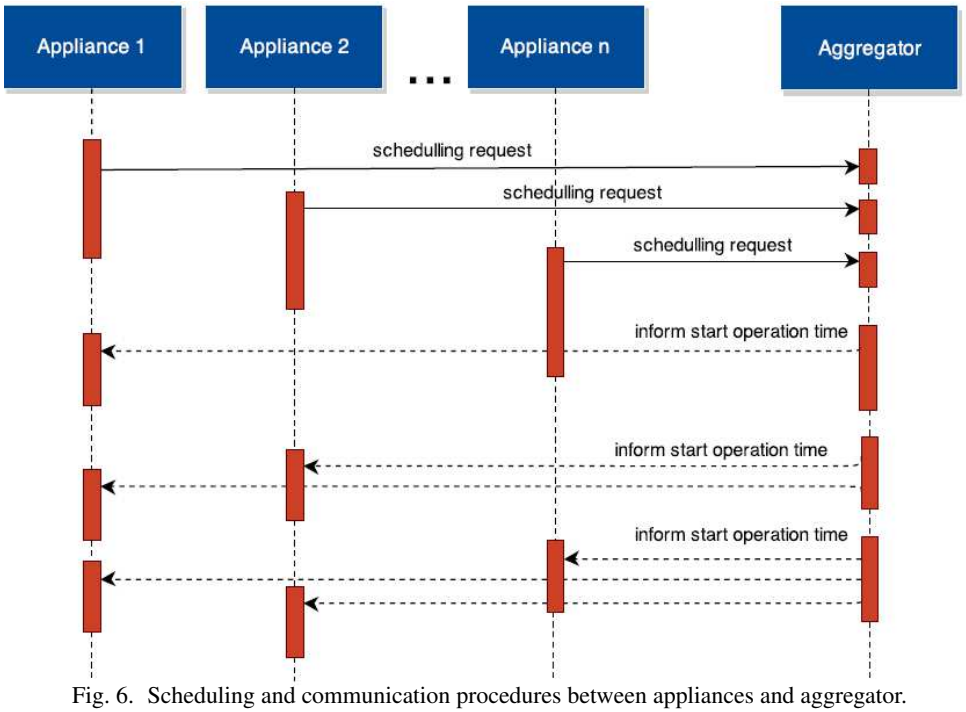


Fig. 6. Scheduling and communication procedures between appliances and aggregator.

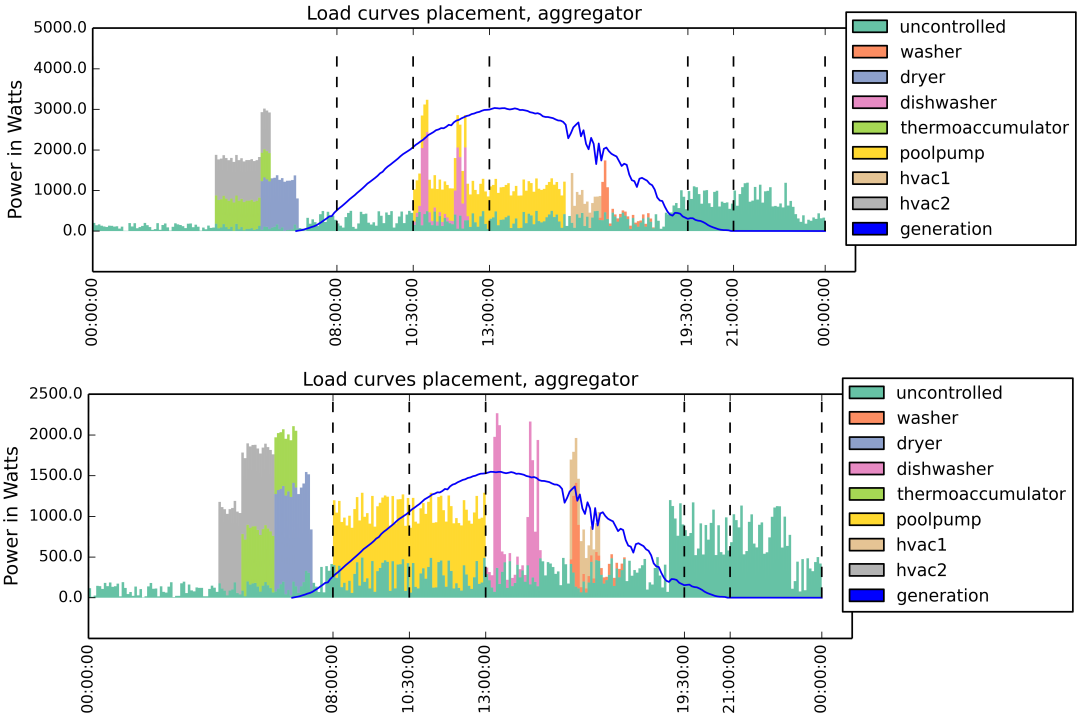


Fig. 7. Top, the load placement resulting from the scheduling algorithm over a 24 time window for the first scenario (SC1), bottom the same but for the second scenario (SC2); see text for details.

request, the scheduler reruns the optimization algorithm from the point it was stopped before. Charges already finished or started in between are first removed. This mean that the between runs a memory is kept, which enhances the performance and keeps a more steady solution for the already informed schedules. Figure 6 depicts the scheduling procedure.

Some results will be presented below. Although the presented examples only take into consideration a small number of loads the described process can take care of a much larger number of loads (Monteiro *et al.*, 2014a).

Simulation results: In the first scenario we have considered seven appliances, namely: two HVAC, a thermo accumulator, a dishwasher, a washer, a dryer, and a pool pump. Table 2 shows requested working time windows of the appliances and the corresponding working time length. The tariff was set with 3 price periods as summarized in Tab. 1.

Two scenarios were set for the power limits. In the first scenario (SC1) it was considered: (a) a contracted power of 6.6 KVA but with a maximum programmable load (P_{max}), at any time, of 4.5 KVA. This restriction has to do with the possibility of occurrence of uncontrollable (not scheduled) charges. Furthermore, (b) a power generation curve obtained from a solar photovoltaic plant with a peak production of 3 kW (blue line in Fig. 7) was considered.

Figure 7 top shows the load placement resulting from the scheduling algorithm over a 24 time window. In dark green are the uncontrollable charges common to a home residence (e.g., lights, TV). The loads where place to minimize the costs, satisfying the users preferences. For example, the pull pump was place on the renewable energy generation period.

The second scenario (SC2) is slightly more demanding. In this case (a) the contracted power was 3.3 KVA but with a maximum programmable load (P_{max}), at any time, of 2.5 KVA. This is justified since lower contracted power has in general a cheaper contract. With respect to the power generation, it was considered a curve from the solar photovoltaic plant with a peak production of 1.5 kW. Figure 7 bottom show the load placement for this last case.

It can be see from the figures that the charges where scheduled taking as the main priority the user's preferences but also the cheapest overall cost. Furthermore, it is shown that with the proper scheduling, it is possible to contract a cheapest power contract without problems of incurring in overloads.

Having executed the optimization algorithm, and having defined the device/appliance schedule along the day, it is then necessary to inform the user about the result, enabling his interaction in real time with the system. This will allow the user to become aware of how much he/she is benefiting (i.e., saving), and which device/appliance is scheduled at any specific hour of the day, but mainly, to

Table 1. Tariff with 3 price periods in hh:mm and respective price in Euros.

Tariff	Time window	price
T_0	$[00:00, 8:00]$	0.0955
T_1	$[8:00, 10:30] \cup [13:00, 19:30] \cup [21:00, 24:00]$	0.1642
T_2	$[10:30, 13:00] \cup [19:30, 21:00]$	0.2066

Table 2. User programmed appliances working time windows ($[t_{min}, t_{max}]$) and corresponding program lengths (w_i) in hh:mm.

Appliance	$[t_{min}, t_{max}]$	w_i
HVAC1	[16:00,20:00]	0:58
HVAC2	[5:00, 8:30]	1:49
Thermoaccumulator	[4:00, 9:30]	1:49
Dishwasher	[7:00, 21:00]	1:47
Washer	[16:00, 21:00]	1:48
Dryer	[4:00, 8:00]	1:12
Poolpump	[7:00, 6:59]	5:00

permit any change of the parameter/schedules in order to adapt the system to the subjective preferences of the user. In the next section the interface and the main interactions with the system is going to be presented.

4. Interface and Interaction

The Leap Motion uses the 3D space as a standard Cartesian coordinate system. The origin of the coordinate system is centered at the top of the device, being the front of the device the side with the green light; for an illustration see Leap (2014a). The x-axis is placed horizontally along the device, with positive values increasing from left to right. The z-axis is placed also on the horizontal plane, perpendicular with x-axis and with values increasing towards the user (the front side of the device). The y-axis is placed in the vertical, with positive values increasing upwards.

By using the Application Programming Interface (API), the Leap Motion sensor is capable of detecting multiple hand gestures, such as straight line, fingers extended, a circle movement using a finger, a forward tapping movement using a finger and a downward tapping movement. Examples and illustrations of these gestures (swipe, circle, screen tap and key tap) can be seen in (Leap, 2014b) and (Leap, 2014a).

To improve gesture detection there are a few optional configurations properties. For example in a circle gesture, where the user can do a circle with a finger, there are two selectable properties: *minimum radius* and *minimum arc* (by default, *minimum radius* was set to $5mm$ and *minimum arc* was set to 1.5π radians). For a swipe gesture, there are also two selectable properties, *minimum length*, set by default to $150mm$, and *minimum velocity*, set to

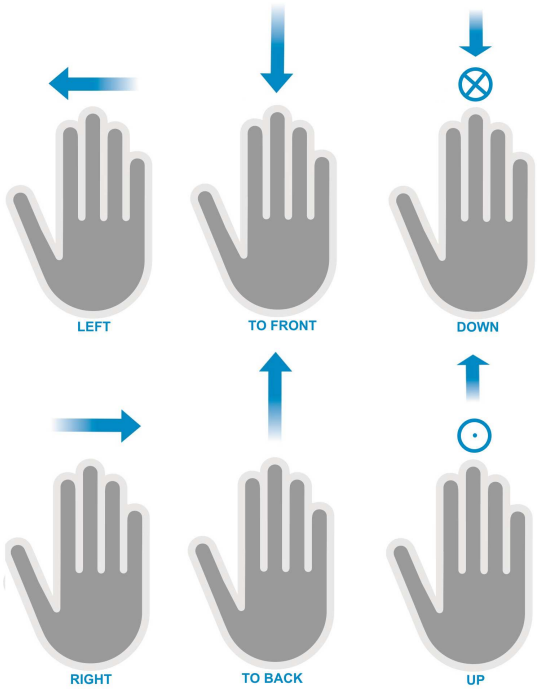


Fig. 8. Six types of swipes for the application.

$1000mm/s$. It is also possible to join some of these gestures, like the positions and rotation of each finger and the use of both hands at the same time, to recognize other gestures, such as open or close hand which can be done with one or both hands.

For the development of the interface we tried to define a short number of movements, and at the same time select the most intuitive ones - the swipe, was considered as the best approach. Thus, swipe gesture was the gesture mainly used in the interface developed in this paper, i.e., for the interaction with the energy monitoring and control applications. After regulating the minimum length swipe and velocity to $100mm$ and $400mm/s$, respectively, it is possible to detect swipe gestures at any direction.

Leap Motion API has a direction vector for the swipe gesture; a gesture completely recognized is associate with a 3D direction vector. This vector has values ranging from -1.0 to $+1.0$. As different types of swipe gesture exist we needed to detect and differentiate them.

The interface was designed to react to six different independent types of swipe gestures as shown in Fig. 8, three of them, are the opposite of the other three:

(i) Select and deselected is realized by a top to bottom

swipe or a bottom to top swipe, respectively (y-axis). In this case, of a top to bottom and bottom to top swipe, the movement depends mainly on the y-axis. If the direction vector has an upward direction ($y \approx +1$) then a "deselected" action has occurred. Otherwise, if the vector has a downward direction ($y \approx -1$), then it is considered a "select" action. Since it is almost impossible to do a swipe gesture with a vector direction component of exactly,

$$x = 0 \wedge y = \pm 1 \wedge z = 0, \tag{4}$$

the algorithm should instead select a range of values to detect and differentiate between swipes types.

So, any swipe direction that agrees with the condition

$$y \leq -0.5 \wedge |x| \leq 0.5 \wedge |z| \leq 0.5, \tag{5}$$

is considered a downward swipe. Contrariwise, if a swipe direction agrees with the condition

$$y \geq 0.5 \wedge |x| \leq 0.5 \wedge |z| \leq 0.5, \tag{6}$$

then it is considered as an upward swipe.

(ii) Select different floors (see Fig. 9) of a house is realized by a front to back and back to front swipes, to select lower and upper floors, respectively (z-axis). In this case, as shown on Fig. 8, a vector with direction in z, who has a value approximately 1, is considered to be a back to front swipe. Otherwise, it is considered as a front to back swipe. Similar to (i), any swipe direction that agrees with the condition

$$z \leq -0.5 \wedge |x| \leq 0.5 \wedge |y| \leq 0.5, \tag{7}$$

is considered as a front to back swipe. Contrariwise, if a swipe direction agrees with the condition

$$z \geq 0.5 \wedge |x| \leq 0.5 \wedge |y| \leq 0.5, \tag{8}$$

then it is considered a back to front swipe.

(iii) Selecting the next and previous item/object is done by a left to right or right to left swipe, respectively (x-axis), again it is similar to (i) and (ii). Any swipe direction that agrees with the condition

$$x \leq -0.5 \wedge |z| \leq 0.5 \wedge |y| \leq 0.5, \tag{9}$$

is considered as a right to left swipe. Contrariwise, if a swipe direction agrees with the condition

$$x \geq 0.5 \wedge |z| \leq 0.5 \wedge |y| \leq 0.5, \tag{10}$$

then it is considered as a right to left swipe.

These swipes are mutually independent, for every type of swipes there is only one possible choice, see Fig. 8

In order to build a proof of concept, a house similar with a real one was created using Sweet Home 3D (SweetHome, 2014). This home includes three floors and a garden with an outside pool, creating 4 different areas, as represented in Fig. 9. The control interface was built using the same number of floors and rooms.

Each of those areas has a set of different electrical devices/appliances. For example in the ground floor (Fig. 9, 3rd row), there are various types of appliances such as a fridge, a stove, a washing machine, a drying machine, an air conditioning and various types of lightings. It is important to stress at this time, that not all of the devices/appliances that can be selected in the interface are used in the optimization algorithm. For instance while the information about lightings (lamps) can be accessed, they weren't scheduled by the optimization algorithm. Regarding the other appliances, such as the pool pump, dish washer, drying machine, etc. all the information and full interaction was made available, including the creation of new programs, scheduled programs and statistics.

The interface was made in Unity 3D (Unity, 2014). It was divided in 3 different levels. In the first level, (i) the user can select between different floors/zones. In the second level, (ii) the user can select the different machines in the selected floor. In the final level, and after selecting a machine, (iii) a picture of the selected machine is shown together with some information regarding statistics, scheduled programs and the option to add new programs.

In the first level of the application, the four different zones were positioned one above the other, as shown in Fig. 9. On the second level, each electric device was considered as an individual object. The selection of that appliance could be represented through an image rescaling or by changing its colour, see Fig. 10 top row; the device is mark with the red colour. After performing some tests with users, it was decided to mark the selected object with an arrow, see Fig. 11, 2nd row.

Figure 10 also shows some of the other option that are available for the view representation of the house plant, that can be "top view", as in Fig. 9 (or in Fig. 11), or can be seen as "3D view" as can be seen in Fig. 10, bottom two rows. This menu is available in every plan (e.g., floor, house, hotel, factory), and is selected as any normal device. The enumeration of all the options available is out of the focus of the paper, nevertheless it is important to mention that the navigation on those options are done only with 6 swipes.

Figure 11 shows the interface being operated. At the top, from left to right, the ground floor is shown, which is the default zone/floor selected, and the navigation in the floors: basement and the first floor, plus the

Fig. 9. Four different zones. From top to bottom: outside, first floor, ground floor and basement.

Fig. 10. Top row, the device mark with the red colour instead of an arrow. Two bottom rows, two possible 3D views of two different house zones, showed in Fig. 9.

selection of the last. In the second row, the selected ground floor is shown as well as the navigation among the devices available in that floor, being selected the watching machine. Bottom two rows the different options available for this device.

As explained previously, there are six different swipes that the user can do to navigate in the interface with the Leap Motion. To move from different floors/zones, the user has to do swipe movements along the z-axis of the Leap Motion. A back to front swipe moves the interface to the basement and a front to back swipe moves the interface to the first floor and, if repeated, to the outside zone.

A downward swipe selects the zone in the middle of the screen and the second application level appears. Simultaneously, the zone plan is zoomed in, to fill the entire screen, as Fig. 11 top row right and 2nd row shows. When doing the opposite swipe (i.e., upwards swipe) the zone is deselected and changed to the first level of the application.

A default appliance is selected when one zone/floor is selected. In the example, the default machine in the ground floor is the fridge in the kitchen, and is marked with a green arrow, see Fig. 11, 2nd row left image. The swipes along the x-axis are used to select between different appliances (e.g., machines, lights, oven, and air conditioner). Doing a right to left swipe changes the selection for the next machine, while a left to right swipe changes to the previous selection.

The green arrow only "pre-selects" the machine. Doing a downward swipe selects the machine pointed by the green arrow, and changes to the third and last application level. In this last level, the 3D model of the selected machine (appliance/device) is shown together with a new menu showing information about the associated machine. As shown in Fig. 11, 2nd row right, after selecting, e.g., the washing machine, all the available information about the respective device is shown in a menu.

The last menu is composed by four different selectable options; Fig. 11 two bottom row (respectively, from left to right and top to bottom): New program, Scheduled Programs, Statistics and Information. To change between those options the user can do a swipe around the x-axis. If a left to right swipe is done the option from the right menu is selected, otherwise, in case a right to left swipe is done, the option from the left menu is selected.

5. Conclusions and Future Work

In this paper an optimization procedure is presented to schedule appliance with the objective of minimizing the operating cost while maintaining the users preferences. The load-scheduling procedure takes into consideration different variables, such as the forecasted power generation from renewable sources, different tariffs' rates, electric circuit constraints, user restrictions and correspondent comfort levels. The proposed algorithm performs the energy management supported on a Genetic Algorithm.

Complementary, a combined 3D gesture recognition solution and 2D/3D representation of buildings, objects and configuration options is explored, that allows humans to interact intuitively with electronic devices, in a way that cannot be achieved with other interface paradigms. The initial tests with a small group of users (10) have shown very promising results, as all the opinions were favorable to the interface, nevertheless the proposed system requires more tests, with more users, and adjustments to reach the final version. Once, some usability problems were marked, as well as design flaws.

Some small problems were also noticed with the Leap Motion sensor, since it does not work properly in some lighting conditions. Sometimes, gestures can be triggered/detected by moving the hand above the device, when the user do not want to do a swipe gesture (to minimize this, tests with several people were done, and the configurations properties were changed accordingly; see section 4).

It was also noticed that some users do the swipe movement in a diagonal (not in the direction that agrees totally with the conditions predefined in section 4), in this case, as expected the system does not respond, nevertheless, this aspect has to be resolved in the future, since it was noticed that this is a natural movement that users tend to do.

In terms of future work, a bigger case study is being prepared with more electric devices and rooms. The scheduling process has to be optimized to properly work on small computers (e.g., single board computers such as the Raspberry Pi or the BeagleBone). In terms of the HCI, other 3D sensors, such as the Structure Sensor are also going to be explored, in order to create an even more intuitive interface, as well as minimize some false positives and negatives obtained by the swipe movement.

Acknowledgment

This work was supported by FCT project LARSyS (PEst-OE/EEI/LA0009/2013), CIAC (PEstOE/EAT/UI4019/2013) and projects QREN I&DT n. 33845, "PRHOLO: The realistic holographic public relations" (project leader SPIC Creative Solutions [http://www.spic.pt/]) and QREN I&DT n. 30260 "MTI: Managing the intelligence" (project leader Certigarve [http://www.certigarve.pt/]).

References

Advisory, C. (2011). IEEE standard for ubiquitous green community control network protocol IEEE-SA board of

Fig. 11. The system in operation. At the top, from left to right, the ground floor is shown, which is the default floor selected, and the navigation in the floors: basement and the first floor, plus the selection of the last. In the second row, the selected ground floor and the navigation among the devices available in that floor, being selected the watching machine. Bottom two rows the different options available for this device.

governors, *IEEE Std* **2011**: 10016–5997.

Ahn, Y. and Jung, K. (2012). Implementation of a word suggestion keypad system utilizing a 3D space hand gesture recognition, *Recent Researches in Telecommunications, Informatics, Electronics and Signal Processing*, WSEAS, pp. 333–339.

Alliance (2014). OpenADR 2.0 profile specification, a profile, Online: http://www.openadr.org/specification. Retrived: Nov. 10, 2014.

Alves, R., Madeira, M., Ferrer, J., Costa, S., Lopes, D., da Silva, B. M., Sousa, L., Martins, J. and Rodrigues, J. (2014). Fátima revisites: an interactive installation, *In Proc. International Multidisciplinary Scientific Conference on Social Sciences and Arts*, SGEM, pp. 141–148.

Argente, E., Palanca, J., Aranda, G., Julian, V., Botti, V., Garcia-Fornes, A. and Espinosa, A. (2007). Supporting agent organizations, *Multi-Agent Systems and Applications V*, Springer, pp. 236–245.

Asus (2014). Asus Xtion Pro, http://www.asus.com/pt/Multimedia/Xtion_PRO/. Retrived: Nov. 10, 2014.

Bassily, D., Georgoulas, C., Guettler, J., Linner, T. and Bock, T. (2014). Intuitive and adaptive robotic arm manipulation using the leap motion controller, *In Proc. 41st International Symposium on Robotics (ISR/Robotik 2014)*, VDE, pp. 1–7.

Bhattacharyya, B., Gupta, V. K. and Goswami, S. (2012). Application of DE and PSO algorithm for the placement of FACTS devices for economic operation of a power system, *WSEAS Trans. Power Syst* **7**(4): 209–216.

Chung, I.-C., Huang, C.-Y., Yeh, S.-C., Chiang, W.-C. and Tseng, M.-H. (2014). Developing Kinect games integrated with virtual reality on activities of daily living for children with developmental delay, *Advanced Technologies, Embedded and Multimedia for Human-centric Computing*, Springer, pp. 1091–1097.

Dumas, B., Lalanne, D. and Oviatt, S. (2009). Multimodal interfaces: A survey of principles, models and frameworks, *Human Machine Interaction*, Springer, pp. 3–26.

Erol-Kantarci, M. and Mouftah, H. T. (2010). The impact of smart grid residential energy management schemes on the carbon footprint of the household electricity consumption, *In Proc. IEEE Electric Power and Energy Conference (EPEC)*, IEEE, pp. 1–6.

Erol-Kantarci, M. and Mouftah, H. T. (2011). Wireless sensor networks for cost-efficient residential energy management in the smart grid, *IEEE Transactions on Smart Grid* **2**(2): 314–325.

Figueiredo, M., Sousa, L., Cardoso, P., Rodrigues, J., Gonalves, C. and Alves, R. (2014). Learning technical drawing with augmented reality and holograms, *In Proc. 13th International Conference on Education and Educational Technology*, WSEAS, pp. 11–20.

FIPA (2014). FIPA specifications, Online: http://www.fipa.org/. Retrived: november 10, 2014.

Grid, I. S. (2011). Arrival of smart appliances is a milestone on the path to the smart grid, http://goo.gl/F9RUz9. Retrived: Nov. 10, 2014.

Guinard, D. (2011). What the internet-of-things will mean for the smart-grid, IEEE smart grid, http://goo.gl/tp34nu. Retrived: Nov. 10, 2014.

Hamid, Z., Musirin, I., Rahim, M. and Kamari, N. (2012). Optimization assisted load tracing via hybrid ant colony algorithm for deregulated power system, *WSEAS Transactions on Power Systems* **7**(3): 145–158.

Infield, D. G., Short, J., Home, C. and Freris, L. L. (2007). Potential for domestic dynamic demand-side management in the UK, *In Proc. IEEE Power Engineering Society General Meeting*, IEEE, pp. 1–6.

Kasprzak, W., Wilkowski, A. and Czapnik, K. (2012). Hand gesture recognition based on free-form contours and probabilistic inference, *International Journal of Applied Mathematics and Computer Science* **22**(2): 437–448.

Kinect (2014). Kinect for windows, http://www.microsoft.com/en-us/kinectforwindows/. Retrived: Nov. 10, 2014.

Leap (2014a). Leap developer portal, http://goo.gl/yKddrN. Retrived: Nov. 10, 2014.

Leap (2014b). Leap motion, https://www.leapmotion.com/. Retrived: Nov. 10, 2014.

Monteiro, J., Cardoso, P. J. S., Serra, R. and Fernandes, L. (2014b). Evaluation of the human factor in the scheduling of smart appliances in smart grids, *Universal Access in Human-Computer Interaction. Aging and Assistive Environments*, Vol. 8515 of *LNCS*, Springer, pp. 537–548.

Monteiro, J., Eduardo, J., Cardoso, P. and Semião, J. (2014a). A distributed load scheduling mechanism for micro grids, *In Proc. IEEE International Conference on Smart Grid Communications (IEEESmartGridComm)*, IEEE, pp. 1–2.

Nishioka, A. S. (2014). Power system stabilization, *Future Renewable Energy and Smart Grid Technologies*, CIGRE TNC Technical Seminar, Hitachi, Ltd.

Potter, L. E., Araullo, J. and Carter, L. (2013). The leap motion controller: a view on sign language, *In Proc. 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, ACM, pp. 175–178.

Roeva, O., Fidanova, S. and Paprzycki, M. (2013). Influence of the population size on the genetic algorithm performance in case of cultivation process modelling, *In Proc. Federated Conference on Computer Science and Information Systems (FedCSIS)*, IEEE, pp. 371–376.

Romero, J. J. (2012). Blackouts illuminate India's power problems, *Spectrum, IEEE* **49**(10): 11–12.

Rybka, J. and Janicki, A. (2013). Comparison of speaker dependent and speaker independent emotion recognition, *International Journal of Applied Mathematics and Computer Science* **23**(4): 797–808.

Saleiro, M., Farrajota, M., Terzić, K., Rodrigues, J. M. and du Buf, J. H. (2013). A biological and real-time framework for hand gestures and head poses, *Universal Access in Human-Computer Interaction. Design Methods, Tools, and Interaction Techniques for eInclusion*, Springer, pp. 556–565.

Schneider Electric (2013). *Electrical installation guide 2013. According to IEC international standards*.

Shen, X. S. (2012). Empowering the smart grid with wireless technologies [Editor's note], *Network, IEEE* **26**(3): 2–3.

Shoham, Y. and Leyton-Brown, K. (2009). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, Cambridge University Press.

Spiegelmock, M. (2013). *Leap Motion Development Essentials*, Packt Publishing Ltd.

Struture (2014). Struture sensor, `http://structure.io/`. Retrived: Nov. 10, 2014.

Subramaniam, C., Balaji, A., Raj, V. and V.C., H. (2013). Data driven automated model using intelligent hand gesture orientation and recognition system, *Recent Advances in Computer Science*, WSEAS, pp. 168–173.

Sutton, J. (2013). Air painting with Corel Painter freestyle and the leap motion controller: a revolutionary new way to paint!, *ACM SIGGRAPH 2013 Studio Talks*, ACM, p. 21.

Sweeney, J. L. (2002). The California electricity crisis: Lessons for the future, *BRIDGE-WASHINGTON-* **32**(2): 23–31.

Sweeney, J. L. (2008). *The California electricity crisis*, Hoover Press.

SweetHome (2014). Sweet home 3D, `http://www.sweethome3d.com/pt/`. Retrived: Nov. 10, 2014.

Unity (2014). Unity, `http://unity3d.com`. Retrived: Nov. 10, 2014.

Vasseur, J.-P. and Dunkels, A. (2010). *Interconnecting smart objects with ip: The next internet*, Morgan Kaufmann.

Younes, M., Kherfane, R. L. and Khodja, F. (2013). A hybrid approach for economic power dispatch., *WSEAS Transactions on Power Systems* **8**(3).

ZigBee (2014). Smart energy profile 2, application protocol standard, zigbee public document 13-0200-00, `http://goo.gl/zj0BAX`. Retrived: Nov. 10, 2014.

João Rodrigues Professor João Rodrigues graduated in Electrical Engineering in 1993, he got his M.Sc. in Computer Systems Engineering in 1998 and Ph.D. Electronics and Computer Engineering in 2008 from University of the Algarve, Portugal. He is Adjunct Professor at Instituto Superior de Engenharia, also in the University of the Algarve, where he lectures Computer Science and Computer Vision since 1994. He is member of LARSyS and the ISR in Lisbon. He participates in 13 financed scientific projects, and he is co-author more than 90 scientific publications. His major research interests lies on computer and human vision, assistive technologies and human-computer interaction.

Pedro Cardoso Professor Pedro Cardoso graduated in Mathematics/Computer Science from the University of Coimbra (Portugal) in 1996 , got his M.Sc. in Computational Mathematics from the University of Minho (Portugal) in 1999 and Ph.D. in Discrete Mathematics in 2007 from University of Seville (Spain). He is Adjunct Professor at Instituto Superior de Engenharia of the University of Algarve and a member of LARSyS (ISR - Lisbon). He is author of more than 30 scientific publications. His major research interests lies on operational research, databases, and human-computer interaction.

Jânio Monteiro Professor Jânio Monteiro graduated in Electrical and Computers Engineering in 1995 from University of Porto, Portugal. In 2003 and 2010 he got a M.Sc. and Ph.D. degrees, respectively, in Electrical and Computers Engineering, from the Technical University of Lisbon, Portugal. He is an Adjunct Professor at Instituto Superior de Engenharia in the University of Algarve, where he lectures Telecommunication Networks since 1998. He is member of INESC-ID in Lisbon. He participated in several national and European scientific projects. His major research interests comprise Communication Networks, Video Transmission over IP and Smart Grid Protocols.

Luís Sousa Luís Sousa is a researcher at University of Algarve currently finishing his master degree in Electrical and Electronic Engineering. His major interests lies on electronic systems, embedded systems and computer vision, being PoolLiveAid one of his most recognized projects.

Jorge Eduardo Jorge Eduardo is a researcher at University of Algarve currently finishing his master degree in Electrical and Electronic Engineering. His major scientific interests lies on operational research.

Ricardo Alves is currently finishing his masters degree on Electric and Electronic Engineering, Ricardo Alves is also a researcher for the University of the Algarve working with depth sensors. Ricardo, is one of the author of PoolLiveAid project, also spends some of his times developing other small electronics and programming projects.

Received: