AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Carlos Fernández del Valle

DECLARA ser el titular de los derechos de propiedad intelectual de la obra:

Development of an optimization tool based on Evolutionary Algorithm for the operation of Smart Distribution Grids,

que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.
b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
e) Asignar por defecto a estos trabajos una licencia Creative Commons.
f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente).*

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

a) Que la Universidad identifique claramente su nombre como autor de la misma
b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
c) Solicitar la retirada de la obra del repositorio por causa justificada.
d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que

pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 26 de abril de 2018

ACEPTA

Fdo.

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título "Development of and optimization tool based on Evolutionary Algorithm for the operation of Smart Distribution Grids"

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2017/18 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada

de otros documentos está debidamente referenciada.

Fdo.: Carlos Fernández del Valle Fecha: 26 / 04 / 2018

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Claudia Battistelli Fecha: 26 / 04 / 2018

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER EN INGENIERÍA INDUSTRIAL

DEVELOPMENT OF AN OPTIMIZATION TOOL BASED ON EVOLUTIONARY ALGORITHM FOR THE OPERATION OF SMART DISTRIBUTION GRIDS

Autor: Carlos Fernández del Valle
Directora: Claudia Battistelli

Madrid
Abril 2018

DESARROLLO DE UNA HERRAMIENTA DE OPTIMIZACIÓN BASADA EN ALGORITMOS EVOLUTIVOS PARA LA OPERACIÓN DE REDES DE DISTRIBUCIÓN INTELIGENTES

Autor: Fernández del Valle, Carlos.

Directora: Battistelli, Claudia

Entidad Colaboradora: RWTH Aachen University

RESUMEN DEL PROYECTO

INTRODUCCIÓN

La creciente penetración de energías renovables así como de nuevas tecnologías de información y comunicación (TIC) combinadas con la expansión del mercado eléctrico, están cambiando los sistemas eléctricos hacia las llamadas redes inteligentes o Smart Grids (SGs). Las SGs introducen nuevos conceptos como los Recursos Energéticos Distribuidos (DERs) y la Gestión de la Demanda (DSM). Además, el almacenamiento de energía, con tecnologías tales como las baterías puede jugar aquí un papel fundamental, por ejemplo, asegurando el balance de generación y demanda del sistema.

El análisis de la operación del conjunto de DERs, DSM y baterías es muy relevante para minimizar los costes de operación del sistema. Por ello, se resuelve típicamente un problema de „Despacho Económico“ (ED) mediante optimización. Técnicas como la Programación Lineal (LP) o la Programación Entera Mixta (MIP) han sido usadas frecuentemente en la literatura, sin embargo, cuando estas ténicas no son aplicables por las características del problema, se deben buscar otros métodos. Por esta razón, esta tesis se centra en técnicas modernas de optimización, los llamados Algoritmos Evolutivos (EA), y describe como se desarrolló una herramienta de optimización usando el Algoritmo Genético (GA) y la Optimización por Enjambre de Partículas (PSO).

METODOLOGÍA

En primer lugar, se hizo una intensiva revisión de literatura sobre EAs. En esta revisión, que forma parte del Capítulo 2 de la tesis, se describen más de 20 algoritmos distintos. La revisión trata de familiarizar al lector con los conceptos más básicos de los EAs, así como sus diferentes variantes y aplicaciones al sector eléctrico.

Tras ello, se desarrolló una herramienta de optimización en Matlab usando el Algoritmo Genético (GA) y la Optimización por Enjambre de Partículas (PSO). Aquí, primero se hizo una revisión a fondo de las diferentes variantes de estos algoritmos que fueron posteriormente implementadas en la herramienta. Por ejemplo, la herramienta es capaz de resolver el problema de optimización usando un GA adaptativo o un PSO con peso inercial decreciente (DIPS), entre otros. Con estos algoritmos y sus variantes, se resuelve el ED de una SG que incluye generación eólica, solar, diésel, baterías y respuesta en demanda. En la formulación del problema, la parte más innovativa es la inclusión de un modelo de degradación de la batería que considera el envejecimiento cíclico y calendárico de la batería durante su operación.

Para probar y evaluar la herramienta, se emplearon datos de una microrred real implementada en Graciosa (Azores, Portugal). Este Sistema consta de 4.5MW de turbinas eólicas, 1 MW de paneles fotovoltaicos, 4.7MW de generación diésel y una batería de ion litio de 3.2MWh/6MW. Usando este sistema, se hizo una comparación de los distintos algoritmos de la herramienta. Con el algoritmo de mayor rendimiento se analizaron diferentes escenarios de demanda y penetración eólica. Finalmente, se evaluaron algunas alternativas para mejorar el sistema (ej. Implementación de respuesta en demanda en la isla).

RESULTADOS

El principal resultado (y objetivo) de la presente tesis es la herramienta de optimización desarrollada con los dos EAs ya introducidos. Aparte de eso, lo primero que se hizo fue probar su rendimiento con los datos de una microrred real de las islas Azores. En primer lugar, se definió un escenario base que emplea los perfiles medios de demanda y de renovables. Con ellos se probaron las diferentes variantes de los algoritmos implementados en la herramienta. Los resultados se muestran en la TABLA I.

Algoritmo Genético	**Coste Mínimo**	**Coste Medio**	**Tiempo medio de simulación**
Single-point	2049.957	2671.740	36.362
BLX-0.5	1465.328	1892.409	35.576
SBX	2138.160	2871.489	37.319
Max Min Arithmetical	1832.123	2271.208	67.419
FCB Logical	1728.115	2216.255	72.530
FCB Hamacher	1944.220	2607.198	69.926
PSO			
Constricted PSO	1433.764	1892.409	35.576
DIPS	1697.179	2373.845	34.166
DIPS-TVAC	1236.504	1612.688	38.560
HPSO-TVAC	2042.060	5080.043	34.195

TABLA I Comparación de los diferentes algoritmos y sus variantes en el escenario base

En este problema particular, la variante más eficaz es el PSO con peso inercial decreciente y coeficientes de aceleración variables (DIPS-TVAC), ya que es la variante que obtiene el coste más bajo de todas en un tiempo razonable. Esto, sin embargo, es solo aplicable al problema específico que se analiza aquí. Los EAs dependen mucho del problema a resolver y por tanto otras variantes podrían ser mejores para resolver otro tipo de problemas de optimización.

Tras esta comparación, se analizaron diferentes escenarios de demanda y de penetración eólica usando el algoritmo DIPS-TVAC. Un despacho como el que se muestra en la Figura 1 se presenta para cada uno de los escenarios. Esta figura en concreto muestra el despacho óptimo para el escenario base.

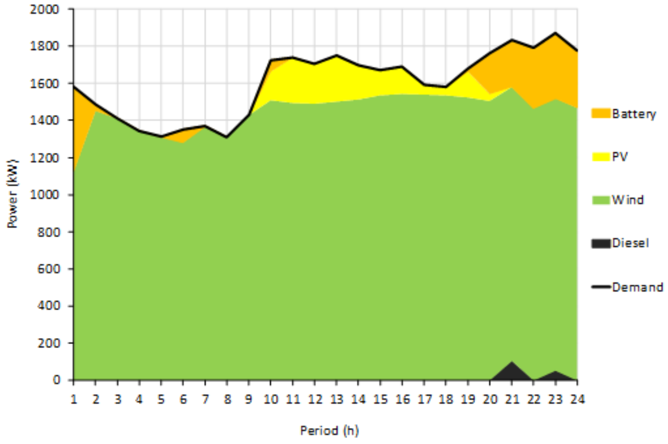


Figura 1 Despacho óptimo para el escenario base

Además de ello, también se analiza el despacho de la batería en cada escenario con las siguientes figuras: la Figura 2 representa el perfil de carga y descarga de la batería en el escenario base mientras que la Figura 3 muestra los recursos que alimentan la batería durante las 24 horas en el escenario base.

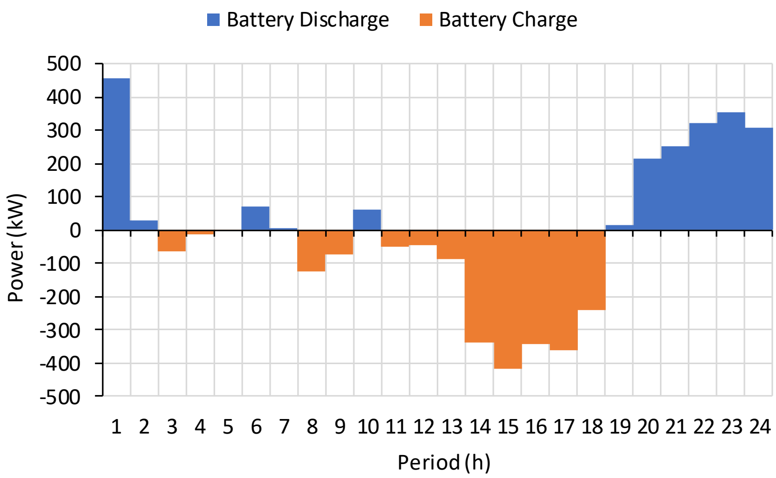


Figura 2 Perfil de carga y descarga de la batería en el escenario base

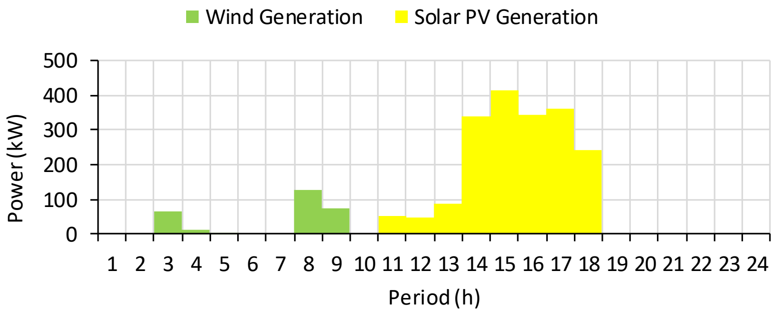


Figura 3 Recursos de los que se alimenta la batería en el escenario base

Analizando estas figuras en cada uno de los escenarios se concluyó que el sistema eléctrico de Graciosa es capaz de operar con un 100% de energía renovable en primavera y otoño si la producción eólica es normal, o en cualquier estación si la producción eólica es alta. Sin embargo, si la demanda es muy alta (verano e invierno) o la penetración eólica es baja entonces la generación diésel sigue necesitándose como respaldo. Además de esto, se muestra que la batería podría ser extremadamente útil en este sistema para absorber la energía renovable sobrante y usarla de nuevo en períodos en los que se pueda evitar la generación diésel.

Adicionalmente, se analizó el efecto de modelar la degradación de la batería en un despacho económico. Para ello, se simuló el escenario base con y sin modelo de degradación de la batería. Los resultados se resumen en la TABLA II.

Escenario Base	ΔDoD Media(%)	SoC Medio(%)	Degradación Total (p.u.)	Coste de Degradación (€)	Coste Total del Sistema (€)
Con Degradación	5.05	54.92	9.12E-5	291.95	1141.72
Sin Degradación	5.20	71.85	1.10E-4	354.53	1180.62

TABLA II Comparación del envejecimiento de la batería en simulaciones con y sin modelo de degradación de la batería

En el modelo de degradación de la batería implementado, el envejecimiento calendárico depende principalmente del estado de carga (SoC). Aunque la temperatura también está incluida en este modelo, ésta se considera constante y por tanto su influencia es despreciable. Por otro lado, el envejecimiento cíclico depende de la variación de la profundidad de descarga (DoD) que se relaciona directamente con el número de ciclos hechos por la batería. En la TABLA II se muestra que en el modelo con degradación el SoC medio así como el ΔDoD medio es menor que en el modelo sin degradación. Esto es debido a que en el primer caso la batería está operando no solo para minimizar los costes de generación del sistema, sino también para minimizar su propia degradación. Esto, si se consideran los costes de reemplazo de la batería en ambos modelos, resulta en mayores costes totales del sistema en el caso de no modelar la degradación, ya que la batería deberá ser reemplazada antes de lo necesario.

Finalmente, se propusieron algunas alternativas para mejorar la eficiencia del sistema. Primero, se simula un caso en el que se implementa un programa de respuesta en demanda y luego se simula otro caso en el que hipotéticamente se doblase el tamaño de la batería.

CONCLUSIONES

Los Algoritmos Evolutivos son muy eficaces resolviendo problemas que incluyen ecuaciones no lineales tales como el modelo de degradación de la batería presentado en este trabajo. En este sentido, los EA superan las características de, por ejemplo, la Programación Entera Mixta. Sin embargo, durante el desarrollo del trabajo he podido observar algunas desventajas de estos algoritmos. En primer lugar, los EAs son algoritmos heurísticos y por ello no es posible saber con certeza si la solución alcanzada es la óptima o no. Además de eso, el hecho de ser algoritmos heurísticos implica que la eficacia de estos depende de parámetros definidos por el usuario. En segundo lugar, los EAs carecen de robustez, queriendo decir que no existe un EA universal que sea capaz de resolver todos los problemas mejor que otro. Por último, los EAs requieren mucho esfuerzo de computación. Por todo ello en muchos casos es posible que otras técnicas de optimización pudieran ser más eficientes que los EAs.

DEVELOPMENT OF AN OPTIMIZATION TOOL BASED ON EVOLUTIONARY ALGORITHM FOR THE OPERATION OF SMART DISTRIBUTION GRIDS

INTRODUCTION

The increasing penetration of renewable energy sources (RESs) alongside new advanced information and communication technologies (ICT), in combination with electricity markets expansion and the appearance of new actors, is reshaping todays' power systems towards the Smart Grid (SG). The SG paradigm introduces some new concepts such as Distributed Energy Resources (DERs) and Demand Side Management (DSM). In the Smart Grid, energy storage (ES) technologies such as battery systems (BS) can play a significant role, ensuring power balance between generation and demand at all times.

It is important to analyze the interplay between DERs, DSM and BS, in order to minimize the operation costs of the system. For this purpose, optimization is used to solve the so-called "economic dispatch" (ED) problem. In the ED problem, optimization modeling techniques such as Linear Programming (LP) or Mixed Integer Programming (MIP) have been widely used in literature. However, when these techniques are inapplicable because of the problem characteristics (e.g. non-linearities or stochasticity), other techniques must be used. This thesis work focuses on a class of these other techniques, the so called Evolutionary Algorithms (EAs) and describe how an optimization tool was developed using the Genetic Algorithm (GA) and the Particle Swarm Optimization (PSO) Algorithm.

METHODOLOGY

First, a deep literature review of EAs was made. In this review, that is part of Chapter 2, more than 20 different algorithms are briefly described. This review familiarizes the reader with the basics of EAs and their different variants and applications to power sector.

Next, an optimization tool was developed in Matlab using the Genetic Algorithm (GA) and the Particle Swarm Optimization Algorithm (PSO). To do so, a deep review of both algorithms was made. Here the different variants of these algorithms are explained. For instance, the tool has the capability to solve the optimization problem using an adaptive GA or a decremental inertia weight PSO (DIPS), among other variants. With these algorithms, the ED problem of a given SG than includes wind, solar PV, diesel generators, demand response and batteries is solved. In the problem formulation, the most innovative part is the inclusion of a battery degradation model that considers both cyclic and calendric ageing of the battery during its operation.

For testing and evaluating the tool, a real microgrid system, implemented in Graciosa (Azores, Portugal), is used. This system is composed by 4.5MW of wind turbines, 1MW of solar PV, a 4.7MW diesel generator and a 3.2MWh/6MW Li-ion battery system. Using this system, a comparison of performance of the algorithms implemented was made. Then, using the best performing algorithm, the current operation of the system is analyzed by simulating different demand and wind scenarios. Finally, some alternatives to improve its performance are assessed (e.g. implement demand response).

RESULTS

The main outcome (and goal) of the thesis is the optimization tool developed using the two abovementioned EAs. As said, to test its performance a real microgrid case study was used. First, a base case scenario was defined. This scenario uses the average demand and average VRE profiles and was used to test the performance of the different algorithms implemented in the tool. The results of this analysis are shown in TABLE I.

Genetic Algorithm	**Minimum Cost**	**Average Cost**	**Average Simulation Time**
Single-point	2049.957	2671.740	36.362
BLX-0.5	1465.328	1892.409	35.576
SBX	2138.160	2871.489	37.319
Max Min Arithmetical	1832.123	2271.208	67.419
FCB Logical	1728.115	2216.255	72.530
FCB Hamacher	1944.220	2607.198	69.926
Particle Swarm Optimization			
Constricted PSO	1433.764	1892.409	35.576
DIPS	1697.179	2373.845	34.166
DIPS-TVAC	1236.504	1612.688	38.560
HPSO-TVAC	2042.060	5080.043	34.195

TABLE I Comparison of performance of different algorithms and their variants for the base case scenario

In this particular case, the best performing algorithm is the Decremental Inertia Weight PSO with Time Varying Acceleration Coefficients (DIPS-TVAC) because it reaches the lowest cost at a reasonable computational time. However, this is only applicable to the specific problem analyzed in this work. EAs are very problem dependent and other variants could perform better when solving other kind of problems.

After the performance analysis, different demand and wind scenarios were analyzed using the DIPS-TVAC variant of the PSO algorithm. A dispatch like the one shown in Figure 1 is presented for every scenario. This figure represents the optimal dispatch given by the tool for the base case scenario.

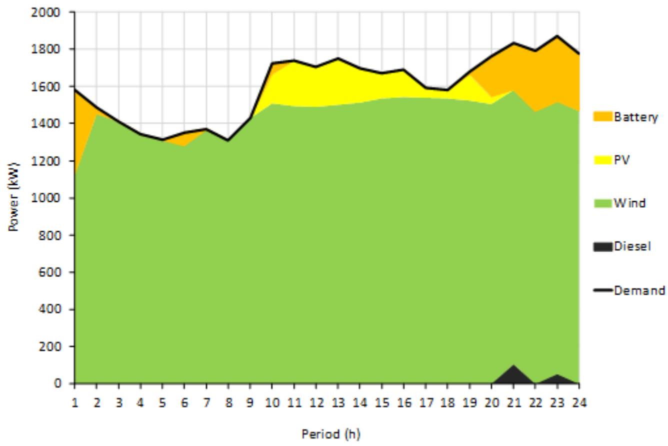


Figure 1 Optimal dispatch for the base case scenario

Also, the battery dispatch is analyzed in every scenario by using the following figures: Figure 2 represents the charging and discharging profile of the battery in the base case scenario, while Figure 3 shows the resources feeding the battery during the 24 hours in the base case.

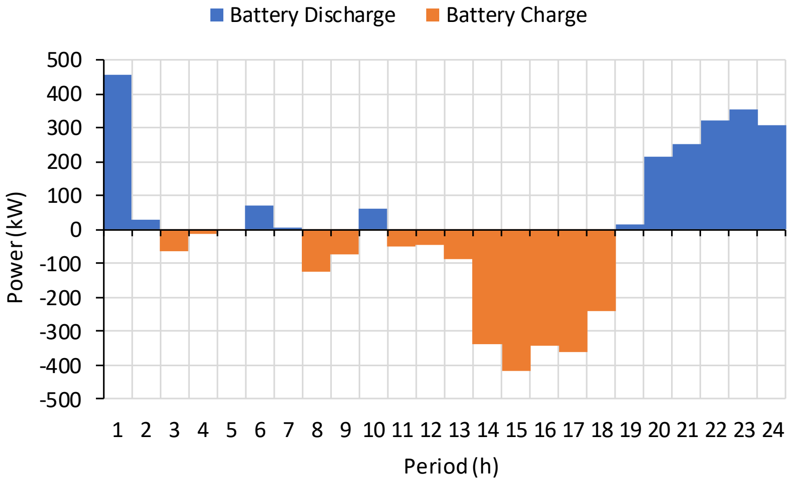


Figure 2 Battery charging and discharging profiles for the base case scenario

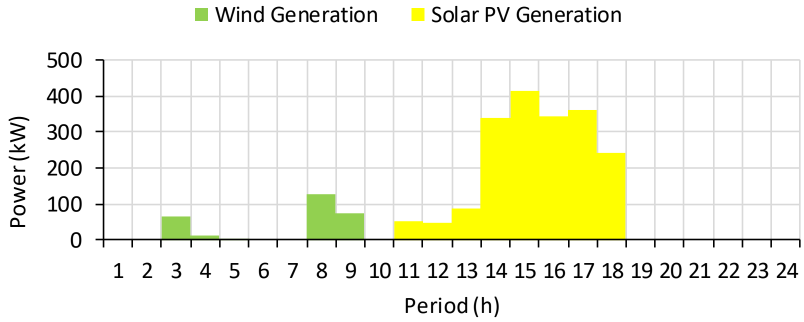


Figure 3 Resources feeding the battery in the base case scenario

By analyzing these figures in each of the scenarios it was concluded that Graciosa's power system is able to operate with 100% of renewable energy in spring and autumn with the average wind profile or in every season if the wind production is high. However, when the demand is very high (summer and winter) or the wind penetration is low, diesel generation is still needed as a backup. Apart from this, it is shown in the thesis that the battery could be very useful in this power system since it absorbs the excessive renewable generation and shifts it to other periods avoiding unnecessary diesel generation.

Additionally, the effect of battery degradation in the model was studied. Here, the base case scenario was simulated with and without degradation and the results were summarized in TABLE II.

Base Case	**Average ΔDoD (%)**	**Average SoC (%)**	**Total Degradation (p.u.)**	**Degradation Cost (€)**	**Total System Costs (€)**
With Degradation	5.05	54.92	9.12E-5	291.95	1141.72
Without Degradation	5.20	71.85	1.10E-4	354.53	1180.62

TABLE II Comparison of battery ageing for simulations with and without degradation model

In the battery degradation model implemented, the calendric ageing is dependent mainly on the State of Charge (SoC). Although the temperature was also included in the degradation model, it was considered constant and its influence is therefore neglected. On the other hand, the cyclic ageing depends on the variation of the Depth of Discharge (DoD) which is directly related with the number of cycles. In TABLE II it is shown that when modeling degradation the average SoC in the simulation as well as the average ΔDoD is lower than in the same case but without modeling degradation. This is because in the first case the battery is operated trying to minimize not only the generation costs but also battery degradation. This, at the end, will result in higher total system costs in the model that does not consider degradation because although generation costs will be lower, degradation costs will be higher and the battery will have to be replacde earlier than expected.

Finally, some alternatives to increase the performance of the system are proposed. First, a case where a demand response program is implemented in Graciosa is simulated and then a hypothetical case in which the size of the battery is doubled was presented.

CONCLUSIONS

Evolutionary Algorithms are very powerful to solve problems that include non-linear equations such as the degradation model presented or the quadratic heat rate function of the diesel generators. In this sense, EAs overcome the capabilities of Mixed Integer Programming. However, I could observe that EAs have some disadvantages. First, EAs are heuristic algorithms and therefore it is not possible to know with certainty if the solution found is the global optimum or not. Also, being heuristic implies that the performance of the algorithm highly depends on the user-defined parameters. Second, EAs lack of robustness, this is, there is not a general EA that will be able to perform better than other in a wide range of problems. Finally, EAs are computationally intensive. For these reasons, it could be possible that other optimization techniques could solve the problem more efficiently than EAs.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Motivation

Electric power systems are experiencing a deep transformation worldwide. Traditionally, electric power systems have had a hierarchical centralized structure which was organized in generation, transmission and distribution, placing customers at the end of the supply chain. This composed a unidirectional structure where electricity was generated by big scale power plants typically owned by utilities, transported by transmission lines and finally delivered to the customers via the distribution network [1]. This traditional grid paradigm has been the predominant one for many years, but the increasing penetration of Renewable Energy Sources (RES), combined with the expansion of markets and the deployment of telecommunication and information technologies in the power sector, has shifted the old grid paradigm towards a new one: the Smart Grid (SG).

A SG could be defined as the collection of all technologies, concepts, topologies and approaches that allow generation, transmission and distribution to be replaced with intelligent, fully integrated environment where the business processes, objectives and needs of all stakeholders are supported by the efficient exchange of data [2]. SGs are a very broad concept that will be covered in Section 2.3. For now, and in a very simple terminology, a SG is an intelligent grid that enables the two-way flow of energy and is typically composed by a set of different loads (industrial, consumers and offices) and what is referred to as distributed generation (DG). On the load side the SG introduces the concept of flexible load which enables Demand Side Management (DSM) or Demand Response (DR) [3]. This is because the two-way energy flow, allowed by the SG, enables consumers to react to price variations fostering the demand of electricity in periods in which is more convenient for them. On the generation side the SG introduces the concept of Distributed Energy Resources (DER) which are smaller power sources that can be aggregated to provide power necessary to meet regular demand [4]. The most common DER are diesel engines, fuel cells, biomass plants, wind turbines, and solar PV panels which are in the generation sider, but they could also be in the demand side. Among these, the last two DER, also known as Variable Renewable Energy (VRE), are exponentially increasing their deployment because of their zero pollutant emissions and zero variable generation costs. The problem of these DER is that they are characterized by intermittency and – consequently – by only partial predictability, causing a threat for the grid stability when their penetration is high, and impacting the SG in different timescales, from the short to the long-term [5].

To cope with this variability, some innovative solutions can be considered to provide flexibility to the SG. Here, as already introduced, flexible loads (DSM or DR) could be enablers of higher shares of VRE, however there are also alternatives in the generation side (e.g. Power-to-Heat, Power-to-Gas or Flexible Thermal Generation). Among these alternatives, Energy Storage Systems (ESS) and specifically (focusing the topic of this thesis) Battery Systems (BS) are acquiring an increasing importance, thanks to to their ability to absorb and inject energy from the gridcounteracting power imbalances, and thanks to their cost reduction potential during the next 20 years [6]. BS can be presented in two different formats: stationary or integrated into an Electric Vehicle (EV). On the stationary side different chemistries are currently under research but Lithium-ion seems to be a competitive technology nowadays due to its high energy density and extended lifetime [7]. Some research has been carried also in other chemistries such as Redox Flow or Sodium-Sulfur [8]. Stationary BS are very advantageous for a SG since they can provide different services to the grid, from energy arbitrage to ancillary services (e.g. primary reserve or black start services) [9]. Within the context of electric vehicles, the prominent battery is Lithium ion in its different variants. In the case of electric vehicles, there is also an impact into the grid and concepts like smart charging or Vehicle-to-Grid (V2G) are arising, however, for now everything is in an experimental phase and not as developed as stationary storage.

The analysis of the interplay between stationary BS, VRE and flexible loads is very relevant for the planning and operation of Smart Grids. This analysis calls for a specific management tool or optimization technique that allows to obtain the optimal dispatch of each technology in order to minimize the cost of operating the SG subject to various constraints that could contain non-convexities, non-linearities or stochasticity adding complexity to the problem. To solve this, classic optimization techniques such as Non-Linear Programing (NLP) or Dynamic Programming (DP) have been used to cope with non-linearities, non-convexities and stochasticity, but have proven to be sometimes inefficient and unable to find the global optimum. NLP is a difficult field and there are only specific case studies that can be solved with this technique and DP, although it is proved to find an optimal solution, using this algorithm is, in most cases, infeasible.

In this context, modern optimization techniques are good candidates to solve the problem in a more efficient way [10]. Among these techniques, Evolutionary Algorithms (EA) are winning positive feedback when trying to solve these problems. EAs are a set of population-based algorithms which root on the concept of natural selection. This is, EAs are based on the fact that populations under certain pressures suffer a process of natural selection that causes a fitness increase of the population [11]. There are many types of EAs, which will be presented in Section 2.1, but in general all the EAs are based on the same underlying idea and understanding the basics of one EA helps to understand the rest. From those, two of the most used EAs in literature are probably Particle Swarm Optimization (PSO), which is based on particles that move across the search space with a given velocity looking for the optimal solution; and Genetic Algorithm (GA) which is based on classical genetic concepts such as selection, crossover or mutation. These algorithms have had a wide applicability in many different problems related to power systems' optimization (e.g. Unit Commitment and Economic Dispatch, Optimal Size and Optimal Reactive Power Dispatch or Optimal Power Flow) as will be shown in Section 2.2.

On the Smart Grids context, PSO and GA have been used to solve the optimal location and size of Distributed Generation [12], Distribution Network Planning [13] or Optimal Scheduling of EVs with V2G [14] or Optimal Scheduling of DER in a SG [15]. The optimization of a Smart Grid's operation considering different generation technologies is still a topic that needs further research. First because the concept of SG is still diffuse and different technologies are continuously being developed and second because EAs although very extended in academia have not been sufficiently tested for its application to industry. This dissertation will try to assess some of these issues by developing an optimization tool based on evolutionary algorithms to solve the economic dispatch problem of a Smart Distribution Grid. To show the capabilities of the tool a real islanded microgrid case study will be used.

1.2 Objectives

This section describes the main objectives of this dissertation. These objectives are not independent, each objective will provide individual results, but the most relevant conclusions are derived from the joint analysis of the full set of objectives.

1.2.1 Literature review of Evolutionary Algorithms

This dissertation makes a comprehensive review of EAs and their possible applications to electric power systems' optimization. This will help the reader understand the basic structure of these algorithms; how new variants can be made and what are their advantages and disadvantages, and the possible application of EAs to diverse optimization problems related to the power sector.

1.2.2 Development of an Optimization tool based on EA

The second objective of this dissertation is, once understood the EAs, investigating more in detail two of them: Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Using these two algorithms, an optimization tool to solve the 24-hour economic dispatch of a Smart Distribution Grid will be developed using Matlab. The tool will explore different variants of both algorithms and a performance analysis will be made to see which variants perform best when solving the specific SG case study.

1.2.3 Test the tool capabilities using a real microgrid case study (Graciosa Island)

Finally, a SG case study represented by a real microgrid will be used to show the capabilities of the developed tools. Specifically, using the best performing algorithms' variants analyzed in the tool development phase, the 100% renewables microgrid recently installed in Graciosa Island (Azores) will be used as a case study. The results presented will be used to analyze the accuracy and performance of the previously developed optimization tool. It will also show an example of the kind of studies that could be made with the optimization tool.

1.3 Outline of the thesis

In Section 2 a review of the relevant state of art for this dissertation is made. Here Section 2.1 will present a review of different evolutionary algorithms, Section 2.2 will show how evolutionary algorithms have been applied to power system's optimizations problems in literature, Section 2.3 will introduce the concept of Smart Grids, while Section 2.4 will introduce the basic concepts of battery energy storage systems and more specifically the lithium ion battery.

In Section 3 the theoretical concepts used in the development of the optimization tool are presented. Section 3.1 present the Genetica Algorithm variants and Section 3.2 the Particle Swarm Optimization theoretical concepts.

Section 4 explains the methodology used to implement all the theoretical concepts into the tool. For this reason, Section 4.1 explains the economic dispatch problem formulation for the Smart Grid, Section 4.2 introduces the degradation model of the battery, Section 4.3 explains the structure of the tool and how it was build and finally, Section 4.4 presents the real case study used for testing.

Finally, Section 5 presents the results from the simulations made with the real case study using different scenarios, while Section 6 draws some conclusions and open a range of possible future developments based on this work.

2 State of the Art

2.1 Evolutionary Algorithms

Evolutionary Algorithms (EAs) [11] are a set of population-based algorithms which root on the concept of natural selection or "survival of the fittest" established by Charles Darwin in the book ***On the Origin of Species*** [16]. EAs state that populations under certain pressures suffer a process of natural selection that causes a fitness increase of the population.

EAs can have several applications among which optimization arises as one of the most relevant ones. In optimization problems the goal is typically to minimize or maximize a particular objective function, which in usually refer to as fitness function. To do so, a set of possible candidate solutions (population of individuals) is first initialized and its performance in the fitness function is evaluated. This population is then evolved via a set of three operators called selection, crossover and mutation. These operations allow the algorithm to obtain new individuals (offspring) that could better fit into the fitness function. The process is then iterated until the optimum of the function is found. An EA must ensure that at the end of the simulation the optimal point reached is the global optimum and not a local one. A schematic representation of a typical evolutionary algorithm is shown in Figure 1.

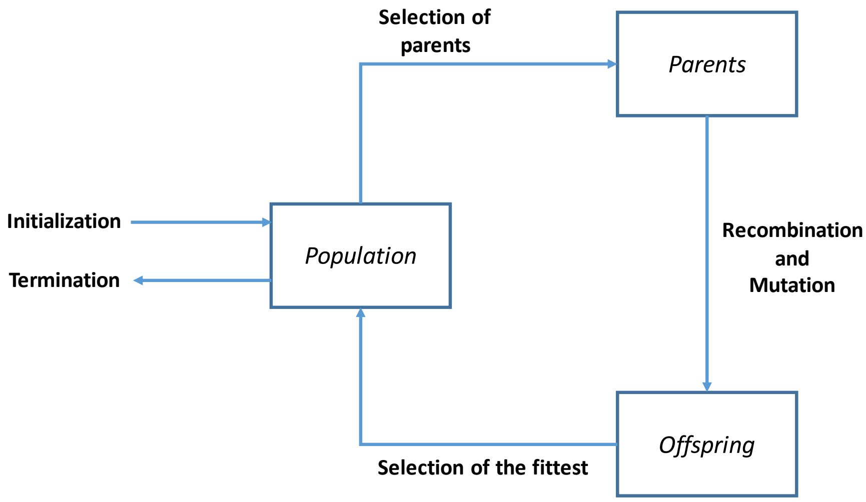


Figure 1 Basic Structure of an Evolutionary Algorithm

Initialization

The initial population of individuals is typically generated randomly. Sometimes a problem-specific heuristic can be used to generate an initial population with a higher fitness value, however it highly depends on the computational effort required.

Selection of parents

At every iteration it is necessary to select the set of individuals that will be used to generate offspring. These individuals are referred to as the parents. Usually they are randomly selected with a given probability. It is a frequent practice to assign individuals probabilities that are proportional to their fitness values. This way, the higher the fitness value the higher the probability of being a parent of the next generation.

Crossover

Crossover or recombination consists on mixing information of two different parents to create offspring. Combination is a stochastic process, this is, the parts of each parents that are combined and the way these are combined is a random decision. EAs do not need to have crossover. For instance, Evolutionary Programming (EP) that will be presented in the next sub-section only uses mutation to generate offspring.

Mutation

Mutation consists on adding random, unbiased information to an individual to generate offspring. Mutation is also a stochastic process. Like crossover, EAs do not necessarily have to present mutation.

Termination

EAs are an iterative process that require the definition of a termination condition that stops the algorithm once it is fulfilled. If this is not defined, the EA will go into an infinite loop. The most straightforward termination condition is, if the optimal value is known, to stop the iteration when the found value is similar to the optimal value with a defined error (ε). However, most of the times the optimal value is unknown and other termination criteria should be used. Some examples could be: maximum number of generations (iterations), maximum CPU time, variation of fitness value between one generation and the following is lower than a defined value…

As already mentioned "Evolutionary Algorithm" is a global term used to define a set of algorithms that have similar structure to the one presented in Figure 1. This brief introduction to EAs will now serve as a basis to explain each algorithm presented in the following subsections. In Section 2.1.1 a review of the four more classic evolutionary algorithms will be made, in Section 2.1.2 some special evolutionary algorithms also commonly found in literature such as swarm-based algorithms will be briefly presented and section 2.1.3 will enumerate some other EAs that can be found in literature but have not been extensively used yet or are less relevant for the purpose of this dissertation. An acknowledgement should be made to the Professor of Cleveland State University Dan Simon since its book ***Evolutionary Optimization Algorithms*** published in 2013 [17] was the main source used for this review of different EAs.

2.1.1 Classic Evolutionary Algorithms

As presented by Back in 1996 [11], there are four classic EAs that have been broadly used in literature. In this section an overview of Genetic Algorithm, Evolutionary Programming, Evolutionary Strategies, and Genetic Programming is made.

2.1.1.1 Genetic Algorithm (GA)

The most prominent example of "evolutionary computation" is the Genetic Algorithm (GA). There are some discussions on who created the first genetic algorithm. In [18] and [19] it is claimed that it was Barricelli in 1954 the one who first created a GA [20]. However, other authors who deeply studied these algorithms claim that it was John Holland the one who first described this algorithm in 1960s and 1970s [21]. Here the author defines GAs as "*Computer programs that "evolve" in ways that resemble natural selection and can solve complex problems that even their creators do not fully understand"*

GA is a method to move from a population of chromosomes (chains of bits or continuous variables that represent a candidate solution) to another one which behaves better under a certain fitness function. To evolve from one population to another the chromosomes suffer a process of selection, crossover and mutation.

GA is composed of different elements whose names are based on classical genetics. These are relevant to understand the algorithm. The main element, as already introduced before, are the chromosomes. Inside a chromosome each bit position is called "locus" and a locus typically has two possible values (0 and 1) which are called "alleles". Additional terminology found in literature is: genome, which is the complete collection of chromosomes taken together, genotype, which refers to the genetic characteristics of the chromosome/gene or phenotype which refers to the physical characteristics if the chromosome/gene. To illustrate this terminology an example similar to the one used in [19] can be used. Imagine that we want to design a household PV+Battery system and we have to decide the optimal size of the PV installation and the battery size and chemistry. We have different options of each technology and each option correspond to a bit string as follows:

1kW PV Panel = 00	Li-ion 3kWh Battery=00
2kW PV Panel = 01	Li-ion 6KWh Battery=01
3kW PV Panel = 10	Lead Acid 3kWh Battery=10
4kW PV Panel = 11	Lead Acid 6kWh Battery=11

Imagine that now we create an individual with the following chromosome:

Individual 1= 1001

We could then say in GA terminology that the chromosome 1001 is composed by two genes (10 and 01). The genotype of its PV panel would be 10 and the genotype of its battery system would be 01. This would represent a phenotype of a 3kW PV panel and a Li-ion 6kWh battery respectively.

From the operators that characterized EAs, GA uses all of them to create offspring:

- Selection: It selects a number of chromosomes of the population for reproduction according to probabilities assigned to each chromosome. These probabilities depend on how well the chromosomes fit to the fitness function. The fittest the chromosome the highest probability to be selected for reproduction
- Crossover: Two chromosomes are randomly picked to exchange bits sequences and create two new offspring. To do so, first the crossover position must be selected in order to exchange information. Suppose that we have selected these two chromosomes for crossover (parents):

Pa = 10000100 Ma=11111111

And we randomly pick the third locus to be the crossover position:

Pa = 100-00100 Ma=111-11111

Considering this we can create two offspring as follows:

Off 1 = 100-11111 Off 2=111-00100

This pair of offspring will be part of what is called the second generation

- Mutation: This operator randomly flips individual bits in the new chromosomes. This usually happens with a very low probability (≈0.001), but it is a relevant operator to avoid local optima and reach the global optimum of the problem. An example could be that offspring 1 (Off 1) mutates the second locus turning into the following chromosome:

  Off 1 = 110-11111

Once the basics of GA are known, a simplified GA pseudocode is shown in Figure 2.

```
Define tuning parameters: crossover and mutation probabilities and population size

Initialize a random population of N chromosomes which represent the different candidate solutions to
the problem

While not (termination criterion)

        Calculate the fitness function F(x) for every chromosome in the population

        Assign probabilities of selection to each chromosome according to the fitness value

        Select the fittest chromosomes for crossover

        Crossover the selection by randomly picking pairs until I have a population of N chromosomes
        again

        Mutate the offspring according to the mutation probability (normally 0.001)

        Replace the current population with the new population

Next generation
```

Figure 2 Genetic Algorithm Pseudocode

Here there is a set of iterations in which new offspring is created. Every iteration is known as a generation. A GA is typically iterated from 50 to 500 generations but depends always on the problem to solve. The whole package of generations is known as a "Run". At the end of each run it is expected to find an optimal solution to the problem. Note that, since randomness plays a role in this algorithm, the results of two different runs might not be exactly the same and therefore it is always recommended to do more than one Run to get a right solution to the problem.

GA is one of the most known evolutionary algorithm worldwide. In this section, just the basics of GA were presented but there are many variations of this algorithm to solve discrete and continuous optimization problems. This allows GA to be applicable to different type of problems beyond optimization such as automatic programming, machine learning or economic models [18].

2.1.1.2 Evolutionary Programming

Proposed by Lawrence Fogel in 1966 [22], Evolutionary Programming (EP) is a classic EA that evolves a population of individuals only by using the mutation operator, this is, there is no crossover. Another relevant feature of EP is that the offspring does not directly substitute the parents. In this case, using a population of size 2N composed by offspring and parents and calculating their fitness values, only the best N individuals are kept for the next generation. EP has been very used in the field of Finite State Machines (FSM) to solve problems like the artificial ant problem developed by Jefferson in 1991 [23].

EP can be applied for discrete optimization (in which case the algorithm would be similar to the one in Figure 2 but only considering mutation) and for continuous optimization. This algorithm deserves further explanation. To begin with, an example of a pseudocode of the EP algorithm is shown in Figure 3.

```
Define fitness function f(x)

Define tuning parameters:

β: Tuning parameter 1 (Normally =1)

γ: Tuning parameter 2 (Normally=0)

Initialize a random population of N chromosomes xi which represent the different candidate solutions to
the problem

While not (termination criterion)

        Calculate the fitness function F(x) for every chromosome in the population

        For each individual xi

                Generate a random vector ri with each element

                x'i = xi + ri * sqrt(β * f(xi) + γ)

        Next individual

        Xi are now the fittest individuals from {xi, xi'}

Next generation
```

Figure 3 Pseudocode of the Evolutionary Programming Algorithm for a Continuous Population

Here the most relevant part is how the mutation is done. To generate the childs x_i', an element (called variance) is added to the original population x_i using parameters β and γ as tuning parameters. Normally β=1 and γ=0, however this is not a necessary condition of the algorithm. As explained by Dan Simon in [17] and by Back in 1996 [11] this algorithm could have some issues:

- F(x) should always be shifted to be positive
- β and γ need to be tuned. If we take the typical values presented above and the domain of x_i is very big the variance will be very small relative to x_i and it will result in a very slow convergence. Also if x_i is very small the variance could be relatively big and start searching for solutions out of the search space
- If β>0 (typical case) and the costs are high then the variance will be approximately constant for all x_i regardless of their cost value, which is not correct

To solve these problem the variance is sometimes also allowed to evolve, creating this way a self-adaptive algorithm. This is known as the meta-EP algorithm where instead of mutating only individuals, child variances would be also mutated as follows:

$$v_i' = \boldsymbol{v_i} + \boldsymbol{r_{vi}} * \sqrt{\boldsymbol{c} * v_i} \tag{2.1}$$

$$v_i' = \boldsymbol{MAX}(v_i', \varepsilon) \tag{2.2}$$

Being ε the maximum value, set by the user, which the variance can take. The meta-EP algorithm usually increases the speed of convergence, but it has been proven that it could also decrease it depending on the problem. Thus, it should be analyzed which algorithm to select to solve every specific problem.

EP has generally been used to solve any type of optimization problems and, as already said, to find optimal finite state machines (FSMs).

2.1.1.3 Evolution Strategies

Evolution Strategies (ES) were first introduced by Rechenberg when he tried to find optimal body shapes in a wind tunnel to minimize air resistance [24] [25]. Similar to GA and EP, ES are population-based algorithms that use selection to create an offspring and select the fittest individuals to become parents for the next generation.

The peculiarity of ES is the notation used for selection and recombination [26]. It is common to find ES noted as $(\mu/\rho \overset{+}{,} \lambda)$ - ES where μ describes the size of the parent population, ρ the number of parent individuals used for recombination (usually omitted) and λ describes the size of the offspring. The selection type is described by the symbol $\overset{+}{,}$. If it is 'plus'-selection (+) then the algorithm is elitist and only the best μ from the entire population of parents and offspring (μ+λ) is selected. On the other hand, if it is 'comma'-selection (,) the parents die after each generation and only the offspring survives. This notation has been expanded sometimes to include the maximum age of individuals (κ), being the final notation as (μ, κ, λ) – ES. Note that if κ=∞ the ES is 'plus'-selection, while if κ=1 the ES is 'comma'-selection.

Another distinctive feature of ES is that they use different crossover operators. The offspring is not only generated by the recombination of two parents as in GAs. These operators are:

- Discrete or dominant recombination: Child elements or variables are picked randomly from different parents, so that all the parents are used to create a single child
- Intermediate recombination: It takes the average value of all parents
- Weighted multi-recombination: It is a generalization of the intermediate recombination, but a weighted average is made depending on the fitness values of the parents. Better parents get higher weights than inferior ones.

The classical recombination operator from GAs could also be used however this is happen rarely in ES.

As for the mutation operators, typically a multivariate normal distribution is used. Based on this, three mutation operators can be distinguished, depending on how the covariance matrix associated to the distribution is: spherical/isotropic, axis-parallel and general. More information can be found in [26].

Finally, it is very relevant in ES to control the parameter of mutation to achieve the right optimal solution. Assuming an isotropic normal distribution with variance (step-size) σ, different strategies for parameter control are explained:

- 1/5th success rule: Discovered by Rechenberg [25]. Here he basically says that if the ratio of total successful mutations is lower than 1/5 then the step-size σ should be decrease, while if it is higher than 1/5 then it should be increased. For this purpose, Schwefel derived a factor c=0.817 by which the step-size should be increased or decreased.
- Self-adaptation: New control parameters are generated through recombination and mutation just as the population. There are different types of self-adaptation such as derandomized self-adaptation or non-local derandomized step-size control (CSA).
- Covariance Matrix Adaptation (CMA): The goal of CMA-ES is to fit as well as possible the distribution of the ES mutations to the objective function. This fit can only succeed for a quadratic function but almost every function can be approximated by a quadratic one. It is a very powerful parameter control strategy however its implementation is complex.

As seen, evolution strategies are very similar to GAs, however they are more oriented to solve problems with continuous variables. Furthermore, while GAs perform mainly crossover, ES are more focused on mutation operators which increase exploration rather than exploitation of the candidate solutions. ES have been applied to solve problems like the Traveling Salesman Problem, Neural Networks, vector optimization or parameter optimization in general [27].

2.1.1.4 Genetic Programming (GP)

The previously explained classic evolutionary algorithms had in common that the form of the candidate solutions was known. This is, it was known if there were real numbers, integer, arrays… However, it might happen that the structure of the candidate solution is not known. Genetic Programming (GP) is a generalization of the GA that tries to search not only the optimal solution of the problem but also the structure of these solution. To do so, GP evolves computer programs. The main idea of developing this type of algorithms was automatic programming, which has been the goal of computer scientist for many decades. The fundamentals of GP can be found in [28], which is the main reference used here to give a brief overview of GP.

As previously introduced, GP evolves computer programs instead of solutions. The question is: How can computer programs be recombined and mutated, like in GAs? Imagine that we have two programs written in Python and we do a crossover of the code lines to create an offspring. With a high probability the result will be an unfeasible new computer program. Then, how can we do this? Koza proposes to use a programming language known as Lisp.

Lisp language

Lisp stands for "List processing" and it is the second oldest computer programming language in the world. It was released only one year after Fortran. Lisp was created as a practical mathematical notation for computer programs. Its structure, in form of s-expression or parenthesis expression, makes it perfect for the crossover and mutation operators that characterize EAs. Some examples of Lisp language are presented here:

$$(+(x\ 2\ 5)(abs\ z)) \rightarrow (2x5) * |z| \tag{2.3}$$
$$(-(\div\ y\ \ 2)\ 3) \rightarrow (\mathrm{y} \div 2) - 3 \tag{2.4}$$
$$\big(if\ (>z\ 4)(set\ x\ y\ )\big) \rightarrow if\ z > 4\ them\ x = y \tag{2.5}$$

The main advantage of Lisp and the reason why it can be applicable to EAs is that these s-expressions can be represented as tree structures in which crossover can be directly applied as shown in Figure 4 where equations (2.3) and (2.4) are recombined:

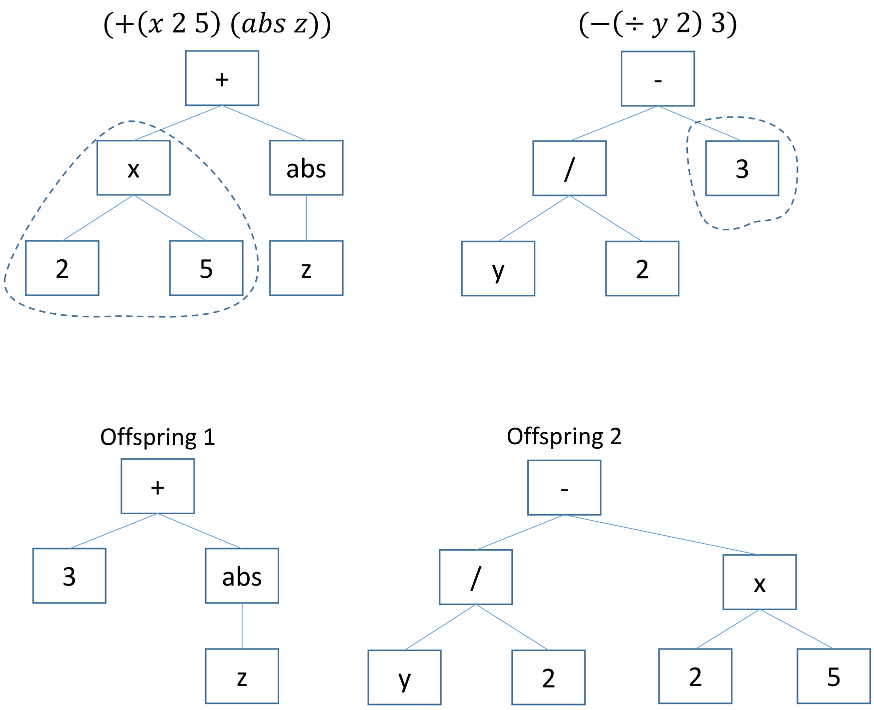


Figure 4 Example of selection and crossover in Genetic Programming using tree structures

The algorithm uses the same logic as the one presented in Figure 2, however GP needs to define some additional tuning parameters which are relevant for the correct working of the algorithm. These are:

- Terminal set: It refers to the set of symbols that can be used to create the syntax tree (variables, constants, random numbers...)
- Function set: It is the set of functions that can appear in the program (mathematical operators, problem-specific functions...)
- D_i: Maximum program size of the initial population. It is a relevant parameter for the initialization
- D_c: Maximum depth of child programs

The rest of the parameters (selection method, population size, mutation probability...) are the same as in GA. The same applies to the termination criterion. As for the fitness function, normally a multiobjective function is defined.

GP is only suitable if the user does not know the structure of the searched solution. If the structure of the solution is known then any other evolutionary algorithm will probably outperform this method.

GPs can have some applicability in artificial intelligence, symbolic regression, hydrology, software developments... but its aplicability to power systems is limited and thus not interesting for the development of this dissertation. More information on GP can be found in [28].

2.1.2 Special Evolutionary Algorithms

In the previous section the four classic EAs were presented, however there are much more algorithms that are categorized as EAs and deserve also some attention. In this section a set of more recent EAs, such as swarm-based algorithms, is described.

2.1.2.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an algorithm based on groups of individuals that work together not only to improve the performance of the group but also to improve the individual performance. This can be seen in animal behavior (e.g. fish swarms, ant colonies…) and in human behavior. PSO has three basic ideas that are essential to understand the algorithm:

- Inertia. We tend to maintain our traditions or habits that have been proven to be successful
- Influence by society. If we see or hear that someone is successful we try to imitate their behavior
- Influence by neighbors. We are more influenced by those who are close to us (e.g. our friend has more influence on us than the society)

PSO, as almost every other evolutionary algorithm, is population-based and therefore it is initiated with a population of N individuals or particles x_i. The peculiarity of PSO with respect the already presented EAs is that these individuals have also an associated velocity v_i at which they move through the search space looking for an optimal solution. Additionally, based on the main ideas explained above, this velocity has an inertia component which tends to maintain the speed and a variable component that produces changes in velocity. This variable component is composed by a cognition learning rate (ϕ_1) which remembers the best position that the particle had in the past and makes it want to return, and a social learning rate (ϕ_2) which contains information about the particle's neighbors and makes it go towards the best performing ones. With these initial ideas a basic pseudocode of the PSO algorithm can be written.

```
Initialize a random population N of individuals xᵢ
Initialize each individual velocity vector vᵢ
Initialize the best so far position of each individual bᵢ
Define the neighborhood size σ<N
Define the maximum influence values ϕ1max and ϕ2max
Define the maximum velocity vmax
While not (termination criterion)
        For each individual xᵢ
                Hᵢ: Nearest neighbors of xᵢ
                hᵢ: Set of neighbors from that minimize the fitness function f(x)
                Generate a random vector ϕ1 and a random vector ϕ2
```

```
            v_i = v_i + φ_1 * (b_i − x_i) + φ_2 * (h_i − x_i)
            If v_i>v_max then
                    V_i=v_max
            End if
            X_i=x_i+v_i
            B_i=Minimum value of f(x_i) and f(b_i)
    Next individual
Next Generation
```

Figure 5 Pseodocode of the Particle Swarm Optimization Algorithm

The most important part of the algorithm is the update of velocity equation where we have an inertial part (v_i), the influences of previous values of the particle ($\phi_1 * (b_i - x_i)$) and the influence of neighbors ($\phi_2 * (h_i - x_i)$). Other relevant features are the size and type of neighbourhoods and the topology of the particle. These will be further explained below.

<u>Types of neighborhood</u>

In PSO there are two types of neighborhoods: 1) global best (*gbest*) where particles are influenced by any member of the swarm and therefore the neighborhood size is N -1, and 2) local best (*lbest*) where particles have only access to the information of their local neighbors according to what is called a swarm topology.

<u>Particle Swarm Toplogies</u>

The arrangement of the neighbors that influence a particle is called topology. The two most common topologies are ring topology where each particle is connected to two neighbors and wheel topology where individuals are isolated from each other and all the information is transmitted to a focal point. Figure 6 shows the different types of swarm topologies that could be used in PSO.

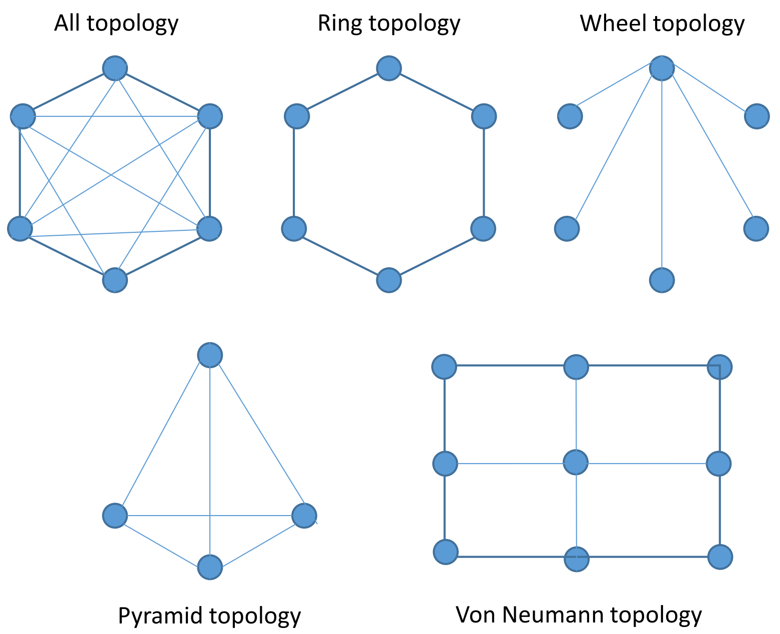


Figure 6 Swarm Topologies

Parameter Selection

1) Maximum velocity

In the algorithm presented in Figure 5 it is shown that the velocity is updated stochastically depending on a series of parameters in every iteration. This velocity is the one at which each particle moves through the search space searching a solution. If this value happens to be very high the particle might have an uncontrolled trajectory causing a lot of oscillation. To limit this effect a maximum velocity is typically defined. This value is selected empirically considering that if the value is too high we might not constrain the oscillation effect that we want to limit and if conversely it is too low, the variations might be so small that optimality is never reached.

2) Selection of constriction factor or inertia weight

Even if the maximum velocity is well-defined it might happen that the particles still diverge and do not reach an optimal solution. To solve this issue, it is a common practice to use a constriction factor or an inertia weight

a) Constriction factor: It improves the convergence of the algorithm by damping the oscillations once the particle is focused on the best point of an optimal region. It can be noted as "K" and it is implemented as follows:

$$v_i = K * [v_i + \phi_1 * (b_i - x_i) + \phi_2 * (h_i - x_i)] \qquad (2.6)$$

b) Inertia Weight: The method is similar to the constriction factor but the parameter (called ω) is now only multiplying the inertia part of the velocity update equation:

$$v_i = \omega * v_i + \phi_1 * (b_i - x_i) + \phi_2 * (h_i - x_i) \qquad (2.7)$$

Here, it is a common practice to use a dynamic inertia weight, starting with high values (≈0.9) to find the neighborhood in which the global optimum is as fast as possible and then decrease it to lower values (≈0.4) narrowing the search. Using this strategy, the algorithm will go progressively from exploration to exploitation and the convergence will be faster.

3) PSO Stability

PSO might have some stability issues if the parameters are not selected in a certain way. Some authors have made some research on what are the optimal parameters in order to avoid stability issues of the algorithm [29][30]. Some common recommendations are:

$$\phi_T = \phi_1 + \phi_2 > 4 \tag{2.8}$$

$$K < \frac{2}{\phi_T - 2 + \sqrt{\phi_T * (\phi_T - 4)}} \tag{2.9}$$

Following these guidelines, the PSO algorithm should be easily tuned for convergence.
PSO is a very powerful optimization tool that has been extensively use for different applications. In this section only a basic PSO algorithm was presented but in literature many variants can be found (e.g. Adaptive PSO, hybrid PSO where it is combined with another EAs, multiobjective PSO…). PSO is also very suitable for its application to power systems and will be a relevant algorithm throughout this dissertation.

2.1.2.2 Simulated Annealing

Simulated Annealing (SA) is an optimization algorithm based on the cooling and crystalizing behavior of the chemical substances. Created by Kirkpatrick in 1983 [31], SA is an individual-based (and not population-based) algorithm and this makes that sometimes SA is not considered an EA.

Before going into the fundamentals of SA, it is relevant to understand the concept of annealing. When a material is heated the internal energy is increased and so the entropy, meaning that there is a higher degree of disorder inside it. When the material is cooled down again, it tends to go to low energy states decreasing entropy and producing crystallization. This process of heating and cooling a material to recrystallize is known as annealing.

SA is based in statistical mechanics that study the behavior of systems with many degrees of freedom in thermal equilibrium at a finite temperature. Since nature is a natural optimizer, systems will tend to low temperature/low energy equilibrium states until it finds the optimum.

In SA an initial “high temperature” candidate solution “s”, which is likely to change to another configuration, is created. Another candidate solution r is randomly generated. If the “energy” in r (E(r)) is lower than the energy in s (E(s)), then the candidate solution is updated to r. To clarify this algorithm a basic pseudocode is shown in Figure 7.

```
Define fitness function f(x)
Initial temperature T > 0
α(T): Cooling Function
Initialized candidate solution x₀ and calculate fitness function f(x₀)
While not (termination criterion)
```

```
        Generate a candidate solution x
        If f(x)<f(x0)
                X0=x
        Else
                r=U(0,1) →Random number distributed between 0 and 1
                If r<exp(f(x0)-f(x)/T) then
                        X0=x
                End if
        End If
        T= α(T)
Next generation
```

Figure 7 Pseudocode of the Simulated Annealing Algorithm

In this algorithm α(T) is usually called cooling function and it is one of the most relevant features of SA, since it controls the rate of convergence. If α(T) is too drastic, then the cooling will happen very fast and a non-optimal state will be reached ,while if α(T) is too gradual then convergence of the algorithm might be too slow. There are many different types of cooling functions such as linear, exponential, inverse linear or dimension dependent. Depending on the problem analyzed one of these function will be used.

SA is one of the oldest evolutionary algorithms (1983) and it has been used in a wide set of optimization problems (constrained optimization, multi-objective optimization...), including both continuous domain and discrete domain.

2.1.2.3 Ant Colony Optimization

Ant Colony Optimization (ACO) is inspired in the foraging behavior of ants [32]. Ants are very simple creatures that can accomplish a lot when working together. When ants search for food they deposit pheromones on their way to make it easier for other ants to find the food source. At first, multiple ways to the food source are created but gradually more ants will use the most efficient way to arrive to the food, depositing more pheromones there. Since pheromones also evaporate the other initial ways will disappear. This pheromone behavior of ants is the main concept of ACO. Some authors do not consider this type of algorithms as evolutionary, however, other authors argue that since it is biologically motivated and population-based, it can be considered an EA.

There are diverse types of ACO algorithms but a simplified and general pseudocode, as the one shown in Figure 8, can provide an overall idea of the algorithm.

```
Set parameters and initialize pheromone trails
While not (termination criterion)
        Construct ant solutions (Iterate per ant, see which point of the solution space is more likely
to take and assign a solution to the ant)
        Apply local search to improve the solution (Optional)
        Update pheromones (evaporation in bad solutions and increase in good solutions)
```

```
Next generation
```

Figure 8 Pseudocode of the Ant Colony Optimization Algorithm

ACO has been usually applied to solve N-P hard problems, telecommunications networks, industrial problems and ultimately to solve dynamic optimizations problems, stochastic optimizations problems, multiobjective optimization or parallel implementations [32].

2.1.2.4 Differential Evolution

Developed by Storn & Price [33], Differential Evolution (DE) is a population-based algorithm that was designed to optimize functions in an n-dimensional continuous domain by performing a parallel direct search. In DE, each individual of the population is an n-dimensional vector that represents a candidate solution to the problem. First, an initial population of vectors is randomly selected by typically using a uniform distribution. Then, to generate a new vector, the algorithm adds the weighted difference between two population vectors to a third vector. This is a mutation process and it can be seen in Figure 9.

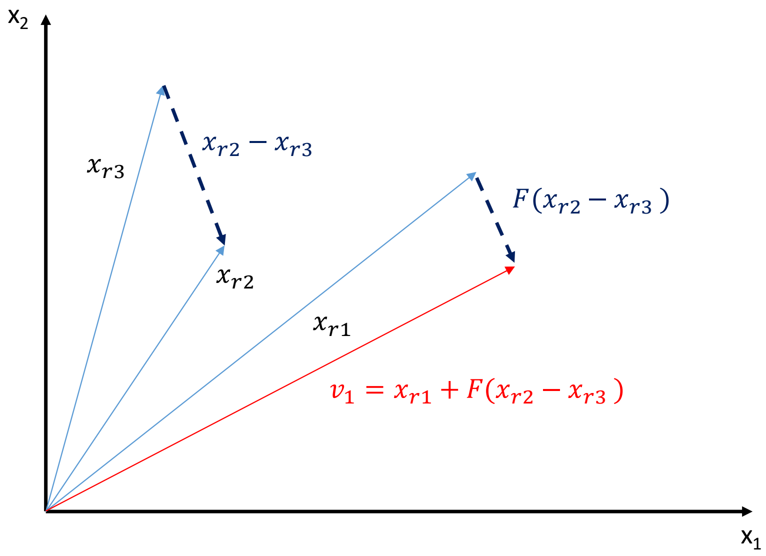


Figure 9 Mutation Process in Differential Evolution

The result of this process is the mutant vector (v_i). This vector suffers a crossover process in which it combines parameters with other individual x_i different from x_{r1}, x_{r2} and x_{r3}. To do so, the positions of the vector are randomly interchanged using a uniform distribution resulting in the trial vector (u_i). After creating the trial vector, the last step is to select the fittest between the initial vector and the trial vector to see who will compose the next generation.

As seen, DE is a very simple algorithm. It only needs three tuning parameters (population size, step size and crossover rate) to run. This, however, is a basic algorithm and some variations can be introduced when creating the mutant or the trial vector or to adjust the step size F. DE has had a wide application in many fields due to its simplicity, fast convergence and possibility of parallelization.

2.1.2.5 Estimation of Distribution Algorithms

Estimation of Distribution Algorithms (EDAs) are population-based algorithms based on probabilistic modelling of promising solutions in combination with the simulation of the induced models to guide their search [34]. EDAs are different from other EAs since there is no mutation or crossover. After initializing a population of individuals, the next generation is created based on an estimated probability distribution from the previous generation. The estimation of this distribution is the most challenging part of this type of algorithms because it requires knowledge on probabilistic graphical models (Bayesian networks or Gaussian networks).

EDAs, as almost every other algorithm can have discrete or continuous domains. Depending on the application it is necessary to decide which one is more suitable. This kind of algorithms are very complex because they require a good knowledge on probabilistic graphical model and going deeper into them is beyond the scope of this dissertation.

2.1.2.6 Biogeography-based Optimization

Biogeography is the study of speciation, extinction and geographical distribution of biological species. This science started with Alfred Wallace and Charles Darwin in the 19th century and it was a pure descriptive science. In 1960s the first mathematical models started to appear. These models tried to describe how species migrate from one island to another, how new species appear and how species die. By island it was meant any habitat that is geographically isolated from other habitats. If an island is friendly to life it will be said that it has a high Habitat Suitability Index (HSI), an index that is calculated based on a set of variables (rainfall, diversity of vegetation, diversity of topographic features…) which are called the Suitability Index Variables (SIV). Normally habitats with high HSI tend to have a higher number of species and a lower immigration rate because the habitat is already saturated. Habitats with low HSI tend to have high immigration rates and therefore will tend to increase their HSI. Figure 10 shows how the immigration and emigration rate relates with the number of species in a habitat.

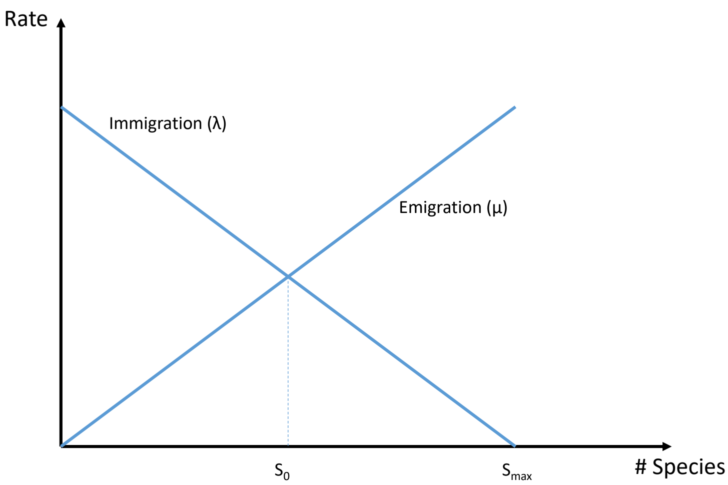


Figure 10 Species model of an habitat

As seen the higher the number of species in an habitat the lower the immigration and the highest the emigration. The equilibrium point S_0 (which is also the optimal point) is reached when the immigration rate equals the emigration rate.

Biogeography-based optimization (BBO) is an algorithm based on the ideas explained above. In this algorithm, islands are considered to be the equivalent to candidate solutions. Thus, an island with high HSI is a good candidate solution and an island with low HSI is a poor solution. In Figure 10 a good island would be in the right side of S_0 and a poor island in the left side. A good island will therefore have a higher emigration rate and the poor one a high immigration rate. In BBO this will represent that good candidate solutions will tend to share their features with poor candidate solutions until reaching an optimal point. The migration rates are then used to share the information probabillistically among individuals. These individuals can also suffer mutation.

BBO has some similarities with other EAs such as PSO or DE. In these three algorithm solutions are maintaned from one iteration to the next (they do not have generation of children like in GA for example). However, BBO is the only one that changes the solution directly via migration from other solutions. This might make this algorihm more suitable to solve some problems in which PSO or DE could have a worse performance. For this reason, BBO has been utilized for different applications such as power system optimization, antenna design, job scheduling problem or parameter estimation in chaotic systems.

2.1.2.7 Cultural Algorithms

All the algorithms presented above were based on the understanding of how natural systems evolve, natural selection and genetics. However, it is frequently suggested that cultural evolution enables societies to adapt to their environment at rates that exceed biological evolution [35]. In order to take advantage of this, Cultural Algorithms (CAs) were developed as a way to improve the performance of EAs. CAs are a type of computational models that try to model the evolution of culture in a society.

The most important feature on CAs are the so-called belief spaces, which are the cultural norms of the virtual society created for the cultural algorithms. Recombination and mutation will be influenced by this belief space. The belief space can be designed by the programmer to impose particular constraints to the problem, to favor preferred features in the population or to avoid undesirable features. A basic scheme of CAs with its belief space is presented in Figure 11.

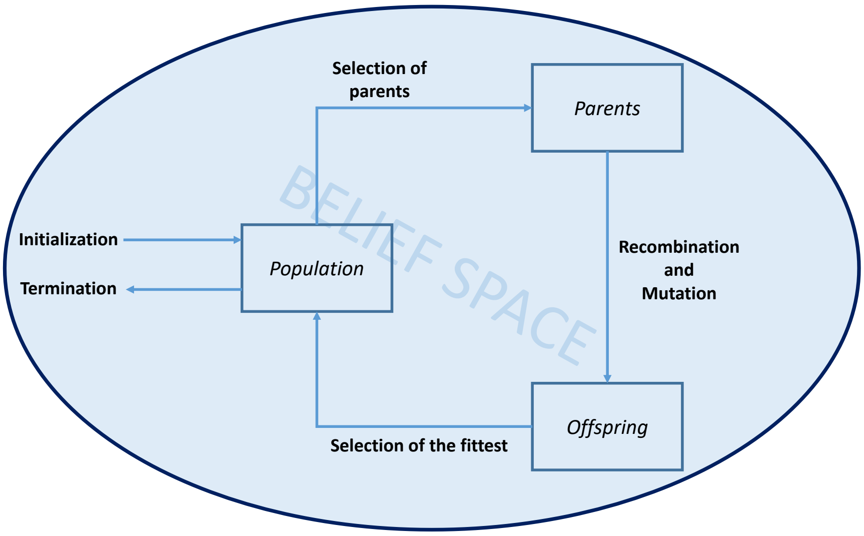


Figure 11 Schematic Representation of a Cultural Algorithm

A CA is then basically any Evolutionary Algorithm which is influenced by a belief space. For instance in [35] a CA based on GA is presented (Version Space Guided Genetic Algorithm). CAs have been applied to different problems such as prediction of neural fuzzy networks, power systems optimization, job scheduling problems or digital filter design.

2.1.2.8 Opposition-based Learning

Opposition-based learning (OBL) was introduced as an attempt to increase the rate of learning of EAs It tries to simulate social revolutions in different algorithms to speed up their convergence. Thus, this is not an EA per se, but something applicable to them. OBL proposes that when randomly generating the initial population of candidate solutions it is possible to look also at the opposite candidates to broaden the search area and increase convergence. There are diverse definitions of what an opposite is. These are presented in Table 2-1 (only for scalars but can be extended to vectors). Here it is assumed that:

$$x \in [a, b] \; where \; a < b \tag{2.10}$$

$$c = \frac{a+b}{2} \; (center \; of \; the \; domain) \tag{2.11}$$

Table 2-1 Type 1 Scalar Opposites

Type 1 Opposites	
Reflected Opposite	$x_{o1} = a + b - x$
Modulo Opposite	$x_{o2} = (x - a + c) * mod(b - a)$
Quasi-opposite	$x_{qo} = rand(c, x_{o1})$
Supper Opposite	$x_{so} = \begin{cases} rand(x_{o1}, b), & x < c \\ rand(a, x_{o1}), & x \geq c \end{cases}$
Quasi-reflected Opposite	$x_{qr} = rand(x, c)$

Once the different methods of doing opposition are known, OBL can be applied to any EA, for instance to GA as in [36]. Opposition-based learning has been proven to be a powerful complement for EAs that can have applicability in many areas such as machine intelligence, image segmentation or power systems optimization.

2.1.2.9 Tabu Search

Introduced by Glover [37], Tabu Search (TS) is not properly a population-based algorithm, however it is still considered as an EA because it is based on the natural world and it performs an iterative search. TS is based on the idea than when a region of the search space has already been visited then that region turns into a tabu and the probability of visiting it again becomes negligible. This can be applied not only to regions of the search space but also to search strategies.

The algorithm defines what is called a „Tabu List (T)“ which includes all the tabu values that the individuals cannot take. When creating children the algorithm does not allow to use features of the tabu list (T). In the first iteration T=0 and thus, every children is considered as valid but as iterations progresively increase the creation of neighbours will be more constrained by the tabu list, since at the end of every iteration new forbidden features are added to T.

TS has been applied to a broad variaety of problems (e.g. artificial neural networks, location routing problem, job scheduling problem, image colouring...). TS has also many applications in power system optimization and therefore it will be relevant for Section 2.2.

2.1.2.10 Artificial Fish Swarm Algorithm

Proposed by Li in 2003 [38], the Artificial Fish Swarm Algorithm (AFSA) is a population-based algorithm inspired in the collective movement of the fish and their various social behaviors such as praying, swarming or following. In this algorithm the artificial fish is the equivalent to a candidate solution that move with a particular behavior trying to find the optimal solution in the search space. These artificial fishes have a visual field within which they can see other fish. The visual field can be described with the following equation, where k are the dimensions of the search space, δ is a tuning parameter and u_k-l_k is the size of the search space in dimension k.

$$v = \delta * \max_k (u_k - l_k) \tag{2.12}$$

Once the visual field is defined, artificial fish can move across the visual field according to five different behaviors: random behavior, chasing behavior, swarming behavior, searching behavior and leaping behavior. Using these behaviors, the fish only moves to the new position if the fitness value is better than the one in the previous position. If this is not the case then the fish ignores the behavior and stay in its current position.

AFSA is a swarm-based algorithm very similar to PSO. Among its different applications the most relevant ones have been optimization of neural networks, job scheduling problems, clustering, PID controller optimization and in a less extent power system optimization.

2.1.2.11 Group Search Optimizer

Group Search Optimizer (GSO), first proposed by He et al. [39], is an EA based on the foraging behavior of land-based animals. In this algorithm the population is called a “group” and the individuals that compose the population are called “members”. Each member has a position in the search space x_i and a head angle θ_i which is used to define it direction. GSO differentiate among three different type of members: producers that search for a better solution, scroungers that follow produces and rangers that walk randomly across the search space.

It is typically assumed that there is only one producer in the population whose objective function value is the best and the rest of the members are scroungers and rangers. The producer will be the one searching for a better solution in the search space by scanning. Scanning can be performed in many ways (physical, visual, chemical…) but GSO typically uses vision since it is the main scanning mechanism in the animal kingdom. The producer will then find the best point and if it is better than its current position, it will move towards it. The scroungers will then follow the producer by using area copying, which means looking at the same search space as the producer, thus allowing the possibility to find a better solution than the producer. If this happens, the scrounger becomes a producer in the next iteration. Finally, rangers randomly search better solutions across the search space and increase the exploration capabilities of the algorithm.

GSO is an algorithm that has been compared in performance with other classic EAs such as PSO, ACO, GA or EP and has performed very well. GSO has several applications such as Artificial Neural Networks training or power systems optimization.

2.1.2.12 Shuffled Frog Leaping Algorithm

Proposed by Eusuff et al [40], the Shuffled Frog Leaping Algorithm (SFLA) is a hybrid between PSO and shuffled complex evolution (SCE), which is based on the idea of allowing sub-populations to evolve independently while periodically allowing interactions between sub-populations.

SFLA starts with a random population of frogs that is partitioned into memeplexes (sub-populations) that evolve differently in order to search the space in different directions. Within each memeplex frogs are influenced by other frogs' ideas. In order to reach the best solution, it is necessary that frogs with better ideas contribute more to the development of new ideas than frogs with poor ideas. Frogs can also change their ideas using information from the best memeplex in the population. After some iterations the memeplexes are randomly shuffled. This process enhances the quality of the ideas after being infected by frogs from different regions of the swamp. This accelerates the searching procedure.

The algorithm is divided in two: A global search and a local search. The global search is in charge of creating all the memeplexes and shuffle them after a number of iterations. The local search is performed inside the global search for each memeplex. Here each individual is updated according to the best individual of the population, the best individual of the memeplex or randomly until the solution is improved.

SFLA is a powerful algorithm that has applicability in different kind of problems (e.g. optimization of water networks, clustering, PID controllers tuning, power systems optimization...).

2.1.2.13 Firefly Algorithm

The Firefly Algorithm (FA), created by Xin-She Yang in 2007 [41] is an EA based on the flashing light that characterizes fireflies' behavior. In the FA three assumptions are made: 1) fireflies are unisex and thus one firefly will be attracted by another no matter the sex, 2) attractiveness is proportional to their brightness and 3) the brightness of a firefly depends on its fitness to the objective function.

Based on this, the fireflies with higher intensity (better performance in fitness function) attract other fireflies to them and increase their light intensity. At the end of the algotihm all the fireflies will have the same light intensity and an optimal solution will be reached. In the FA the distance r is typically the euclidean distance but depending on the problem another type of distance can be defined (e.g. in job scheduling problem this can be defined as the time lag or time interval).

The FA is a swarm-based algorithm that has been applied in different problems such as power systems optimization, stock market price forecasting, vector quantization for image compression or job scheduling problems.

2.1.2.14 Bacterial Foraging Optimization

Bacterial Foraging Optimization Algorithm (BFOA), first proposed by Passino [42] is an algorithm based on the behavior of the Escherichia coli bacteria, commonly known as E.coli. Thus, to understand the algorithm some basic concepts on foraging of real bacteria must be explained. When foraging, bacteria need locomotion, which is achieved by a set of flagella (see Figure 12). When these flagella rotate counterclockwise the bacteria can swim very fast in straight line while when flagella rotate clockwise the bacteria tumble. This is illustrated in Figure 12.

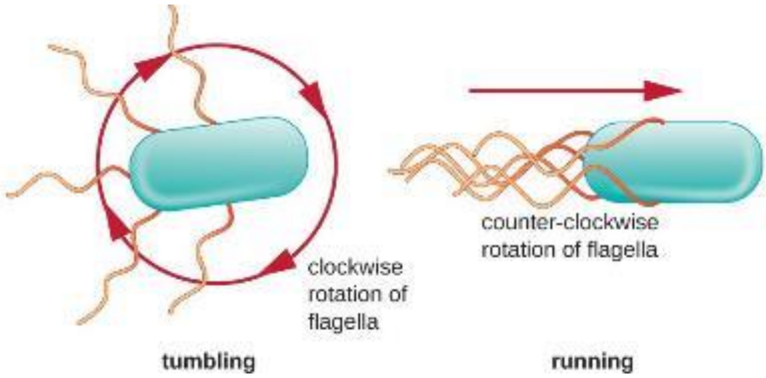


Figure 12 Locomotion of bacteria E.coli

Apart from this, bacteria are characterized by creating exact replicas of themselves when they get sufficient food (reproduction) and by their elimination-dispersal events when they suffer enviromentals changes or attacks.

In BFOA a bacterium is considered as a candidate solution that tries to locate the global optimum. In BFOA bacteria can have four different behaviours: chemotaxis, swarming, reproduction and elimination. Using these behaviors the bacteria moves across the search space and finds the global optimum.

It has been proved that BFOA can compete in performance with similar algorithms such as PSO, ACO or AFSA. As for its possible applications, BFOA have been used to solve different problems such as prediction of stock market indices, power systems optimization or PID tuning.

2.1.2.15 Artificial Bee Colony Algorithm

Artificial Bee Colony (ABC) Algorithm is an optimization algorithm based on the behavior of honey bee swarms. It was first proposed by Karaboga [43]. Here the author divides the bee colony in three different types of bees: employed or forager bees, onlookers and scouts. Employed bees travel back and forth between the food source and the hive. They do this continuously doing local search on their way. Onlooker bees are not associated with any food source. Their task is to wait in the hive and see the behavior of employed bees to then pick a food source. Finally, a scout bee is a bee that searches randomly for nectar. According to this bee classification, the ABC algorithm will consist of three steps: 1) send the employed bees to the food sources with a random possibility to modify their route (local search), 2) select the food sources where the onlookers will go based on the amount of nectar that employed bees are bringing to the hive and 3) send the scout bees to random possible food sources. In the algorithm the food locations will represent the possible solutions to the optimization problem while the amount of nectar in each food source will represent the fitness value of that solution.

The ABC algorithm has been tested together with other swarm algorithms such as GA, PSO or PS-EA concluding that this is an efficient algorithm for multivariable, multimodal function optimizations. ABC has had applicability to different problems (e.g. job scheduling problem, power system optimization, truss structures optimization, training of feed-forward neural networks...)

2.1.2.16 Gravitational Search Algorithm

Proposed by Rashedi et al. in 2009 [44] Gravitational Search Algorithm (GSA) is an EA based on the Newtonian laws of gravity and mass interactions. Therefore, to understand how the algorithm works a brief knowledge of basic gravity laws is required. Gravitation is one of the four fundamental interactions in nature and it is defined as the movement, or a tendency to move, towards a center of gravity, as in the falling of bodies to the earth. Based on this, Newton proposed that every particle in the Earth attracts every other particle with a 'gravitational force'. His first law states that a particle attracts every other particle in the universe using a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between their centers [45].

$$F = G\frac{M_1 M_2}{R^2} \quad (2.13)$$

Where F is the gravitational force, G the gravitational constant M_1 and M_2 the mass of each of the particles and R the distance between them. Newton's second law states that acceleration of a particle depends on this gravitational force and mass:

$$a = \frac{F}{M} \quad (2.14)$$

In GSA individuals are considered as particles whose performance is measured by their masses. These particles attract each other by the gravity force and this force causes particles to move towards heavier particles, which will be optimal solutions in the search space. GSA algorithm is a memory-less algorithm, however in [44] GSA is proven to be an efficient algorithm by comparing it with other classic algorithms with memory like PSO and GA. Some applications of GSA are optimization of power systems or training of feedforward neural networks.

2.1.2.17 Harmony Search

Harmony Search (HS) is an optimization algorithm that tries to mimic the improvisation of music players to solve an optimization problem [46]. Unlike other algorithms presented, HS is based on an artificial phenomenon instead of on a natural phenomenon, which is the process of searching for a better harmony. Harmony can be defined as the combination of sounds that is pleasant from an aesthetic point of view. Musicians seek a best state of harmony determined by aesthetic estimation just as optimization algorithms seek the optimal of the objective function. To reach this state of harmony, musicians improve by doing a lot of practice. In the case of optimization, solutions improve by doing a lot of iterations.

HS makes a new solution considering all the candidate solutions in the population and not only two like in GA. Additionally HS does not require parameter initialization like in other EAs. HS has been applied in optimization of power systems, optimization of the design of steel frames, solving sudokus or optimizations of the design of heat exchangers.

2.1.2.18 Teaching-learning-based optimization

First proposed by Rao et al [47], Teaching-Learning-based Optimization (TLBO) is based on the influences that a teacher has on the students of a classroom. The teacher is considered as a high-learned person that shares his knowledge with the students. The output of the students is measured in grades and it is directly affected by the quality of the teacher.

In order to completely understand this algorithm, suppose that we have a classroom with students whose marks are distributed according to a normal distribution with mean M_1 as shown in Figure 13.

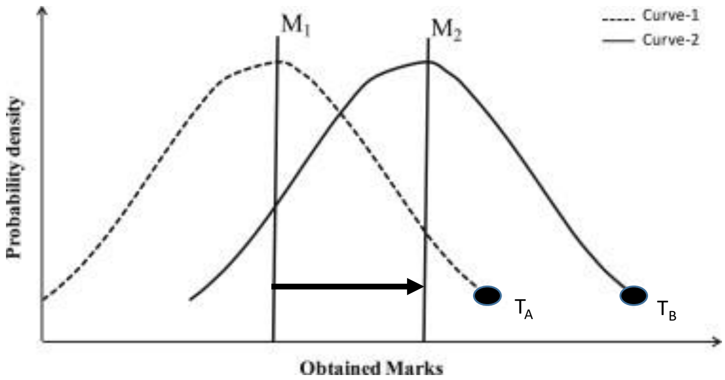


Figure 13 Distribution of marks obtained by a group of learners [47]

That classroom will have a teacher which is the best solution of the problem so far (T_A). The goal of this teacher will be to teach their students as much as possible in order to increase the average grade of the classroom and shift the distribution to M_2, where there will be also a change of teacher to T_B. This learning process, however, not only depends on the best solution (teacher) but also on the interaction between learners. Thus, the knowledge gain will depend on the quality of the teacher and the quality of the students present in class

TLBO has been compared in performance with other EAs in several classical problem producing satisfactory results. As for its applicability, TLBO has been applied to the design of planar steel frames, data clustering or power systems optimization.

2.1.3 Additional Evolutionary Algorithms

Apart from the algorithms described in the previous sections, there are some more algorithms that can be found in literature but have not had as much application as others. These are enumerated in the following list:

- Society and civilization algorithm [48]
- Invasive Weed Optimization [50]
- Intelligent Water Drops [52]
- Gaussian Adaptation [54]
- Imperialist Competitive Algorithm [56]
- Grammatical Evolution [58]
- Chemical Reaction Optimization [60]
- Bat-inspired Algorithm [62]
- Charged System Search [49]
- Cuckoo Search [51]
- Stochastic Diffusion Search [53]
- Big Bang Big Crunch Algorithm [55]
- Squeaky Wheel Optimization [57]
- Glowworm Swarm Optimization [59]
- Krill Herd [61]
- Threshold Accepting [63]

- Great Deluge Algorithm [64]
- Bacterial Chemotaxis Model [65]
- Record to Record Travel [64]
- Grenade Explosion Method [66]

2.2 Applications of EAs to Electric Power Systems

Historically the optimization of power systems has been addressed with a set of classical optimization techniques. All of them try to solve a problem with the following form:

$$minimize\ O.F. \rightarrow\ f_i(x) \qquad (i = 1, 2, \dots, M) \tag{2.15}$$

$$subject\ to \qquad g_j(x) \geq 0 \qquad (j = 1, 2, \dots, N) \tag{2.16}$$

$$h_k(x) \geq 0 \qquad (k = 1, 2, \dots, K) \tag{2.17}$$

Among these methods, in Linear Programming (LP) the objective function is linear, and it is subject only to linear equality and inequality constraints. LP is typically solved using the simplex method but can also be solved with interior point methods. LP has been used to solve many power system optimization problems, such as the security constrained economic dispatch [67], reactive power dispatch [68], optimal power flow [69] or generation expansion planning [70], whenever linear approximation (the so called “linearization”) of the model could be used. When integer variables are considered in the problem, LP can result to be a no longer valid alternative; in this case, Mixed Integer Programming (MIP) must be used. In literature, MIP has been used, for example, to solve the unit commitment problem by representing start-up and shutdown decisions of power plants [71], generation and transmission expansion planning [72] or distribution system planning [73].

Both LP and MIP can be very powerful optimization methods, however sometimes the constraints or the objective function contain non-linearities that these two methods cannot handle. In this context the concept of Non-Linear Programming (NLP) appears. NLP has been used in literature to solve the generation expansion planning [74], optimal power flow [75], capacitor placement [76] or unit commitment [77]. If the objective function is quadratic NLP fails to solve the problem and Quadratic Programming (QP) is needed in this case.

All the above methods deal with determinism, that is to say, everything is assumed as known beforehand, for which no uncertainty is considered in any parameter. There are some parameters that in real life could be uncertain such as the demand profile or the production of VRE such as wind or solar PV. To represent this in an optimization problem it is necessary to deal with stochastic optimization, which introduces the concept of “probability distribution” into the optimization problem. The classical technique to solve these problems is referred to as Dynamic Programming (DP), which has been applied, for instance, to solve the stochastic unit commitment problem [78] or the hydro power scheduling [79], where the so-called Stochastic Dual Dynamic Programming was introduced [80]. These techniques have proven to be powerful and able to reach to an optimal solution; however, according to [81] sometimes it is not feasible to solve the problem with DP or it could require a high amount of computational resources if the complexity of the problem is high (stochastic, non-convex, non-linear…).

In order to overcome these disadvantages and solve complex problems in a reasonable time, an increasing interest on EAs in the power sector has appeared in the last few years. They are categorized as modern optimization techniques because they can sometimes solve complex problems that include non-convexities, non-linearities and stochasticity with a better performance than classic optimization techniques. In this section the goal is to give an overview of the different optimization problems that exist in a power system and which EAs from those presented in Section 2.1 can be used to solve these problems. This is shown in Table 2-2, where some of the most common optimization problem in the electric power sector are presented together with a brief description and a set of references where EAs have been applied to solve these problems.

Table 2-2 Application of Evolutionary Algorithms to Power Sector Optimization problems

Optimization Problem	Description	References
Unit Commitment and Economic Dispatch (UCED)	It consists on dispatching a series of generators in an optimal way given a demand profile and a generation mix	Genetic Algorithm [82] Particle Swarm Optimization [83] Evolutionary Programming [84][85] Ant Colony Optimization [86]
Generation Expansion Planning (GEP)	It is a combinatorial problem that determines the optimal amount and location of new generation in an electric power system over a long-term planning horizon. GEP is basically an expansion of the Economic Dispatch problem presented before in which investment decisions are introduced.	Genetic Algorithm [87] Particle Swarm Optimization [88] Evolutionary Programming [89]
Optimal Power Flow (OPF)	The optimal power flow (OPF) can be defined as "the determination of the complete state of a power system corresponding to the best operation within security constraints" [90], where the best operation typically means the most cost-efficient generation.	Genetic Algorithm [91] Ant Colony Optimization [92] Biogeography-based Optimization [93] Evolutionary Programming [94]
Transmission Expansion Planning (TEP)	The Transmission Expansion Planning (TEP) problem attempts to determine when, where and which type of transmission lines should be installed in a power system to ensure an adequate transmission capacity given a planning horizon and accounting for future installed generation capacity and variations in demand.	Genetic Algorithm [95] Particle Swarm Optimization [96] Simulated Annealing [97] Tabu Search [98]
Reactive Power Optimal Dispatch (RPD)	The reactive power dispatch (RPD) problem is a sub-problem of the OPF calculation, which determines the optimal output of all the devices capable of generate reactive power such as generators, transformers capacitors… by minimizing transmission losses, voltage stability or other objective function	Genetic Algorithm [99] Particle Swarm Optimization [100] Differential Evolution [101] Biogeography-based Optimization [102]
Optimal Placement of FACTS[1] Devices	In the optimal placement of FACTS problem these devices are considered as investment candidates and the objective function could be, for instance, minimize the investment cost, maximize system loadability or minimize voltage deviation.	Genetic Algorithm [104] Particle Swarm Optimization [105] Tabu Search [106] Firefly Algorithm [107]
Disribution Network Reconfiguration	Distribution networks are composed by sectionalizing switches which are normally closed and tie switches that are normally openned. When operating, distribution network feeders are reconfigured by changing the state of the switches from open to close or vicecersa. Since the number of switches is usually high	Genetic Algorithm [108] Particle Swarm Optimization [109] Simulated Annealing [110] Differential Evolution [111]

[1] Flexible AC Transmission System. These are devices that allow to adjust and control electric power systems. Some of their benefits are the improvement of grid stability, active and reactive power flow control, loss minimization and increased grid efficiency [103]

	there are many possible configurations, making necessary to solve an optimization problem to obtain the optimal one. This problem usually tries to minimize losses or to avoid overloading subject to the network constraints	
Optimal Allocation of Distributed Generation	Distributed generation (DG) is electric power generation within distribution networks or on the customer side of the network. [112] This problem strives to obtain the best location and size of DG in a given distribution network with the objective of minimizing power losses, minimizing investment and operation costs or maximizing power quality.	Genetic Algorithm [113] Particle Swarm Optimization [114] Ant Colony Optimization [115] Artificial Bee Colony Algorithm [116]
Automatic Generation Control (AGC)	The main goal of the AGC is to make the Area Control Error (ACE) equal to zero, which means that the frequency deviation as well as the tie line power deviations are zero. The problem to solve here is to optimize the design of the controller (e.g. PID) by selecting the adequate parameters.	Genetic Algorithm [117] Particle Swarm Optimization [118] Bacterial Foraging Optimization [119] Teaching Learning Based Opt. [120]

EAs can have more applications than those showed in Table 2-2, however, these are the most relevant ones found in literature. In this table it can be observed that the two most relevant algorithms that have been applied to all the problems enumerated are the GA and the PSO Algorithm. For this reason, this dissertation will go deeper into those two algorithms to develop an optimization tool and achieve the main goal of this study.

2.3 Smart Grids

In contrast with the traditional scheme, and due to the increasing penetration of VRE, the expansion of markets and the introduction of telecommunications and information technologies into the power sector, a new grid paradigm has been developed: the Smart Grid (SG). SG is a very broad concept and it is not easy to find an accepted definition in literature. A general definition is provided in [2], where the author defines a SG as a collection of all technologies, concepts, topologies, and approaches that allow the silo of hierarchies of generation, transmission and distribution to be replaced with an end-to-end, organically intelligent, fully integrated enviroment where business processes, objectives, and needs of all stakeholders are supported by the effficient exchange of data, services and transactions. Figure 14 shows a schematic representation of a SG.

Figure 14 Smart Grid (courtesy of [121])

The concept of SG affects to different dimensions of the power system, making it necessary to make a division of the SG into subsystems to provide a complete and clear explanation of what a SG is. In this dissertation the division used in [122] was taken as a reference. Here the author subdivides the SG into three different systems: the Smart Infrastructure System, the Smart Management System and the Smart Protection System. In this dissertation the Smart Infrastructure System which includes generation, distribution and transmission infrastructure together with the Smart Management System which includes optimization techniques such as EAs will be the most relevant subsystems. The Smart Protection System, although relevant for a complete understanding of SGs, is beyond the scope of this report and will be only briefly presented.

In this section the term SG will be typically used to refer to a Smart Distribution Grid, which is the most typical type of SG found in literature. This is mainly because the distribution level of the power system is the one that suffers the most significant changes under this new grid paradigm. However, since SGs also apply to the transmission level, a brief introduction to the Smart Transmission Grid will be presented in Section 2.3.4

2.3.1 Smart Infrastructure System

The Smart Infrastructure System refers to the energy infrastructure, information and communications technologies that are used commonly in SGs. This system can be also divided into three different sub-systems: the Smart Energy Subsystem, the Smart Information Subsystem and the Smart Communications Subsystem.

2.3.1.1 Smart Energy Subsystem

The Smart Energy Subsystem includes not only the generation infrastructure but also the transmission and distribution. In the Smart Energy Subsystem, SGs combine the traditional big scale power plants with a smarter and smaller type of generation resources, called Distributed Generation (DG). DG is a set of small power generators, usually refer to as Distributed Energy Resources (DER), which, unlike conventional power plants, are connected to the distribution grid and can send and receive information to the grid, other DER and/or customers via communication technologies. Some examples of DER are small diesel generators, wind turbines or solar PV panels.

When diverse types of DER are connected with a set of different loads (consumers, industrial, office) they compose a Microgrid (MG) which is a small independent grid that could be connected to the system or disconnected if necessary/possible. The implementation and integration of distributed generation to create an MG, however, is not as easy as with conventional generators. This is because, first, DER such as solar PV and wind are characterized by intermittency and partial uncertainty, making the operation of the grid very challenging as their penetration increases and second, DER are typically more costly than conventional power plants discouraging investors and slowing down the deployment of these technologies. To overcome these challenges, in the last few years some innovative technologies and modern grid paradigms have raised. Demand side management (DSM), Power-to-Grid, Energy Storage Systems (ESS) or flexible thermal Generation are now some of the most common solutions proposed in literature. Among these, ESS and particularly Battery Energy Storage Systems

(BESS) are acquiring an increasing interest due to their capability to quickly absorb or inject energy to the grid, and their cost reduction potential expected over the next 20 years [6].

BESS have typically two different applications: stationary and in electromobility. Stationary BESS, where technologies like Lithium-ion or Redox flow batteries are typically used, are very advantageous since they are able to provide different services to the grid, from energy arbitrage to different types of ancillary services. On the other hand, when BESS are used for electromobility, that is to say, inside Electric Vehicles (EVs) their main task is not providing grid services anymore. Although both stationary BESS and EVs will be relevant for the future MGs, this dissertation will only consider stationary BESS.

2.3.1.2 Smart Information Subsystem

The SG does not depend only on the deployment of distributed generation and smart infrastructure but also on the development of monitoring, analysis, optimization and control techniques. For this reason, it is important to include a Smart Information Subsystem. This subsystem is typically divided in metering and measurement devices which study how information is generated in the SG, and information management which covers the analysis, integration and optimization of the measured information. Some example of the Smart Information Subsystem could be smart meters, data modeling software or in the last few years, cloud computing

2.3.1.3 Smart Communications Subsystem

The information retrieved via the Smart Information Subsystem must be somehow transmitted throughout the various levels of the SG. The Smart Communications System serves this purpose. There is an ongoing discussion on which is the optimal infrastructure for the communications system in SGs. The only thing that was recognized is that a communication system must support the quality of service, be reliable, have a high coverage and guarantee security and privacy [122].

To cover these requirements there are wireless technologies such as Wireless Mesh Networks [124], Cellular Communications [125], Cognitive Radio [126], IEEE 802.15 [127], Satellite Communications [128] or Microwave or Free-Space Optical Communications [129]; or wired technologies such as Fiber-Optic Communications [130] and Powerline Communications [131].

2.3.2 Smart Management System

What makes a grid smarter is not only the deployment of innovative infrastructure (hardware), although it is the beginning of a SG. Once the infrastructure has been installed and is working it is necessary to manage it in an optimal way. To do so a set of management objectives and techniques can be used to operate the SG in an optimal and smarter way.

In first place, the SG enables new management goals that were not possible to achieve with the traditional grid. Among these goals, there are three that must be highlighted: 1) Energy efficiency and demand profile improvement, which can be divided in demand profile shaping and energy losses minimization; 2) utility and cost optimization and price stabilization; and 3) emissions control.

To achieve these goals, different techniques have been used in literature. The most widespread technique is optimization, where apart from classic methods such as LP, DP or Stochastic Programming, some modern optimization techniques are now used. In this last group, EAs such as GAs or PSO are being proposed. These algorithms are acquiring an increasing interest due to their capability to solve complex problems efficiently, accurately and with a reasonable computational effort. Apart from optimization techniques, some other popular tools such as machine learning, game theory or auctions have been used for SG management.

2.3.3 Smart Protection System

Finally, a SG must have a Protection System that protects the grid infrastructure from user errors, equipment failures, natural disasters or cyberattacks. In the Smart Protection Systems topics such as reliability of the system, failure prediction and prevention, failure identification and diagnosis or security and privacy on smart metering devices are assessed.

The Smart Protection System is very relevant for the deployment and development of SGs; however, its content is beyond the scope of this dissertation and will not be covered here. More information on protection systems and cybersecurity can be found in [123] and [132] respectively.

2.3.4 Smart Transmission Grid

When using the term "Smart Grids" most authors refer to the Smart Distribution Grid and they frequently leave the transmission grid out of the picture. This typically occurs because this new grid paradigm yields more significant changes in the distribution grid. However, the transmission grid is also expected to become smarter and it constitutes a topic that must be addressed. In [133] it is proposed that the key features of a Smart Transmission Grid are digitalization, flexibility, intelligence, resiliency, sustainability and customization and that it is subdivided into three components: Smart Control Centers, Smart Transmission Network and Smart Substations. This constitutes the scheme presented in Figure 15 where a Smart Transmission Grid structure is shown.

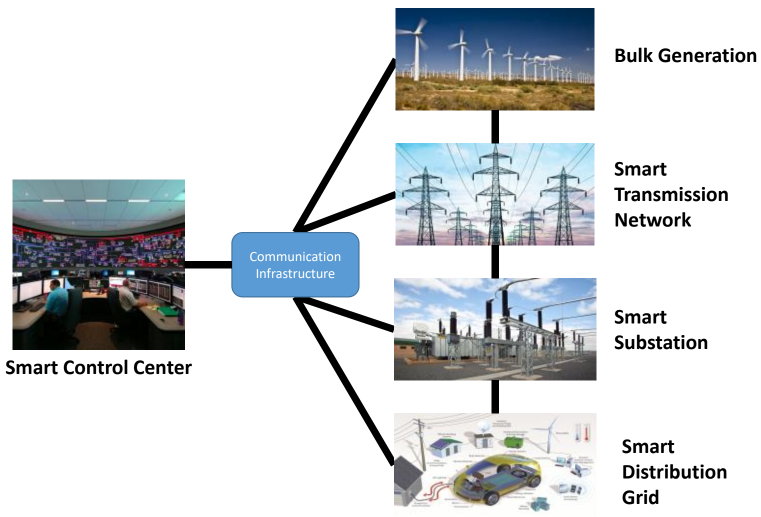


Figure 15 Smart Transmission Grid

Here, Smart Control Centers will be characterized by advanced monitoring or visualization, analytical capability, controllability and advanced interaction with electricity markets. The Smart Transmission Network will be built on the existing transmission infrastructure but new technologies will improve the efficiency, quality and security of existing transmission networks. Advanced FACTS, ultra high voltage lines (UHV), high voltage direct current (HVDC), advance sensing and signal processing are some examples of technologies that will accelerate the transition to a Smart Transmission Grid Finally, Smart Substations are based on classic substations in which reliable and efficient monitoring, operation, control, protection and advance maintenance techniques are enabled. Some of their characterisitcs are: digitalization, autonomy, coordination with other substations or control centers and self-healing characteristics.

The integration of all these subsystems together with bulk generation and Smart Distributions Grids will conform the Smart Transmission Grid.

2.4 Battery Energy Storage Systems

A Battery Energy Storage System (BESS) or battery is a type of Energy Storage System (ESS) based on electrochemical principles such as electrolysis. The first batteries were not rechargeable (first voltaic battery – 1800) but the first rechargeable batteries were invented in 1859, thus, being a relatively mature technology. The most common applications of batteries were in the domestic and industrial sector, however, in the last few years they are being considered for electricity grid support due to their capabilities to quickly absorb and inject energy, their cost reduction potential [6] and, apart from this, due to their modularity that allows them to be packed and reach high capacities. The first pilot projects to integrate batteries in electric power systems were developed in Japan and North America but nowadays a relatively high number of installations can be found. For instance, in Aachen a 5MW/5MWh battery installation was installed to trade energy and exploit the price arbitrage in the market and to provide control reserves for stable grid operation [134]; or in South Australia, where Tesla recently installed a 100MW/129MWh BESS in only 60 days [135].

Batteries are typically classified depending on their chemistry. Each technology is a complete field of study and a huge amount of literature can be found. A summary of the most relevant chemistries and their main characteristics are presented below [136][137][138]:

- Lead-Acid batteries, which are the most mature type of BESS, are characterized by medium-high round trip efficiency (75%-80%), low daily self-discharge (<0.1%) and reasonable cycle life (200-300 cycles at 80% DoD) However, these systems are bulky and heavy (low specific energy) and have low power capabilities. These shortcomings are being overcome by the experimental advanced lead carbon batteries
- Lithium-ion batteries were made popular by electronic devices and, recently, electric vehicles They have numerous advantages such as high energy density, fast charge and discharge capability and high round-trip efficiency (80-90%) that make them ideal for short-term applications. These batteries are still too expensive, but a huge cost reduction is expected by 2030
- Nickel-Cadmium batteries provide good cycle life (>1000 cycles at 80% DoD) and quick charge and discharge capabilities at the expense of low efficiency compared to other technologies (60-70%) and high toxicity. They can operate at extreme temperatures and allow ultra-fast charging with minimal stress

- Sodium-Sulphur batteries can only operate at high temperatures (≈350ºC). Under these conditions, they provide very high energy density, high efficiency (≈80%), nearly zero self-discharge and good cycle life (>3000 cycles). Furthermore, they need minimal maintenance and are 99% recyclable
- Flow batteries, where Vanadium Redox flow is the most common, are a type of rechargeable batteries composed of two chemicals dissolved in liquids contained within the system and separated by a membrane. These batteries are typically characterized by a high efficiency and good cycle life (>10000 cycles), however they have a low energy density since they require a lot of space. They are suitable for large-scale BESS installations.

It must be noted that these are only the most relevant types of batteries but in literature other chemistries could be found (e.g. Sodium-Nickel-Chloride, Zinc-Air or Silver-oxide). In this dissertation we will focus on Lithium-ion batteries which, due to their high energy density and high efficiency are the most promising technology for Smart Distribution Grids. In the following sections, the working principle of Lithium-ion batteries as well as a comparison of different chemistries will be presented. Then the ageing mechanisms of lithium-ion batteries will be explained to finally introduce the importance of considering these effects when operating a battery.

2.4.1 Lithium Batteries: the Lithium-ion battery

Lithium is one of the most attractive material for battery storage systems due to its high potential with respect to hydrogen (-3.05V) and its high specific capacity (3.86 Ah/g). Excluding Lead Acid batteries, which are used in almost every car, lithium batteries are the most used type of batteries worldwide. Their first use was in electronic devices and domestic appliances but nowadays Lithium batteries, due to their high specific energy and efficiency, can be also found in electric vehicles and stationary BESS. The most famous type of Lithium batteries are Lithium-ion batteries, which would be the main type considered in this dissertation, however it is possible to find also Li-Metal batteries (divided in Li-Metal-liquid and Li-Metal-Polymer), which were the first lithium batteries introduced to the market. The problem of Li-metal batteries it that they have some security and lifetime issues that make them not as suitable as Li-ion for grid related applications. In the following sections the working principles, aging mechanisms and diverse types of Li-ion batteries will be presented.

2.4.1.1 Working Principles

A Lithium-ion battery cell, as any other battery cell, is composed by a positive electrode (cathode), a negative electrode (anode) an electrolyte, a separator and a current collector. The different components are shown in Figure 16.

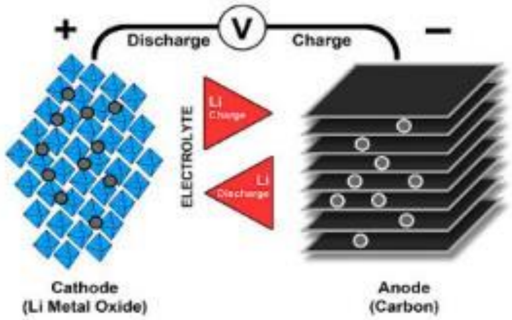


Figure 16 Battery Components (courtesy of [139])

Here the electrodes or active masses are the main components of the battery and the main participants in the reactions, the electrolyte is an aqueous solution or solid material that conducts ions and isolate electrons and does not participate in the reaction, the separator is used to isolate the actives masses and the current collector is where the electricity can be collected. In Li-ion cells the cathode is typically composed by a Lithium-metal-oxide, while the anode is composed typically by graphite (but could be also silicon or titanate) and lithium ions. Depending on the composition of these electrodes the Li-ion battery type will be different, as will be explained in Section 2.4.1.2.

When Li-ion cells are fully charged all the Lithium ions (Li^{+}) are concentrated in the anode in the form of Li_xC_6 (if it is a graphite-based anode). To discharge, this component is decomposed in Li^{+}, C_6 and electrons, which are sent to the current collector to produce electricity. The cell will be completely discharged when there are no lithium ions in the anode anymore and the cathode has all the lithium in form of a Lithium-metal-oxide. The process of charging is the opposite: with the help of a current source that supplies electrons the Lithium-metal-oxide expels Li^{+}that are stored again in the anode.

The charging/discharging process is an electrochemical process that is more complex than this brief explanation and could be an entire topic of discussion. This section is just trying to familiarize the reader with the components of a Li-ion battery and the basic principles and reactions that occur, which will be useful to better understand the following sections.

2.4.1.2 Classification

As already introduced in Section 2.4.1.1, depending on the composition of the cathode and sometimes of the anode, there are several types of Lithium-ion batteries [140]. In order to compare and present them it is a common practice to use six characteristics: (1) specific energy, which is the amount of energy per mass unit, (2) specific power, which relates to the power per mass unit, (3) safety (e.g. if the battery is likely to suffer thermal runaway), (4) performance, (5) life span, which could be divided in cyclic and calendric lifetime and (6) cost. In this section the most common Li-ion chemistries are presented based on these parameters and suggesting some applications to finalize with a comparison of them using hexagonal spider graphics in Figure 17.

- Lithium Cobalt Oxide ($LiCoO_2$). Composed by a cobalt oxide cathode and a graphite anode this chemistry is characterized by its high specific energy making it suitable for small devices. However, these batteries have a low specific power and a relatively short life span being nowadays not the first choice in the industry. Lithium Cobalt Oxide is not the most suitable chemistry for grid applications and, thus, they will not be considered in this dissertation.

- Lithium Manganese Oxide ($LiMn_2O_4$ or LMO). These batteries have manganese oxide as a cathode material. They are characterized by having a relatively good specific energy and power at expenses of cycle life and performance. Among their used we can find from small domestic appliances to electric vehicles and they could be suitable for grid applications.
- Lithium Nickel Manganese Cobalt Oxide ($LiNiMnCoO_2$ or NMC). Being nowadays one of the most used chemistries in the industry, NMC batteries combine Nickel, Manganese and Cobalt to obtain an excellent specific energy and a relatively good specific power, lifetime, safety and performance. These batteries are used in electric bikes, electric vehicles and stationary BESS (e.g. Tesla Powerwall).
- Lithium Iron Phosphate ($LiFePO_4$). The main characteristic of Lithium Iron Phosphate is their safety since they avoid the thermal runaway that could be a problem for other chemistries. Apart from this they are characterize by a high specific power and a high life span, however, they have a low specific energy compared with other chemistries. They are used for instance to replace the lead acid batteries in car.
- Lithium Nickel Cobalt Aluminum Oxide ($LiNiCoAlO_2$ or NCA). This chemistry is very similar to NMC being characterized mainly by a very high specific energy, however, NCA is more stable than NMC because of the addition of Aluminum but they are also more expensive. Among their used NCA batteries can be found in electric vehicles (e.g. Tesla EVs) and could be suitable for grid applications.
- Lithium Titanate ($Li_4Ti_5O_{12}$ or LTO). LTO batteries are the only type of Li-ion batteries that are named after their anode. Instead of using graphite they use Li-titanate. As for the cathode it could be NMC or LMO. These batteries are excellent in performance, life span and safety however they have a very low specific energy and are very expensive. Among their used we can find UPS or solar-powered street lightning. They are probably not the most convenient chemistry for grid applications.

To sum up a comparison of these 6 chemistries is shown in Figure 17.

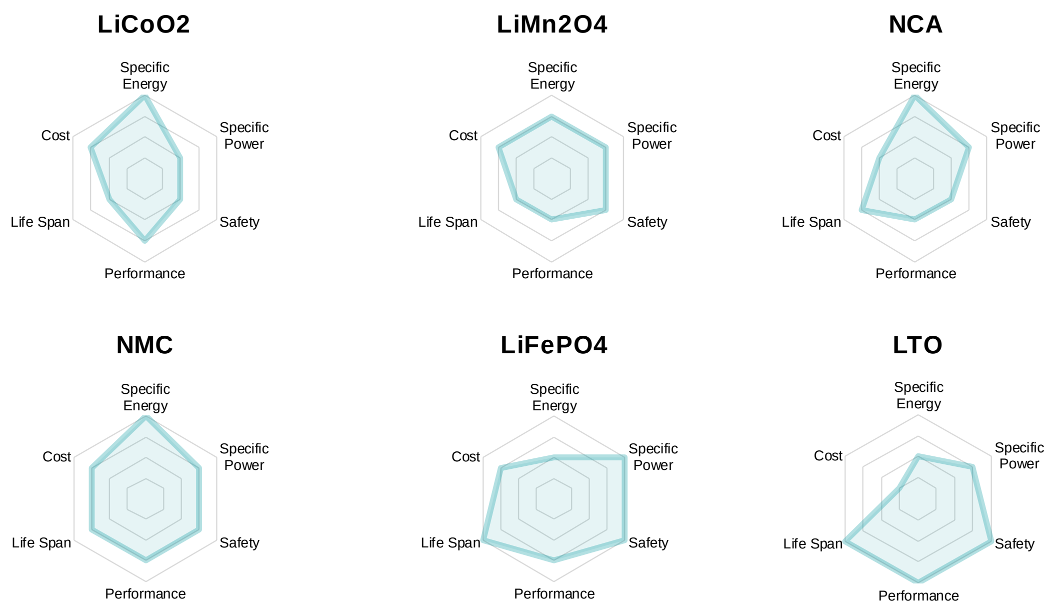


Figure 17 Classification of Li-ion batteries and their characteristics

In this dissertation only the chemistries that, because of their characteristics, could be applicable to the proposed case study will be considered.

2.4.1.3 Ageing Mechanisms

An important feature to consider in the study of batteries is that they are affected by ageing mechanisms that could cause, for instance, capacity fade affecting the capabilities of the battery and increasing its degradation. Identifying these ageing mechanisms is a complex task since they depend on many environmental and internal factors of the battery that interact with each other to generate different effects. Every specific battery type is affected by different ageing mechanisms that could be either mechanical or chemical and therefore, a general explanation of ageing mechanisms is not possible. For the case of interest in this dissertation (Li-ion batteries) some authors have gathered virtually all the possible ageing mechanisms that Li-ion batteries could suffer[141][142]. These are typically divided between anode ageing mechanisms and cathode ageing mechanisms.

Anode Ageing Mechanisms

The negative electrode of Li-ion batteries could be composed by graphite, carbon, titanate or silicon. The most common anode material among these four is graphite and therefore it is the material usually found literature to explain anode ageing. Anode ageing starts with the first charging process with the formation of the Solid Electrolyte Interface (SEI). Since the anode operates in a voltage range outside the stability window of the electrolyte components, a set of reduction reactions are produced in the electrode/electrolyte interface producing the loss of lithium ions and decomposition of the electrolyte. The products of this decomposition process create the SEI, which is intended to protect the battery, however, the SEI is also one of the main reasons of anode's ageing. The loss of lithium ions and the electrode decomposition occurs every cycle producing degradation. Apart from this, the SEI is permeable to Lithium ions and it is partially permeable to charged elements and neutral elements (solvents). When solvents penetrate the SEI they interact with graphite producing exfoliation and creating a gas that cracks the SEI and allows its expansion producing a loss of lithium and active material. Another ageing mechanism, also related with the SEI, is the introduction of SEI materials inside the pores of the electrode increasing the internal impedance and reducing its active surface.

It is also important to discuss the impact of thermal effects on the SEI. It has been observed that at elevated temperatures there is an increase of kinetics of lithium insertion/removal process, a change of composition of the SEI and in extreme cases, the risk of thermal runaway causing fire or explosion. On the other hand, at low temperatures there is a decrease of the diffusion of Lithium ions within the SEI and the graphite that causes a set of parasitic side reactions during charging that produce lithium plating and dendrites growth. Finally, there might be also ageing due to changes in the active material (e.g. structural changes due to mechanical stress) or changes of the composite electrode (e.g. contact loss of battery components with each other or current collector corrosion). The possible interaction of cathode materials with the anode (e.g. transition metal dissolutions) must also be considered as an ageing mechanism.

These mechanisms all together produce an increase of the impedance that leads to power fade together with a loss of lithium ions and a loss of active material which lead to capacity fade.

Furthermore, ageing mechanisms will be influenced by the storage conditions (temperature and state of charge) the depth of discharge and the cycling frequency, which will determine respectively the calendric and cyclic ageing. A summary of all the mentioned ageing mechanisms is shown in and.

Table 2-3 Lithium-ion anode ageing (courtesy of [141] with added calendric/cyclic column)

Cause	Effect	Leads to	Enhanced by	Cyclic/Calendric
Electrolyte decomposition	Loss of Lithium Impedance rise	Capacity Fade Power Fade	High T High SoC	Calendric
Solvent co-intercalation, gas evolution and cracking	Loss of graphite Loss of Lithium	Capacity Fade	Overcharge	Cyclic
Loss of active surface because of SEI growth	Impedance rise	Power Fade	High T High SoC	Calendric
Changes in porosity due to SEI formation and growth	Impedance rise Overpotentials	Power Fade	High cycling rate High SoC	Calendric Cyclic
Contact loss of active material	Loss of active material	Capacity Fade	High cycling rate High DoD	Cyclic
Decomposition of binder	Loss of Lithium Loss of mechanical stability	Capacity Fade	High SoC High T	Calendric
Current collector corrosion	Overpotentials Impedance rise	Power Fade	Overdischarge Low SoC	Cyclic Calendric
Lithium plating and dendrites growing	Loss of Lithium (Loss of electrolyte)	Capacity Fade (Power Fade)	Low T High cycling rates Poor cell balance Geometric misfits	Cyclic Calendric

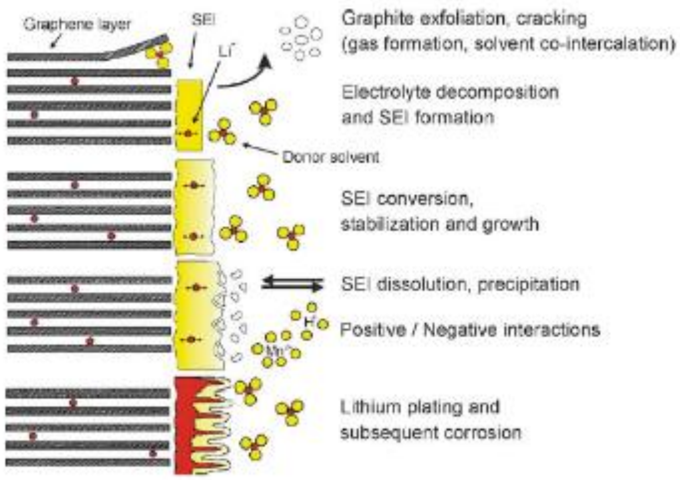


Figure 18 Ageing mechanisms in the anode (courtesy of [141])

Cathode Ageing Mechanisms

The cathode can also affect the cyclic and calendric life of the battery. There are several types of cathode's materials and specific studies should be carried out for each of them. In [141] the authors present a general description of cathode's ageing mechanisms, which will be the one presented here.

First, some changes that can be produced in the cathode and affect battery life are: ageing of the active material, degradation of conducting agents, corrosion of current collector, oxidation of the electrolyte and interaction of ageing products with the negative electrode. These changes, as in the anode, are influenced by cycling and storage conditions and are typically originated by structural changes during cycling, chemical decomposition/dissolution reactions or surface film modification. In Figure 19 a summary of some of the ageing mechanisms that could be found in the cathode are shown:

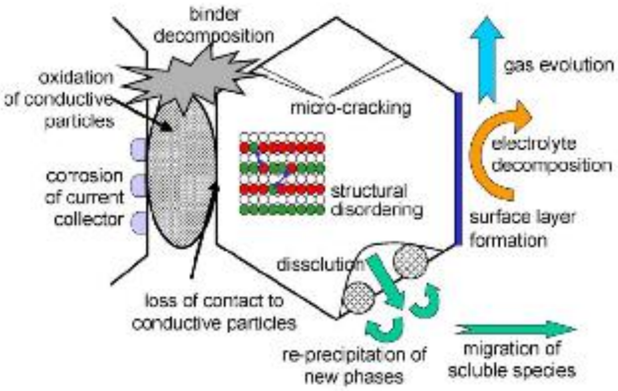


Figure 19 Summary of ageing mechanisms of cathode materials (courtesy of [141])

In this Figure, binder decomposition, oxidation of the conductive agent and corrosion of the current collector produce a loss of contact of the cathode components leading to an impedance increase. This impedance increase could be also originated by the electrolyte decomposition and the metal dissolution. As presented before and impedance increase will result in power fading. On the other hand, structural disordering of the lithium metal oxide and phase transitions produce capacity fading. The ageing of the cathode highly depends on the chemistry used and depending on the chemistry selected the ageing mechanisms should be analyzed. In this case, a Lithium nickel cobalt oxide cathode was used to present an overview of cathode ageing mechanisms. It will have thing in common with other chemistries but will never be the same.

Calendric vs. Cyclic Ageing

In its simplest representation battery ageing can be divided into two different types: calendric ageing and cyclic ageing, which are the most common terms used in literature.

Calendric ageing refers to the proportion of lost capacity during storage [142], this is, the degradation caused when the battery is idle. This type of ageing is mainly influenced by the storage conditions which are the temperature and the SoC.

Cyclic ageing occurs when the battery is either charging or discharging. The typical factor used to account for cyclic ageing is the variation of the state of charge (ΔSoC), being also relevant the operation voltage, current peak and the cycling rate. When there is cyclic ageing the calendric ageing does not stop, therefore, both must be account for at the same time.

3 Theory

This section presents a review of al the theoretical concepts that were used in the development of the optimization tool. The basic theory presented in Section 2 will be here complemented with more detailed concepts on Evolutionary Algorithms and power systems modeling. The first two subsection will explain all the theoretical concepts on Genetic Algorithms and Particle Swarm Optimization needed to understand the tool capabilities, while the last section presents the economic dispatch problem formulation that will be used to solve the optimal operation of a Smart Grid.

3.1 Genetic Algorithm in the tool

3.1.1 Basic Information

The basic information about Genetic Algorithm and the pseudocode of the algorithm can be found in Section 2.1.1.1.

3.1.2 Selection Types

The first operation of the GA is selection. At every iteration it is necessary to select the set of individuals that will be used to generate offspring. These individuals are referred to as the parents. Usually they are randomly selected with a given probability, but it is a common practice to assign individuals probabilities that are proportional to their fitness values. This way, the higher the fitness value the higher the probability of being a parent of the next generation. In literature there are many types of selection, however in the EA-Tool only 4 were implemented.

3.1.2.1 Roulette Wheel Selection

In roulette wheel selection the parents are selected randomly taking into account the fitness values by means of probabilities of selection. This way, the fittest the individual the higher the probability of selection. A very good analogy to understand this type of selection is the casino roulette shown in Figure 20. Here the roulette is rotated to select one parent. Those individuals with a better fitness value have a higher share of the roulette and therefore higher probability of being selected.

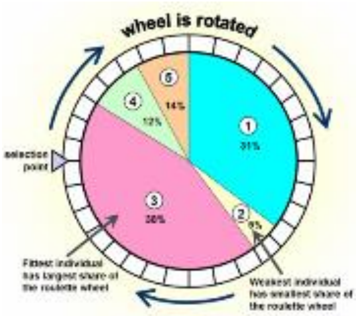


Figure 20 Roulette Wheel Selection (courtesy of [143])

3.1.2.2 Stochastic Universal Sampling

In roulette wheel selection there is the possibility that the fittest individuals are never selected, which might be unacceptable. For this reason, Stochastic Universal Sampling was proposed [144]. In Stochastic Universal Sampling the logic is similar to roulette wheel selection, however, in this case some equally-spaced arrows are added to the "casino" roulette and instead of picking only one individual in each roll, we select as much parents as arrows we have. This way, we ensure that we are at least selecting the fittest individual of the population once. An example of Stochastic Universal Sampling is shown in Figure 21.

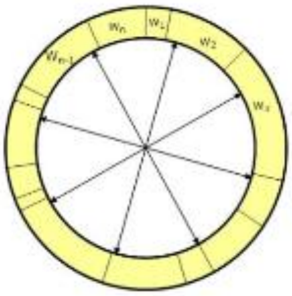


Figure 21 Stochastic Universal Sampling (courtesy of [145])

3.1.2.3 Over-selection

Over-selection consists on assigning a higher probability of selection to the fittest individuals of the population and a lower probability of selection to the worst performing ones fostering, this way, the selection of the fittest. This type of selection was proposed by Koza in [28] where he proposed that there should be an 80% of probability of selecting the best 32% of the population and a 20% of probability of selecting the worst 68%. In this dissertation the recommendation made by Koza was used.

3.1.2.4 Tournament selection

In tournament selection [146], τ (tournament size) individuals of the population are chosen taking into consideration that τ must be greater or equal than two and that there cannot be duplicates in the selection. The chosen individuals "compete" with each other and the one with the highest fitness value is selected for crossover. Tournament selection typically performs better than roulette wheel selection and will be therefore the one used for the case study.

3.1.3 Crossover Type

Once the parents have been selected, the first operation that chromosomes suffer is crossover or recombination. This consists on randomly mixing information of two different parents to generate offspring. As in selection, there are several types of crossover. The EA-Tool is able to perform 9 different types of crossover.

3.1.3.1 Single-point Crossover

Explained in Section 2.1.1.1

3.1.3.2 Two-point or multiple crossover

It is the same as single-point crossover but instead of selecting one recombination point we select two or more points. In the EA-Tool there is up to two-point crossover, however three-point or multiple-point crossover could be easily implemented. This type of crossover, including single-point is broadly used in binary optimization but it does not perform well in continuous GA.

3.1.3.3 Uniform crossover

In uniform crossover [147], we have two parents (x_1 and x_2) and we want to create a child (y_1) What uniform crossover does is to pick randomly, using a uniform distribution, whether the feature $y_1(i)$ of child y_1 comes from x_1 or from x_2. Since we are using two parents and the distribution is uniform parent has always a 50% of probabilities of being selected.

3.1.3.4 Arithmetic crossover

In arithmetic crossover [148] every parent contributes to each of the features of their children, via the parameter alpha (α). This way, for each feature, this type of crossover calculates the following:

$$y_1(i) = (1-\alpha) * x_1(i) + \alpha * x_2(i) \tag{3.1}$$

$$y_2(i) = \alpha * x_1(i) + (1-\alpha) * x_2(i) \tag{3.2}$$

where α can be any value between 0 and 1

3.1.3.5 BLX-alpha or blended crossover

In blended crossover [149] the parents are combined as follows to create offspring:

$$x_{Max}(i) = \max\left(x_1(i), x_2(i)\right) \tag{3.3}$$

$$x_{Min}(i) = \min(x_1(i), x_2(i)) \tag{3.4}$$

$$\Delta x(i) = x_{Max}(i) - x_{Min}(i) \tag{3.5}$$

$$y_1(i) = U[x_{Min}(i) - \alpha * \Delta x(i), x_{Max}(i) + \alpha * \Delta x(i)] \tag{3.6}$$

where α can be greater, equal or lower than 0. A recommended value is $\alpha = 0.5$. . If α is equal to 0 then we have uniform crossover. Some authors have tested the BLX- α crossover using different functions and they have proposed $\alpha = 0.5$ as a recommended value [150]. The BLX- α crossover operator has an interesting property: the location of the child solution depends on the difference in parents' solutions. Therefore, if the difference between parents is small the difference between child and parents is also small. This is important for self-adaptation of the GA algorithm [151].

3.1.3.6 Linear crossover

In linear crossover [152] instead of two children, three children are generated, and we maintain only the two fittest ones for the next generation. The equations used in this type of crossover are:

$$y_1(i) = \frac{1}{2} * x_1(i) + \frac{1}{2} * x_2(i) \tag{3.7}$$

$$y_2(i) = \frac{3}{2} * x_1(i) - \frac{1}{2} * x_2(i) \tag{3.8}$$

$$y_3(i) = -\frac{1}{2} * x_1(i) + \frac{2}{3} * x_2(i) \tag{3.9}$$

3.1.3.7 Simulated Binary Crossover

Simulated Binary Crossover (SBX) [153] simulates the working principle of single-point crossover in the discrete GA. In this type of crossover, the offspring is generated using the following equations:

$$y_1(i) = \frac{1}{2} * [(1 - \beta_i) * x_1(i) + (1 + \beta_i) * x_2(i)] \tag{3.10}$$

$$y_2(i) = \frac{1}{2} * [(1 + \beta_i) * x_1(i) + (1 - \beta_i) * x_2(i) \tag{3.11}$$

where β_i is a random number that is generated using the following function:

$$PDF(\beta) \begin{cases} \frac{1}{2} * (\eta + 1) * \beta^{\eta} & if\ \beta \leq 1 \\ \frac{1}{2} * (\eta + 1) * \beta^{-(\eta+2)} & otherwise \end{cases} \tag{3.12}$$

where η is any non-negative real number. In this type of crossover, as in BLX- α, the spread of children solutions is proportional to that of the parents' solutions. Therefore, if the parents are distant then the children are also distant allowing exploration, while if the parents are closely spaced then the children are also closely space fostering exploitation. This is closely related with the concept of self-adaptive GA.

3.1.3.8 Max-min arithmetical crossover

In max-min arithmetical crossover we calculate 4 different children and then choose only the fittest two. The first two children are calculated using the formulas proposed in arithmetic crossover (Section 3.1.3.4):

$$y_1(i) = (1 - \alpha) * x_1(i) + \alpha * x_2(i) \tag{3.13}$$

$$y_2(i) = \alpha * x_1(i) + (1 - \alpha) * x_2(i) \tag{3.14}$$

The third one is a child that contains the minimum values of all the existing features and the fourth child contains the maximum values of all the existing features.

$$y_3(i) = \min(x_1(i), x_2(i)) \tag{3.15}$$

$$y_4(i) = \max(x_1(i), x_2(i)) \tag{3.16}$$

This type of recombination is proposed in [154] where the author solve a network-constrained economic dispatch using a continuous GA with roulette wheel selection and max-min arithmetical crossover.

3.1.3.9 Fuzzy connectives-based crossover

The fuzzy connectives-based crossover was proposed by Herrera et al. in 1997 [155] to avoid the premature convergence of the GA, this is, to avoid the stagnation of the algorithm into local optima which is provoked by a lack of diversity in the population or a disproportionate exploitation/exploration rate. As the name implies, this crossover method is based on fuzzy connectives which is a concept that comes from fuzzy logic theory. Since the objective is not to master fuzzy logic theory, this will be explained in a simplified way. This crossover operator divides the search space for each variable in different intervals as show in Figure 22.

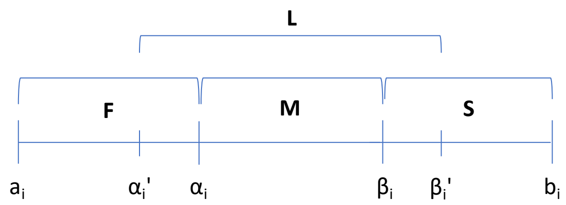


Figure 22 Division of the search space in the fuzzy connectives-based crossover

where the interval $[a_i, b_i]$ are the limits of the search space, the interval $[\alpha_i, \beta_i]$ are maximum and minimum values of the parent selected for variable i and $[\alpha_i{}', \beta_i{}']$ is called the relaxed exploitation interval and is broader than the previous one. As for the letters F, S, M and L, these are function that represent values in these intervals and are the most important part of FCB. They are defined as:

$$F = a + (b - a) * T(s, s') \tag{3.17}$$

$$S = a + (b - a) * G(s, s') \tag{3.18}$$

$$M = a + (b - a) * P(s, s') \tag{3.19}$$

$$L = a + (b - a) * C(s, s') \tag{3.20}$$

where a and b are the limits of the search space and s and s' are the normalized values of the variables in the interval [0,1] and the function T, G, P and S are the fuzzy connectives. These are respectively: t-norm, t-conorm, averaging function and generalized compensation operator. There are many ways of defining these four fuzzy connectives but in [155] the author proposed the following table:

Family	t-norm	t-conorm	Averaging fun. $(0 \leqslant \lambda \leqslant 1)$	Gen. comp. op.
Logical	$T_1(x, y) = \min(x, y)$	$G_1(x, y) = \max(x, y)$	$P_1(x, y) = (1 - \lambda)x + \lambda y$	$\hat{C}_1 = T_1^{1-\lambda} \cdot G_1^{\lambda}$
Hamacher	$T_2(x, y) = \frac{xy}{x + y - xy}$	$G_2(x, y) = \frac{x + y - 2xy}{1 - xy}$	$P_2(x, y) = \frac{1}{\frac{y - y\lambda - xy + x\lambda}{xy} + 1}$	$\hat{C}_2 = P_2(T_2, G_2)$
Algebraic	$T_3(x, y) = xy$	$G_3(x, y) = x + y - xy$	$P_3(x, y) = x^{1-\lambda} y^{\lambda}$	$\hat{C}_3 = P_3(T_3, G_3)$
Einstein	$T_4(x, y) = \frac{xy}{1 + (1 - x)(1 - y)}$	$G_4(x, y) = \frac{x + y}{1 + xy}$	$P_4(x, y) = \frac{2}{1 + \left(\frac{2-x}{x}\right)^{1-\lambda} \left(\frac{2-y}{y}\right)^{\lambda}}$	$\hat{C}_4 = P_4(T_4, G_4)$

Figure 23 Fuzzy connectives table (courtesy of [155])

All these families of fuzzy connectives were implemented in the EA-Tool and the user can do some testing with all of them, however, Herrera demonstrates in [155] that the Logical and Hamacher fuzzy connectives families are the best performing ones. In the Matlab tool the algorithm calculates the four fuzzy connectives, but it only keeps the two best performing ones, as in Max-min arithmetic crossover.

3.1.4 Mutation Types

The last operation in GA is called mutation. Mutation consists on adding random, unbiased information to an individual to generate offspring. Again, there are several types of mutation but in the EA-Tool only 5 types were implemented.

3.1.4.1 Uniform mutation centered at the middle of the search domain

Taking into account the probability of mutation which is typically low, a random number **between the maximum and minimum values** of the variable, is selected. This would be expressed as:

$$y_1(i) = U[y_{Min}(i), y_{Max}(i)] \tag{3.21}$$

3.1.4.2 Uniform mutation centered at $x_1(i)$

Same as before but instead of choosing a random value between maximum and minimum we set $x_1(i)$ at the center of the interval and define a deviation α. The resulting operation is:

$$y_1(i) = U[y_1(i) - \alpha_1(i),\ y_1(i) + \alpha_1(i)] \tag{3.22}$$

3.1.4.3 Gaussian mutation centered at the middle of the search domain

Instead of using a uniform distribution we can use a gaussian distribution and create random values. In this case we define a normal distribution with average in $c_i = (y_{Max}(i) + y_{Min}(i))/2$ and standard deviation defined by the user and proportional to the mutation magnitude. The resulting operation would be:

$$y_1(i) = \max[\min\left(y_{Max}(i), N\left(c_1(i), \sigma_1^2(i)\right), y_{Min}(i)\right)] \tag{3.23}$$

3.1.4.4 Gaussian mutation centered at $x_1(i)$

Same as before but the average of the normal distribution is $y_1(i)$. The resulting operation would be:

$$y_1(i) = \max[\min\left(y_{Max}(i), N\left(y_1(i), \sigma_1^2(i)\right), y_{Min}(i)\right)] \tag{3.24}$$

3.1.4.5 *Michalewicz's mutation or non-uniform mutation*

In [156], Michalewicz proposed a mutation operator that have been widely used by practitioners as in[154]. Imagine that we have a child in the form $y_1 = [y_1(1), y_1(2), \dots, y_1(i)]$. Then this mutation operator is defined as follows:

$$y_1^{mut}(i) = \begin{cases} y_1(i) + \Delta\left(t, y_1^{max} - y_1(i)\right) & if\ r = 0 \\ y_1(i) - \Delta\left(t, y_1(i) - y_1^{min}\right) & if\ r = 1 \end{cases} \tag{3.25}$$

Where r is random binary number and the function delta is defined as follows:

$$\Delta(t, y) = y * (1 - \xi^{\left(1-\frac{t}{T}\right)^b}) \tag{3.26}$$

Where ξ is a random floating point in the interval [0-1]; t is the current generation number, T is the maximum number of generations and b is a parameter to determine the degree of dependence in the number of generations.

With this type of mutation, we make a uniform search at the beginning of the algorithm and narrow the search to a local are at the end.

3.1.5 Adaptive Genetic Algorithm

There is a possibility to improve the performance of the GA if the probabilities of mutation and crossover are able to vary in every generation to change, for example, from exploration to exploitation. This is usually called Adaptive Genetic Algorithm [10].

In the Adaptive GA it is recommended to start with a high probability of crossover to promote local exploration and a low probability of mutation to avoid a random search. However, as the generations increase, the probability of crossover should decrease, and the probability of mutation should increase since it is likely that the process is stagnated in a local minimum. The way to implement this in the EA-Tool is just by adding an equation that increments or decrement these values linearly as the number of generations increases:

$$prob(g) = prob_{ini} + \frac{prob_{end} - prob_{ini}}{Generation\ Limit} * g \tag{3.27}$$

where $prob(g)$ is the probability at generation g, $prob_{end}$ is the probability at the end of the simulation, $prob_{ini}$ is the probability at the beginning of the simulation, $Generation\ Limit$ is the maximum number of generations that the algorithm will do and g is the current generation number.

3.2 Particle Swarm Optimization in the tool

3.2.1 Basic Information

The basic information about Particle Swarm Optimization and the pseudocode of the algorithm can be found in Section 2.1.2.1

3.2.2 The velocity update equation

In the algorithm presented in Figure 5, it is shown that the velocity is updated stochastically depending on a series of parameters in each generation. This velocity is the one at which each particle moves through the search space searching a solution and it is updated using the following equation:

$$v_i(t) = v_i(t-1) + \phi_1 * (b_i - x_i) + \phi_2 * (h_i - x_i) \tag{3.28}$$

$$\phi_i = r_i * \phi_{i,max} \quad \forall i = 1,2 \tag{3.29}$$

where x_i is the current candidate solutions, b_i is the historical best position of the current individual, h_i is the best neighbor so far and ϕ_i are the acceleration coefficients which are calculated randomly using the user-defined maximum values and a random number r_i. In this case:

- ϕ_1 is the cognition learning rate and it indicates the degree of influence that the best position so far of the current individual has over itself.
- ϕ_2 is the social learning rate and indicates the degree of influence that the best individual in the neighborhood has over the rest of the neighborhood. If the neighborhood is equal to the population size (gbest) then it the influence that the best individual of the population has over the rest.

3.2.3 PSO Variations

Something very important in the PSO Algorithm is to ensure convergence and prevent the "explosion" of the swarm, which occurs when particles diverge and go to infinity. To do so, there are different variations found in literature that are presented in this section.

3.2.3.1 Define the maximum velocity

As seen in Section 3.2.2 the velocity in PSO is calculated randomly and there are no limitations in the value it can take. This could be an issue since it can produce an uncontrolled trajectory of the particles producing wide oscillations in the search space [157]. To control this, a maximum velocity (v_{Max}) could be defined:

$$-v_{Max} \leq v_i \leq v_{Max} \tag{3.30}$$

This parameter is usually selected empirically, taking into account that a very big value could produce deep oscillations while a very small value will limit the search and the algorithm might not converge. In the tool the value was chosen equal to the maximum value of the search space for a specific variable. The maximum velocity definition is easy to implement; however it does not ensure the convergence of the algorithm.

3.2.3.2 Decremental Inertia Particle Swarm Optimization (DIPS)

One of the methods to avoid the "explosion" of the swarm is called inertia weight (w). This method was proposed by Shi and Eberhart [158] and controls the impact of the previous history of velocities on the current velocity by multiplying the inertia term of the velocity update equation $v_i(t-1)$ by a number w defined by the user:

$$v_i(t) = w * v_i(t-1) + \phi_1 * (b_i - x_i) + \phi_2 * (h_i - x_i) \tag{3.31}$$

If this number is large, then it facilitates exploration (searching new areas) and if it is low it facilitates exploitation (search deeper in a specific area). For this reason, it is common in literature to find a dynamic inertia weight that starts at high values (e.g. 0.9) to foster exploration and decrease it linearly with the number of generations to a low value (e.g. 0.4) to foster exploitation. This is referred to as "Decremental Inertia Particle Swarm Optimization" (DIPSO) [159] and it adds the following equation to the algorithm:

$$w = w_{ini} + \frac{w_{end} - w_{ini}}{GenLimit} * Gen_t \tag{3.32}$$

where w_{ini} is the initial inertia weight, w_{end} is the final inertia weight, $GenLimit$ is the total number of generations and Gen_t is the current generation.

3.2.3.3 Constricted Particle Swarm Optimization

Another variation that avoids the "explosion" of the swarm is known as constricted PSO (cPSO) and was developed by Clerc and Kennedy [29]. It consists on multiplying the entire velocity update equation by a factor K, which is called the constriction factor. The new velocity update equation would then be:

$$v_i(t) = K * [v_i(t-1) + \phi_1 * (b_i - x_i) + \phi_2 * (h_i - x_i)] \tag{3.33}$$

The constriction factor, unlike the inertia weight, does not have a typical value but it has to follow some rules in order to ensure the stability of the algorithm. This was demonstrated mathematically in [29] where they explain that to ensure the stability of the swarm the following conditions must be met:

$$\phi_T = \phi_1 + \phi_2 > 4 \tag{3.34}$$

$$K = \frac{2 * \alpha}{2 - \phi_T - \sqrt{\phi_T^2 - 4 * \phi_T}} \tag{3.35}$$

where α is called the constriction coefficient and it is a value between 0 and 1.

The PSO algorithm can have either inertia weight or constriction factor but not both. If the user selects one in the tool, the other is automatically deselected so that there is not possibility of mistake.

3.2.3.4 PSO with Time Varying Acceleration Coefficients

The most relevant operator in the PSO algorithm is the velocity update equation which is influenced mainly by the acceleration coefficients ϕ_i. For this reason, in [160], the author studied the influence of these coefficients on the algorithm and he observed that if the cognition learning rate (ϕ_1) is high compared with the social learning rate (ϕ_2) then the algorithm will perform exploration. If, on the other hand, the social learning rate is higher the algorithm will perform exploitation. Considering these ideas, Ratnaweera proposed in 2004 the concept of time-varying acceleration coefficients (TVAC) [161], with the goal of enhancing the global search at the beginning of the algorithm and encourage the particles to converge towards the global optimum at the final stages. To do so, the following equations are added to the algorithm:

$$\phi_i = \phi_{i_{end}} - \phi_{i_{ini}} * \frac{Gen_t}{GenLimit} * \phi_{i_{ini}} \quad \forall i = 1,2 \tag{3.36}$$

In [161] several tests are done using different values for $\phi_{i_{end}}$ and $\phi_{i_{ini}}$ but they show that the best performing strategy is to decrease the cognition learning rate (ϕ_1) from 2.5 to 0.5 while increasing the social learning rate (ϕ_2) from 0.5 to 2.5.

3.2.3.5 Self-organizing hierarchical PSO with Time Varying Acceleration Coefficients

Additional to the time varying acceleration coefficients, in [161] the author proposes the self-organizing hierarchical PSO. In this PSO variation the inertia component of the velocity update equation is eliminated, which produces a fast convergence towards a local optimum resulting in an incorrect solution. To avoid this, it was proposed to reinitialize the velocity in case it is stagnated (equal to 0). The following equation is added to the algorithm together with the TVAC:

$$if\ v_t = 0\ then\ y_1^{mut}(i) = \begin{cases} v_t = rand * v_{Max} & if\ r < 0.5 \\ v_t = -rand * v_{Max} & if\ r \geq 0.5 \end{cases} \tag{3.37}$$

3.2.4 Mutation in PSO

One of the main limitations of the PSO algorithm is the premature convergence, this is, going rapidly towards a local optimum as if it was the global one. To avoid this, the introduction of a mutation operator typical from Genetic Algorithm, has been proposed. This operator randomly mutates a variable from an individual whenever the algorithm seems to be stagnated, this is, when the best individual of the population does not improve its fitness value from one generation to the next. Although there are more in literature, only two types of mutation were implemented in the tool

3.2.4.1 Non-uniform mutation

Proposed by Michalewicz's for Genetic Algorithms [156], this type of mutation can be extended to PSO, as in [162]. This mutation technique is very convenient since it searches uniformly at early stages and locally at the end. For a complete explanation of non-uniform mutation see the Genetic Algorithm documentation.

3.2.4.2 Cauchy Mutation

The Cauchy mutation was proposed by Hu et al. [162] because of its capability in generating a larger range of jump steps compared to other operators. In this type of mutation, a random variable from a random individual is mutated using the following equation:

$$x_{i,k} = x_{i,k} + cauchy(\delta_i) \tag{3.38}$$

$$cauchy(\delta_i) = \tan\left(\pi * \left(y - \frac{1}{2}\right)\right) * \delta_i \tag{3.39}$$

where y is a random number between 0 and 1 and δ_i is the scaling factor of the Cauchy distribution. This parameter must have two properties according to Hu et al.: (1) ensure the magnitude of the mutation is at the same scale as the particle movement and (2) enable larger moves at initial stages and smaller moves at the end. To satisfy these conditions first the Euclidean norm of each velocity vector is calculated:

$$\delta_i \propto [\|v_1\|_2, \dots, \|v_n\|_2] \tag{3.40}$$

And then a constant k is also calculated:

$$k = PopSize * \frac{Gen_t}{GenLimit}^{2-\frac{Gen_t}{GenLimit}} \tag{3.41}$$

This constant increases non-linearly with the number of generations. Once this is calculated and in order to satisfy the two properties explained above the scaling factor δ_i is set equal to the kth largest value of the velocity norms vector.

4 Implementation

This section shows how the previously explained theoretical concepts where put into practice. In Section 4.3 the structure of the optimization tool will be explained, while in Section 4.4 will present the data collection process of the real case study chosen to test the tool: Graciosa Island in Azores (Portugal).

4.1 Economic Dispatch problem in the Smart Grids' context

4.1.1 Background

In this dissertation two Evolutionary Algorithms (EAs) will be used, GA and PSO, to solve the optimal operation of a Smart Distribution Grid. The case study to be solved will include technologies such as solar PV, wind, stationary storage or diesel generation. Apart from that, it might be possible to include demand response on the load side and a connection to the electricity network, that will not be used in this dissertation but will exist as a possibility in the tool. The basic scheme of this SG is presented in Figure 24.

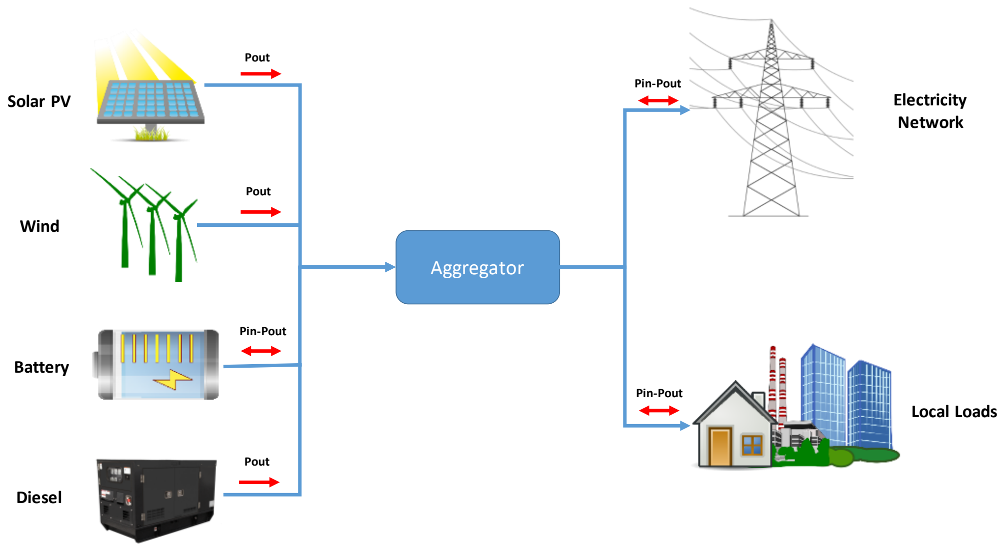


Figure 24 Scheme of the Smart Grid implemented in the tool

Here the figure of the aggregator is introduced. An aggregator is typically defined as an "*entity that acts as a mediator between end-users and electricity service operators*" [163][164]. In this thesis, the aggregator will be representing also the SG operator in charge of the optimal scheduling and operation of the system. This entity will be used to solve the SGs problem proposed. The scheduling horizon of the problem will be up to 24 hours, since price, demand and VRE forecasts are typically obtained 24 hours ahead. Each technology has different parameters, variables that will be relevant to formulate the optimization problem. These will be described in the following sections.

4.1.2 Economic Dispatch Problem Formulation

The Economic Dispatch (ED) problem of a Smart Grid consists on dispatching a series of DER and flexible loads in an optimal way given a demand profile and a generation mix. With the technologies and structure presented above the problem formulation can be made. As any other ED optimization problem, it will be composed by an objective function that will be used to calculate the fitness value of the candidate solutions, and some constraints that must be satisfied in order to consider the candidate solution as valid. The presented problem is similar to the formulation of the classical ED presented in [165]., which is the reference used to build the formulation.

Nomenclature

Indexes and sets

$b \in B$	Battery Storage Systems
$t \in T$	Time period (in our case, hour)
$vre \in VRE$	Variable Renewable Energy Technologies

Parameters

$P_{vre_{installed}}$	Total capacity installed of a VRE (kW)
CF_{vre_t}	Capacity Factor of a VRE in time t (%)
$\overline{P_b}.$	Maximum power of battery b (kW)
$\overline{E_b}$	Maximum energy that battery b can store (kWh)
η_{charge_b}	Charging efficiency of battery b (%)
$\eta_{discharge_b}$	Discharging efficiency of battery b (%)
$RepCost_b$	Replacement cost of battery b (€/kWh)
$SoC_{ini,b}$	Initial State of Charge of battery b (%)
$SoC_{end,b}$	Final State of Charge of battery b (%)
SoC_{Min}, b	Minimum State of Charge of battery b (%)
SoC_{Max}, b	Maximum State of Charge of battery b (%)
$\overline{P_{diesel}}.$	Maximum Power of the diesel generator (kW)
a	No load cost of the diesel generator (GJ/h)
b	Linear term of the heat rate function for the diesel generator (GJ/kWh)
c	Quadratic term of the heat rate function for the diesel generator (GJ/kWh^2)
$Fcost$	Price of diesel per thermal unit (€/GJ)
D_t	Demand at time t (kW)
$\overline{DR_t}$	Maximum amount of demand response at time t (%)
DR_{price}	Price of demand response (€/kW)
$\overline{P_{grid}}.$	Maximum power of the grid connection (kW)
$Spot_t$	Spot price[2] at time t (€/kWh)

[2] The spot price is typically defined as the marginal cost of the most expensive generator commited in the day-ahead market

Variables

nse_t	Non-supplied energy at time t (kW)
$curt_t$	Curtailment at time t (kW)
P_{vre_t}	Power generated by a VRE at time t (kW)
$P_{b,t}$	Power output of the battery b at time t (kW)
$AgingFactor_{b,t}$	Ageing Factor of the battery b at time t (-)
$DegCost_{b,t}$	Degradation Cost of the battery b at time t (€)
P_{diesel_t}	Power output of the diesel generator at time t (kW)
$DieselCost_t$	Cost of diesel generation at time t (€)
DR_t	Demand response used at time t (kW)
P_{grid_t}	Power exchanged with the grid at time t (kW)

<u>Formulation</u>

Objective Function

The objective function represents the sum of costs of buying energy from the grid and degradation cost of the battery, which are to be minimized. Another possibility is to maximize the profit of the aggregator, but it can be easily proven from a modeling perspective that the results should be the same [1]. The objective function proposed is:

$$\min \sum_t P_{grid_t} * Spot_t \quad + DegCost_t + DR_t * DR_{price} + DieselCost_t \tag{4.1}$$

$s.t.$

Power Balance Constraint

The sum of all the power supplied by generators plus demand response plus a possible non-supplied energy has to be equal to the power demand plus a possible power curtailment.

$$P_{grid_t} + P_{b,t} + P_{vre_t} + DR_t + P_{diesel_t} + nse_t = D_t + curt_t \qquad \forall t \tag{4.2}$$

VRE Constraints

The power supplied by VRE is equal to the installed capacity multiplied by the capacity factor at time t

$$P_{vre_t} = P_{vre_{installed}} * CF_{vre,t} \qquad \forall t \tag{4.3}$$

Battery Storage Constraints

The power supplied or consumed by the battery has to be within the limits of the inverter.

$$-\overline{P_b} \leq P_{b,t} \leq \overline{P_b} \qquad \forall t, b \tag{4.4}$$

The SoC of the battery has to be within reasonable limits for the battery chemistry selected. In the case of Li-ion batteries the minimum SoC could be 15-20%, while the maximum SoC should be 85-90%.

$$SoC_{Min} \leq SoC_{b,t} \leq SoC_{Max} \qquad \forall t, b \qquad (4.5)$$

At every time step the SoC changes depending on the discharged or charged power. The efficiency (η) used is different in each case. Also, the equation changes in the first and last period of the simulation to include the user-defined initial and final SoC.

$$SoC_{b,t} = SoC_{b,t-1} - \frac{P_{b,t} * 100}{\overline{E_b}} * \eta \qquad \forall t, b \qquad (4.6)$$

$$\eta = \begin{cases} \dfrac{1}{\eta_{discharge_b}} & if\ P_{b,t} \geq 0 \\ \eta_{charge_b} & if\ P_{b,t} < 0 \end{cases} \qquad \forall t, b \qquad (4.7)$$

$$SoC_{b,t-1} = \begin{cases} SoC_{ini_b} & if\ t = 1 \\ SoC_{end_b} & if\ t = T \end{cases} \qquad \forall t, b \qquad (4.8)$$

The degradation cost of the battery is equal to an ageing factor, which is calculated using a battery ageing model, multiplied by the replacement cost, which is typically equal to the battery storage cost.

$$DegCost_{b,t} = AgingFactor_t * RepCost \qquad \forall t, b \qquad (4.9)$$

Diesel Generator

The power supplied by the diesel generator cannot be higher than the installed capacity or lower than 0.

$$0 \leq P_{diesel_t} \leq \overline{P_{diesel}} \qquad \forall t \qquad (4.10)$$

The cost incurred by the diesel generator is a quadratic function multiplied by the fuel cost.

$$DieselCost_t = \left(a + b * P_{diesel_t} + c * P_{diesel_t}^2\right) * FCost \qquad \forall t \qquad (4.11)$$

Demand Response

At every time step the amount of demand response (shedded load) cannot be higher than the user-defined demand response band [186].

$$0 \leq DR_t \leq \overline{DR_t} * D_t \qquad \forall t \qquad (4.12)$$

Grid Connection

The amount of power absorbed or spilled to the grid cannt be higher than the hired grid connection power.

$$-\overline{P_{grid}} \leq P_{grid_t} \leq \overline{P_{grid}} \qquad \forall t \tag{4.13}$$

4.2 Battery Degradation Model

Ageing is one of the key issues of the batteries and it is necessary to take it into account in their evaluation. To do so an indicator that measures this should be created. In literature the most common one is the State of Health (SOH), defined in [166] as

$$SOH(t) = \frac{C_{nominal_t}}{C_{nominal_{initial}}} * 100\ (\%) \tag{4.14}$$

where $C_{nominal_t}$ is the nominal capacity at time t after suffering ageing and $C_{nominal_{initial}}$ is the initial nominal capacity of the battery (as in the data sheet). In this case, the indicator is measured over the total capacity, however it is a common practice to evaluate it over the End Of Life (EOL) which is usually defined as 80% of the nominal capacity [167]. The equation would then be as follows:

$$SOH(t) = \frac{C_{nominal_t} - C_{nominal_{initial}}}{EOL * C_{nominal_{initial}}} * 100\ (\%)) \tag{4.15}$$

Here a 0% SOH means that the battery has reached the EOL while in 3.1 the EOL would be 80% SOH.

Once an indicator has been measured, the main problem is to estimate the nominal capacity at time t. For this purpose, different methods have been proposed in literature. In [142] and [168] the following methods can be found:

- Electrochemical models, which model the battery in detail taking into account all the reactions that occur inside. It is divided in phenomenological approaches and atomistic and molecular approaches
- Equivalent circuit-based models, that use simple circuits composed by series resistances a couples of resistances and capacitors in parallel to model the battery
- Performances based models, which use equations obtained from empirical data to model the degradation
- Analytical models with empirical fitting, that uses estimators to obtain the battery degradation
- Statistical approach, which is based on data and no knowledge about the battery is required
- Weighted Ah-throughput model, that uses the nominal Ah-throughput of the battery and the Ah-throughput at time t weighted by specific factors that represent the operating conditions
- Event-oriented model, that is based on Wöhler curves

The main issue of most of these methods is that empirical data is needed, and most of the times it is not possible to obtain this data from a specific battery, sincelaboratory experiments are required. Only the last two methods (weighted Ah-throughput model and event-oriented model) can be applied using expert estimates and data from manufacturers data sheets.

4.2.1 Degradation model in the tool

After a deep literature review on battery degradation modeling, it was concluded that the most adequate model for the purposes of this dissertation is the one proposed by Magnor et al. [170], where a semi-experimental model is proposed. The equations obtained come from laboratory experiments using a Lithium-ion battery with a capacity of 7.2Ah, however the model can be generalized for other Li-ion batteries by modifying the input data. An advantage of this model is that the input data can be sometimes found in data sheets or in papers in which a specific battery was tested. Additionally, this model considers not only cyclic ageing, but also calendric ageing as will be shown in the following sections.

4.2.1.1 Calendric Ageing Model

Calendric ageing is the ageing produced due to the progress in time and it is mainly influenced by the storage conditions of the battery which are the temperature and state of charge (SoC) as already explained in Section 2.4.1.3.

In [170] the temperature influence is considered using the Arrhenius' law that says that an increase in temperature by 10-15K halves the lifetime of the battery. The equation proposed is the following:

$$c_{temp} = \frac{1}{t_{cal,ref}} * \int_{t_0}^{t_1} 2^{(\frac{T_{bat}-T_0}{\Delta T})} dt \tag{4.16}$$

where $t_{cal,ref}$ is the reference calendric lifetime of the battery, T_0 is the temperature at which the calendric lifetime is obtained, ΔT is the variation of temperature that halves the lifetime of the battery (10-15K) and T_{bat} is the temperature of the battery. The most accurate way of calculating T_{bat} would be using a thermal model of the battery, however, this is beyond the scope of this dissertation and T_{bat} will be assumed to be constant as in [171], resulting the following equation:

$$c_{temp} = \frac{1}{t_{cal,ref}} * 2^{\left(\frac{T_{bat}-T_0}{\Delta T}\right)} * t\_step \tag{4.17}$$

where t_step is the length of the time steps in the simulation (typically 1 hour).

As for the impact of the state of charge, the following exponential equation is used:

$$c_{SoC} = \frac{1}{a + b * \exp(c * (100 - SoC))} \tag{4.18}$$

Where a,b and c are the adjusting parameters and are typically obtained from empirical studies. In this dissertation the values proposed by [170] for Li-ion batteries will be used. These are:

$$a = 2$$

$$b = -1.2$$

$$c = -0.0275$$

Once c_{temp} and c_{SoC} have been calculated, $c_{calendric}$ can be simply obtained by multiplying the two previous values:

$$c_{calendric} = c_{temp} * c_{SoC} \tag{4.19}$$

4.2.1.2 Cyclic Ageing Model

Cyclic ageing describes the degradation of the battery due to operation. Here the main influencing factors are the depth of discharge (DoD) and the frequency of charging/discharging the battery. In [170] only the DoD is considered since it has a higher influence than the frequency of cycling on the total degradation. To model cyclic ageing a Wöhler Curve, as shown in Figure 25 is used.

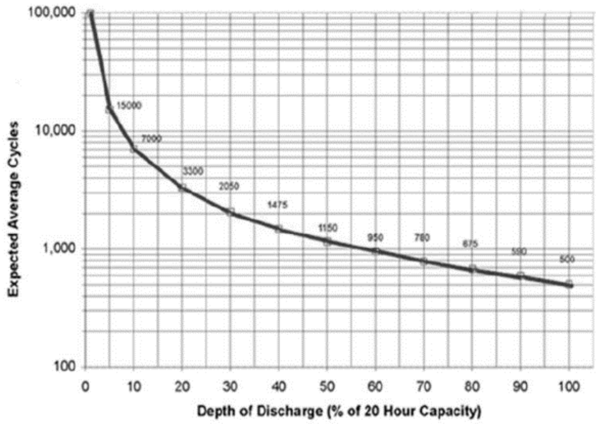


Figure 25 Battery Wöhler Curve (courtesy of [172])

The Wöhler curve shows the number of cycles that the battery can made depending on the depth of discharge. In [170] the authors propose the following equation to fit the Wöhler curve:

$$N_{max,\Delta SoC} = a * \Delta SoC^{b} \tag{4.20}$$

Where $N_{max,\Delta SoC}$ is the maximum number of cycles that the battery can make at that ΔSoC and a and b are the adjusting parameters of the curve. Then, given a Wöhler curve and knowing two points, the user can calculate parameters a and b.

Once all the points from the curve are known and the ΔSoC is given as an input, the cyclic ageing factor (c_{cyclic}) can be calculated as:

$$c_{cyclic} = \frac{N_{\Delta SoC}}{N_{max,\Delta SoC}} \tag{4.21}$$

Where $N_{\Delta SoC}$ is the number of cycles produced because of the depth of discharge at time t.

4.2.1.3 Total degradation

The total ageing factor is calculated as the maximum between calendric ageing and cyclic ageing. This ageing factor is also equal to the variation of the state of health of the battery that is needed to include degradation in the model proposed.

$$c_{total} = \Delta SOH(t) = \max(c_{calendric}, c_{cyclic}) \tag{4.22}$$

4.2.2 Accounting for degradation in an economic dispatch problem

Degradation models have already been used in literature to calculate the amount of degradation of a battery related to its operation. This has been used, for instance, for storage project valuation to calculate the number of battery replacements needed in a given time period and finally calculate the replacement costs as in [173]. These costs are defined as:

$$C_{Rep} = \mathrm{k} * \mathrm{C_{bat}} \tag{4.23}$$

where C_{Rep} are the replacement costs in €/kWh, k is the number of replacements in the period considered and $\mathrm{C_{bat}}$ is the cost of the battery energy storage in €/kWh. In the replacement costs calculation only the cost of the energy storage is considered, since battery cells are the typical components to be replaced (the inverter and other components have a longer life). It is therefore clear that the cost related with degradation will be the replacement costs.

When degradation is considered nowadays it is typically based on a 'posteriori' calculation in which the dispatch of the battery has already been obtained. This could be sometimes a good approach, however, the battery might be cycling too much or operating under extreme storage conditions which might result in an overestimation of the number of replacements and a misuse of the battery storage system.

To solve this issue, battery degradation must be directly considered in the economic dispatch problem by adding a cost into the objective function. This is the one of the main original contributions of this thesis work. This way the battery will be dispatched only if the savings produced to the system overcome a certain cost of degradation at time t. The term added to the objective function is:

$$\sum_t DegCost_t = \sum_t \mathrm{C_{Rep}} * \Delta SOH(t) \tag{4.24}$$

where $\mathrm{C_{Rep}} = C_{bat}$ and $\Delta SOH(t)$ is calculated with the degradation model explained in the previous section.

4.3 Optimization Tool Structure

The present section describes the structure of the Matlab-based Tool that was developed and is the main outcome of this dissertation. It will start from the Graphical User Interface (GUI) developed with Matlab 's AppDesigner to later explain every single function that runs in the background of the tool.

4.3.1 Graphical User Interface

The only part of the tool that the user is intended to interact with is what has been called the Graphical User Interface or GUI. This has been developed using 'AppDesigner' [174], a Matlab tool that allows to create applications. An example of how the GUI looks like is shown in Figure 26.

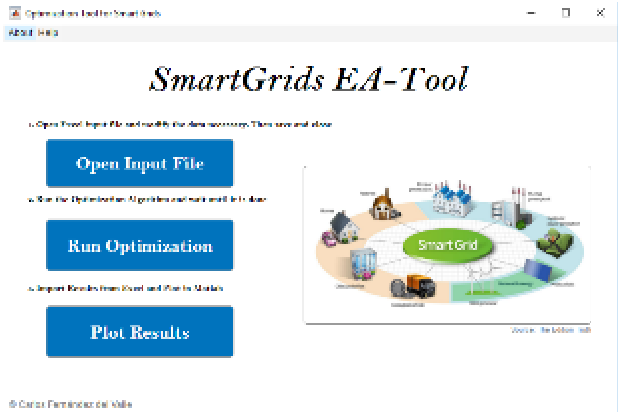


Figure 26 EA-Tool Graphical User Interface

As seen, the tool has 3 different options where the user can choose. In Figure 27 the task performed by each of the options is shown.

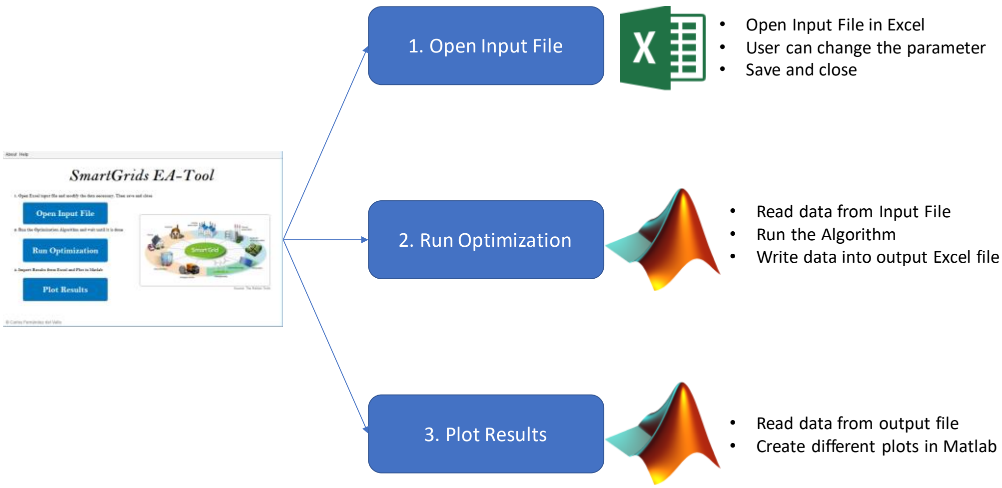


Figure 27 Main structure of the EA Tool

The tool does not only use Matlab but also uses Excel since the author believes it is a more powerful tool to handle the input data and store all the results from simulations. In the following sections we will go deeper into each of the three options that can be selected in the GUI.

4.3.2 Input Data

The first button of the GUI opens an Excel file where the user can introduce the data that will serve as an input for the optimization problem. This file contains some Macros to improve the user experience. The Input File contains 8 different sheets that are described in Table 4-1.

Table 4-1 Description of the Input Data File

Sheet	Description
Info	The Info Sheet presents a brief description of the Input File, information about the author as well as an index of sheets that the file contains and what the user can do in each of them. This sheet has no influence in the operation of the tool.
Algorithm Options	In this sheet the user can find the available options for both Genetic Algorithm (GA) and Particle Swarm Optimization (PSO).
Problem Info	In the Problem Info Sheet, the user is intended to introduce general information about the problem to solve or about the simulation
Demand	In the Demand Sheet the user will introduce all the information related to the demand (e.g. Demand profile or demand response information)
Grid Connection	In this sheet the user is supposed to introduce all the information related with a possible connection of the Smart Distribution Grid to a bigger electric power system.
Diesel	In the diesel sheet the user will introduce the information related with the diesel generator in case there is a diesel generator in the specific case study.
Battery	In the Battery Sheet the user can introduce diverse types of battery chemistries that will be included in the simulation. At the top of this sheet there is a button called 'Add empty row' that will add a row to the Excel table where the user can include a new type of battery
VRE	VRE stands for Variable Renewable Energy and therefore in this sheet the user will introduce all the information related with wind energy and solar PV in the Smart Grid (e.g. time series and capacity installed)

4.3.3 Run Optimization

The second button 'Run Optimization' will run the optimization problem using the parameters specified in the Input file and at the end of the simulation will create an output Excel file with a unique name in order to store every single simulation made by the user. The structure of the algorithm that this button executes is shown in Figure 28

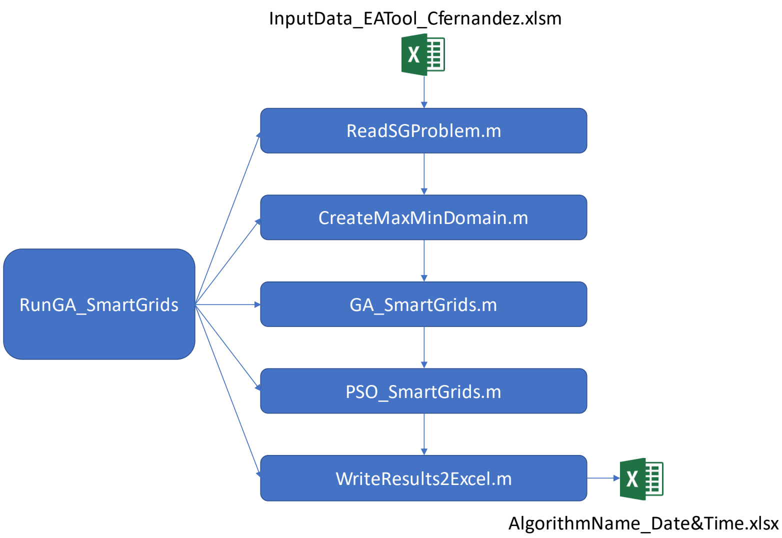


Figure 28 Structure of the main algorithm

The main algorithm is called ‚RunGA_SmartGrids', which is a script that runs a set of functions. Each of these function is explained in Table 4-2.

Table 4-2 Functions executed by the main algorithm

Matlab Function	Description

ReadSGProblem	This function receives the path where the Input File is located, reads all the information from there using the ActiveX server from Matlab and finally store this information into a Matlab structure called SGProblem.
CreateMaxMinDomain	This function creates the limits of the search space for each variable in the simulation. The function is used to define bounded constraints (e.g. maximum power of the diesel generator)
GA_SmartGrids and PSO_SmartGrids	These two functions contain the core algorithm or, saying it in classical optimization words, the solvers. Since the objective is to be able to run the same problem with two different evolutionary algorithms there is a function for the Genetic Algorithm (GA) and another for the PSO Algorithm (PSO).
WriteResults2Excel	This function receives the set of results that the user wants to store and it post-process them for a clearer presentation. After that, it writes the results into an Excel file via the ActiveX server. The function creates an Excel file for PSO and another for GA with the following format: 'AlgorithmName_Date&Time.xlsx'. These files can be found inside the Results folder which is inside the working folder

From all these functions the most relevant ones are those that contain the GA and PSO algorithms. These are very complex functions whose structure should be explain separately. A scheme of these functions is shown in Figure 29 and its description is presented in Table 4-3.

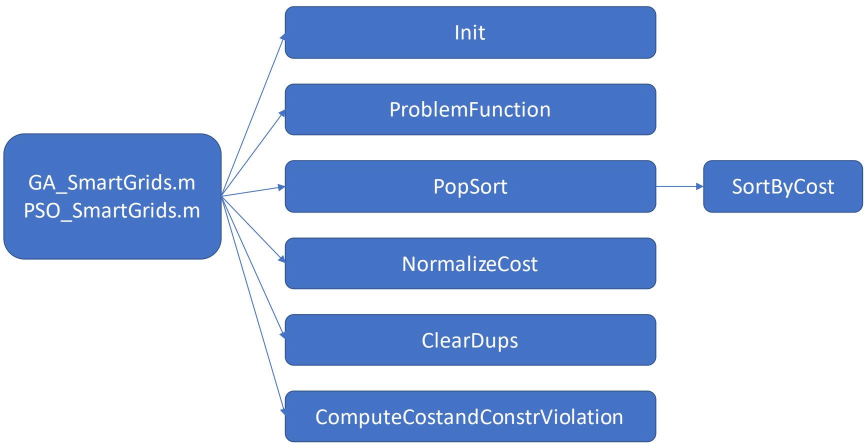


Figure 29 Structure of the GA and PSO main function

Table 4-3 Functions executed by the solver

Matlab Function	Description
Init	The function 'Init' initializes a random population with a user-specified size and within the limits of the search space.
Problem Function	The problem function includes the objective function and constraints and it is used to calculate the fitness value of each element from the population. In this dissertation the problem function is called 'SmartGrids_CaseStudy.m'.
PopSort	This function receives a population in which the fitness values have already been calculated and sort it by cost. To do so it uses a second function called 'SortByCost'.
NormalizeCost	The function 'NormalizeCost' receives the sorted population and normalize its costs being 0 the best individual and 1 the worst (if the goal is to minimize). This, together with some operation to invert the values, is used in the Genetic Algorithm to assign probabilities of selection to individuals of the population.
ClearDups	This function compares each individual of the population with each other and in case of finding two identical ones, it creates a random individual that substitute the duplicate. This increase the exploration of the algorithm.
ComputeCostandConstrViolation	This function receives the Population and the current generation number and has two main tasks: (1) it stores the minimum cost, average cost, minimum constraint violation and average constraint violation, and (2) it displays this information in the Matlab desktop to let the user know how the simulation is going.

4.3.4 Plot Results

The third button ('Plot Results') receives an Excel file selected by the user with the format previously created and plots a set of results to make a quick assessment of the simulation. The structure of the algorithm that this button executes is shown in Figure 30.

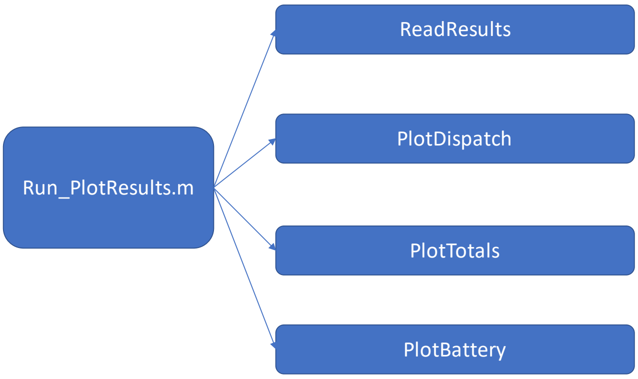


Figure 30 Structure of the post-processing algorithm

The post-processing of the tool is ready to plot only a one-battery system with no grid connection. This part of the tool could be improved in the future to make a more genera post-processing that allows more than one battery and a grid connection. It is left open for further development. Additionally, in the thesis report, the figures presented will be made with Excel to provide uniformity to the document (input data and other figures are in Excel and not in Matlab).

The main algorithm is called 'Run_PlotResults' and it executes a set of functions that are described in Table 4-4.

Table 4-4 Functions executed by the post-processing algorithm

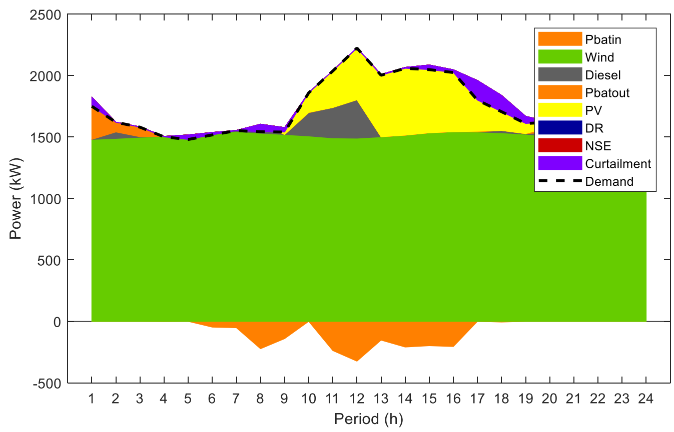
<table>
<tr><th>Matlab Function</th><th>Description</th></tr>
<tr><td>ReadResults</td><td>This function read the results from the selected Excel files, then it filters by run and selects the one with the lowest cost. The output of this function is a matrix with the values of the variable for the lowest cost run.</td></tr>
<tr><td>PlotDispatch</td><td>The function PlotDispatch organize the results and plots a 24-hour dispatch using areas with a predefined format. An example is shown in Figure 31


Figure 31 Dispatch of Graciosa island in an average Winter day</td></tr>
<tr><td>PlotTotals</td><td>This function plots a pie chart with the proportion of demand covered with each of the sources used to solve the optimization problem. An example is shown in Figure 32</td></tr>
</table>

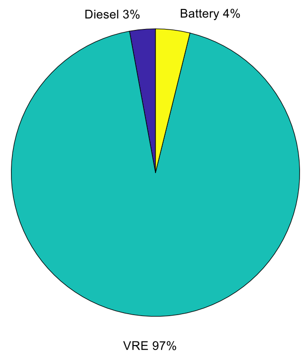
<table>
<tr>
<td></td>
<td>


Figure 32 Demand provision by source for Graciosa Island in an average Winter day
</td>
</tr>
<tr>
<td>**PlotBattery**</td>
<td>
The function PlotBattery, plots the net generation of the battery on one side and on the other side the state of charge per time step of the battery. An example is shown in Figure 33

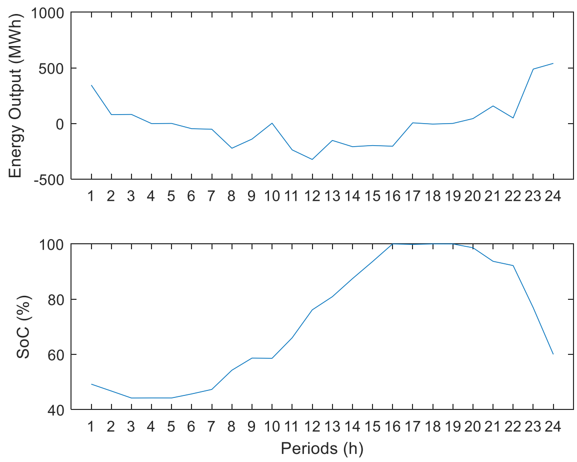


Figure 33 Energy output vs. SoC of the battery installed in Graciosa Island
</td>
</tr>
</table>

4.4 Case Study: Graciosa Island microgrid

4.4.1 Basic Information

Graciosa is a small island located in the Atlantic Ocean, which is part of the Azores arcipelagum (Portugal), together with 8 other islands (see Figure 34 Location of Graciosa Island). Distributed in a total area of around 60.66 km^2, the population of this small piece of the Azores is estimated as of 4301 inhabitants in 2016 [175], which leads to a population density of 70.9 inhabitants per km.

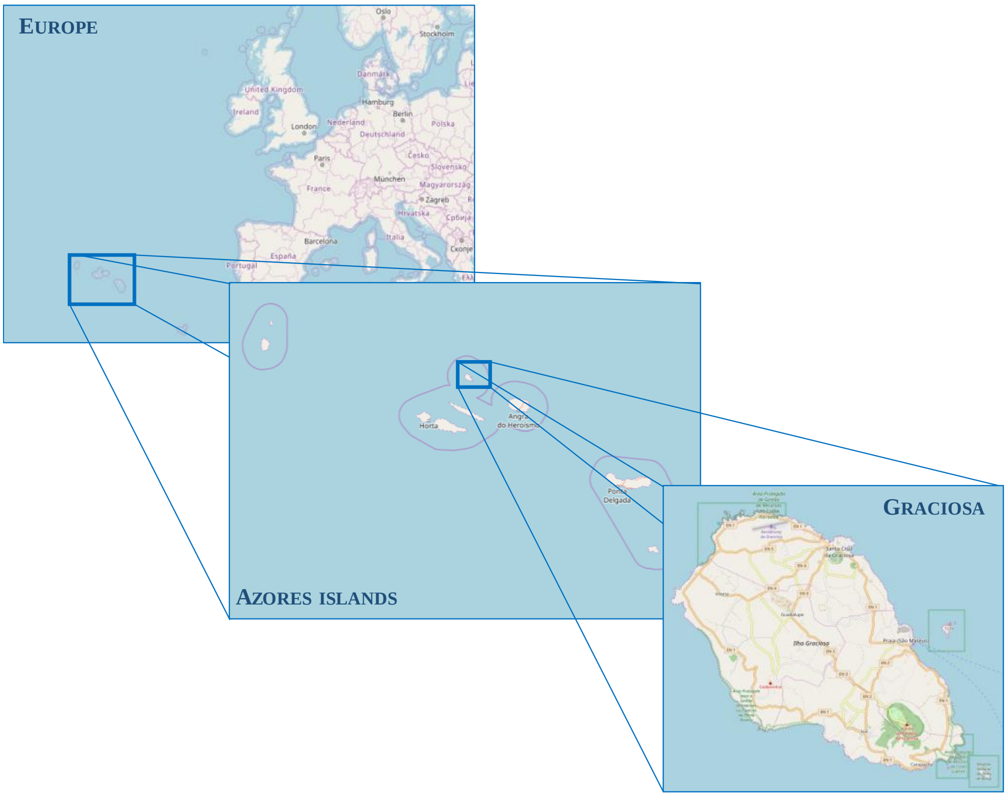


Figure 34 Location of Graciosa Island

4.4.2 Microgrid system

According to [176], the total electricity demand in Graciosa was 12.73GWh in 2015. This demand can be then divided by sector as in Figure 35.

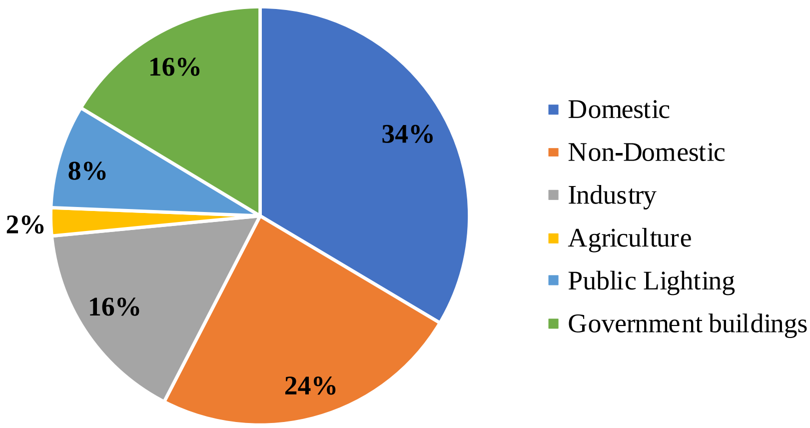


Figure 35 Electricity consumption by sector in Graciosa

Most of the electricity consumption in Graciosa is due to domestic and non-domestic applications; agriculture, industry and government buildings must also be taken under consideration.

The demand in Graciosa is not uniform throughout the year and is typically higher during summer months when tourism in the island increases. This can be clearly observed in Figure 36, which shows the monthly electricity consumption in Graciosa for year 2015.

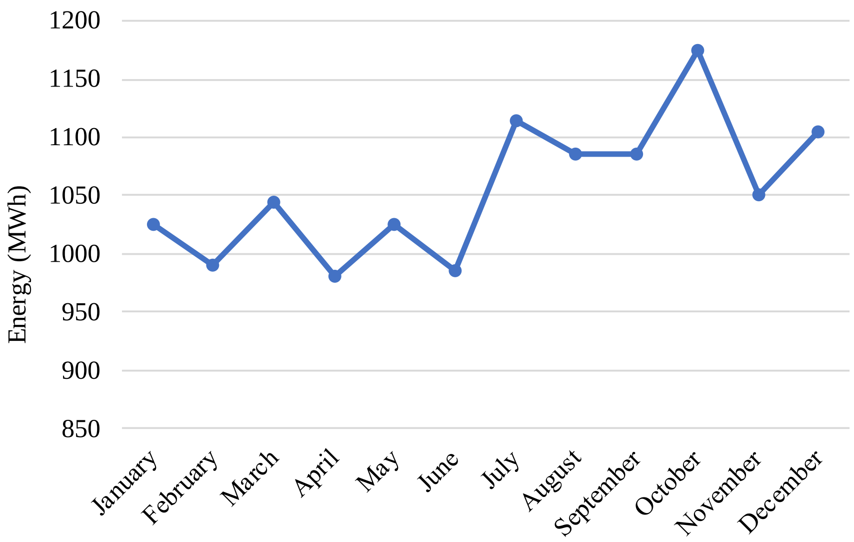


Figure 36 Monthly electricity consumption in Graciosa in 2015

As for the hourly profiles, Electricidade dos Açores (EDA), which is the main utility in the Azores, proposes every year four characteristic hourly demand profiles; one per season [176]. These profiles are shown in Figure 37 and correspond to the 3rd Wednesday of May, August, October and December representing respectively Spring, Summer, Autumn and Winter.

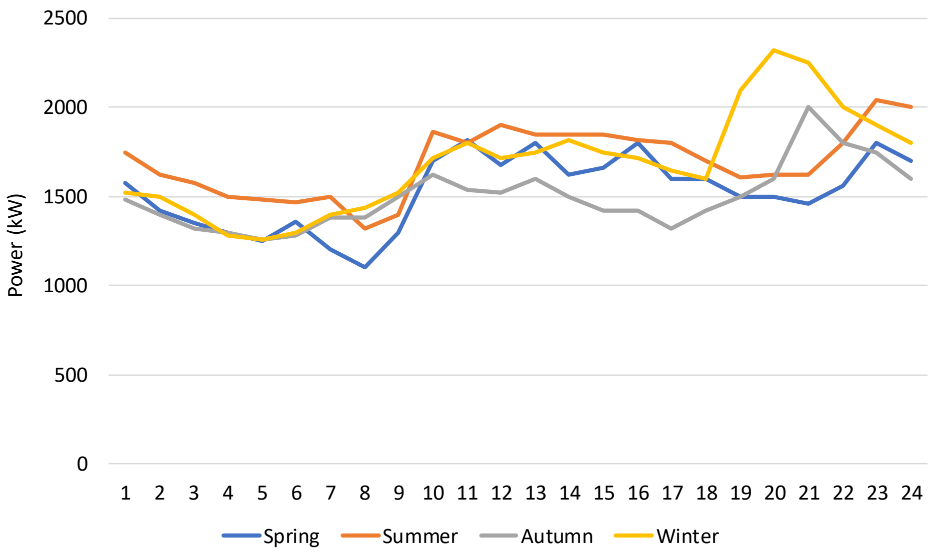


Figure 37 Hourly demand profile for a characteristic day in the season

There are notable differences among the profiles, and for this reason they will be all considered in the analysis of the microgrid operation. A more accurate approach would be to analyze the patterns of the load using an annual time series; however, this information is not available, and so the seasonal profiles will be used in the simulation phase.

On the generation side, the Graciosa microgrid originally includedonly a power plant named "Central Termoeléctrica da Graciosa (CTGR)". This power plant has in total six diesel generators, one with a maximum power of 810kW, three with 600KW and two with 1000 kW, yielding a total power of 4610kW, which is enough to cover the peak demand at any time.

Diesel generators are typically characterized by its excellent ramping capability and short start-up time and are optimal for small isolated islands; however, diesel generation has several drawbacks: first, diesel must be shipped to the island making the cost of energy very high and second, the combustion of diesel produces several pollutant emissions that could be very harmful for the environment and human life. For this reason, from 2005, the possibilityof combining a Battery Energy Storage System (BESS) with renewable energy generation from wind and solar resources started to be investigated. This idea was consolidated in 2015 when Leclanché [177] and Younicos [178], two powerful companies in the microgrid sector, announced the construction of a microgrid consisting of the three elements already mentioned, keeping the diesel generators as a backup. The aim was to reduce the highly pollutant diesel generation and the cost of electricity in the island.

The project was expected to be in operation in summer 2017, however, in the last report from EDA (December 2017) [179] the electricity in the island was still produced by diesel. There is evidence that the system is already built, as seen in Figure 38, but not yet in operation.

Figure 38 Graciosa PV panels, battery building and diesel power plant (courtesy of [178])

For the applicative case study of this thesis, the technical characteristics of the Graciosa microgrid have been mostly obtained from Stenzel et al. [180], where the authors present a life cycle assessment of the generation in Graciosa using the real data from the microgrid. The microgrid is composed by 4000 250W-polycrystalline solar PV panels that yield a total power of 1MW, 5 Enercon E44 wind turbines with a hub height of 55m and 900kW yielding a total power of 4.5MW and, finally, a Lithium-ion battery with 45600 cells with Lithium Titanate cathodes and Lithium Cobalt anodes amounting a total energy of 3.2MWh and maximum power of 6MW.

For the diesel generator, there is no information available about the heat rate function - therefore an assumption had to be made. Because of its similarities and data availability, the heat rate function from the 12 MW-diesel generators inside Candelaria power plant in Tenerife (Canary Islands, Spain) was used [181]. Apart from this, the hourly time series of solar PV and wind were obtained from [182] using the exact location of the corresponding power plants, which is shown in Figure 39. From this data a statistical inference will be made to obtain a set of characteristic daily profiles and indicators to be used in the simulation phase.

Figure 39 Location of the generation in Graciosa Island

5 Simulation and Results

5.1 Comparison of Performance of both Algorithms

The GA and PSO algorithms have several variants, some of which were implemented in the tool. These variants can be found in evolutionary algorithms research papers in which authors usually present the new variant and test it over a set of functions to finally compare it with other variants of the same algorithm or even other algorithms. In these type of research papers, it can be observed that authors usually present the results in order to make the points they want (e.g. to show that the proposed variant is more efficient that all other algorithms selected for comparison) and are sometime not honest or unclear with the reader [17]. Additionally, in optimization there is a well-known theorem called the “No Free Lunch Theorem” which states that all optimization algorithms perform equally well when averaged over all possible problems [183]. This theorem shows that no optimization algorithm is better than any other and that depending on the specific problem to solve one might be better than other. It is therefore an evidence that it is not the best idea to select a specific GA or PSO variant based on literature reviews. The best way to deal with this algorithm selection process is to carry out real tests and use the results to calculate a set of performance indicators that will highlight the superiority of some algorithms over the others. As a result, the best-performing PSO variant and the best-performing GA variant for the specific problem proposed will be discovered.

Before carrying out these tests, it is necessary to define: (1) the performance indicators that will serve to compare the variants, (2) the optimization problem settings (microgrid settings in this case) and (3) the algorithms’ settings that all the simulations will have in common. The performance indicators that will be used are the average cost, the best cost and the average simulation time. Other possible indicators could be the worst cost or the standard deviation. The optimization problem will be the real microgrid case study developed for this dissertation (Graciosa Island). The generation mix of Graciosa will be considered as it is, using the average demand profile, average solar PV and wind profiles and considering degradation of the battery storage system. This will be called ‘base case’ scenario in the next section.

As for the simulation settings, since evolutionary algorithms are characterized by stochasticity, it is necessary to do more than one run to obtain consistent results. In literature, there is no conventional number of run, but the most common is from 10 to 50 runs [46][47] (some authors go to 1000 [39]) For this reason, the number of runs chosen was 30, each one with a different random number seed. Each run will have a population of 100 individuals and the total number of generations per run will be 300, that is when the algorithm has almost converged. Apart from this there will be specific settings per algorithm that will be presented in the following sections.

5.1.1 Genetic Algorithm Settings

As already explained, the genetic algorithm is mainly characterized by three operators: selection, crossover and mutation. In the tool it is possible to select among four types of selection, nine types of crossover and four types of mutation. Since crossover is the main operator of the genetic algorithm and doing all possible combinations of the three operators would yield a vast number of variants, it has been decided to fix the selection and mutation operators. The GA chosen practices tournament selection and non-uniform mutation because it is the only type of mutation that performs exploration at the beginning and exploitation at the end. In addition, there is elitism with one elite and the genetic algorithm is adaptive, this is, the probability of selection and mutation varies with the number of generations. The GA variants depend therefore on the crossover operator where 6 (out of 9) different possibilities were tested. These are enumerated in Table 5-1.

Table 5-1 Genetic Algorithm Variants considered

Variant Name	Description
Single-point	Classic single-point crossover
BLX-0.5	Blended crossover with alpha equal to 0.5
SBX	Simulated binary crossover
Max min arithmetical	Max-min arithmetical crossover with alpha equal to 0.5
FCB Logical	Fuzzy-connectives based crossover. Logical fuzzy connectives family
FCB Hamacher	Fuzzy-connectives based crossover. Hamacher fuzzy connectives family

5.1.2 Particle Swarm Optimization Settings

In Particle Swarm Optimization the equivalent to crossover in GA is the velocity update equation and the variants of PSO usually depend on modifications to this equation or its parameters. In this case, the common configuration of the algorithm is a gbest neighborhood, elitism with one elite and a non-uniform mutation operator that avoids algorithm stagnation. The variants considered are shown in Table 5-2.

Table 5-2 Particle Swarm Optimization variants considered

Variant Name	Description
Constricted PSO	PSO with a constriction factor
DIPS	Decremental Inertia Weight PSO
DIPS-TVAC	Decremental Inertia Weight PSO with Time Varying Acceleration Coefficients
HPSO-TVAC	Self-organizing hierarchical PSO with Time Varying Acceleration Coefficients

5.1.3 Results

Considering the features mentioned above, Table 5-3 shows the performance indicators for each of the variants studied.

Table 5-3 Comparison of performance of different algorithms and their variants for the base case scenario

Genetic Algorithm	**Minimum Cost**	**Average Cost**	**Average Simulation Time**
Single-point	2049.957	2671.740	36.362
BLX-0.5	1465.328	1892.409	35.576
SBX	2138.160	2871.489	37.319
Max Min Arithmetical	1832.123	2271.208	67.419
FCB Logical	1728.115	2216.255	72.530
FCB Hamacher	1944.220	2607.198	69.926

Particle Swarm Optimization			
Constricted PSO	1433.764	1892.409	35.576
DIPS	1697.179	2373.845	34.166
DIPS-TVAC	1236.504	1612.688	38.560
HPSO-TVAC	2042.060	5080.043	34.195

The first observation that can be made is that PSO performs in general better than GA for the specific problem considered. From the PSO variants considered the best performing one is clearly the DIPS-TVAC since it has both the lowest average cost and the lowest minimum cost among all the variants indicating that results are very similar in all the runs. However, this variant is not the fastest one as it can be seen in the simulation time column. If simulation speed is the preference, then the best algorithm would be DIPS or HPSO-TVAC, but accuracy would be sacrificed to only gain one second. Regarding GA, the best performing variant is clearly BLX-0.5, since it almost reaches the minimum achieve by DIPS-TVAC or constricted PSO and has an average cost even lower than some of the other PSO variants. This variant has an average simulation time of 35.576 seconds being almost the same as PSO.

Thus, it can be said that if PSO is to be used to solve the case study, the right choice would be to use the DIPS-TVAC or even the constricted PSO, while if GA is preferred (it is typically a more robust method) then BLX-0.5 will best possible variant to use. These statements, however, apply only for the specific problem consider and not for every optimization problem in the world (remember the No Free Lunch Theorem). A new performance assessment would be needed in case of doing meaningful changes to the problem formulation.

In the next section, where different Graciosa scenarios will be solved, DIPS-TVAC will be the algorithm chosen, since it achieves the best tradeoff between simulation time and solution accuracy.

5.2 Case Study Results

This section will first present the different results obtained for a base case and a set of demand an VRE scenarios (summarized in Annex B) in Graciosa island. Then, with these preliminary results, the effect of battery degradation modeling onto the simulation will be analyzed and some possible measures to improve the performance of the existing microgrid will be tested.

5.2.1 Base Case

The base case will be the first scenario simulated. This scenario includes Graciosa's power system as it is nowadays. The demand profile as well as the PV and wind generation profiles will be average profiles obtained from the yearly time series. These are shown in Figure 40. Here the wind profile is flatter than normal, but this is something normal in that location according to MERRA-2 data provided by NASA. The goal of this scenario is to identify possible generation adequacy issues in the island and to have a base of comparison for the set of scenarios that will be analyzed in the following sections.

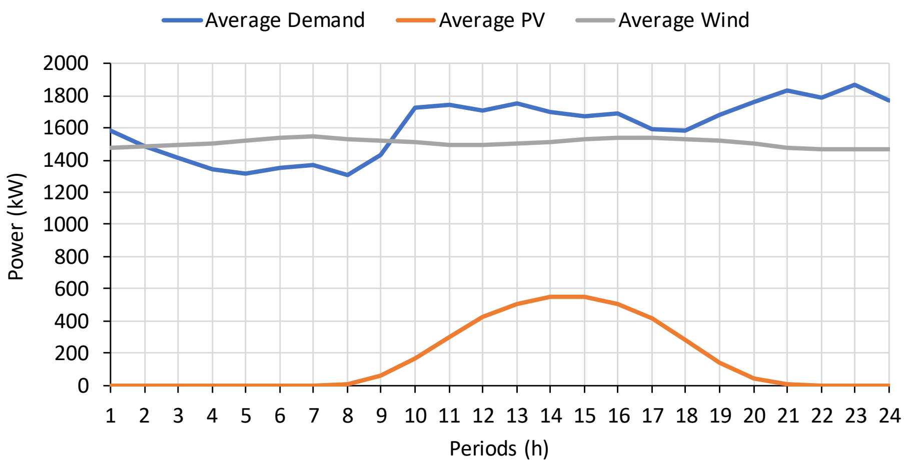


Figure 40 Average demand, wind and solar PV profiles in the base case scenario

With this input data and using the Decremental Inertia Weight Particle Swarm Optimization with Time Varying Acceleration Coefficients (DIPS-TVAC) algorithm, which was proven to be the most efficient when solving this specific problem, ten runs of the case study were made. This is because the stochasticity of the algorithms will yield different results in different runs and therefore it is necessary to do more than one to get valid results [17]. From these ten runs, the one with the minimum total costs was chosen. Figure 41 shows the dispatch obtained in this run and Figure 42 the total generation of the base case scenario.

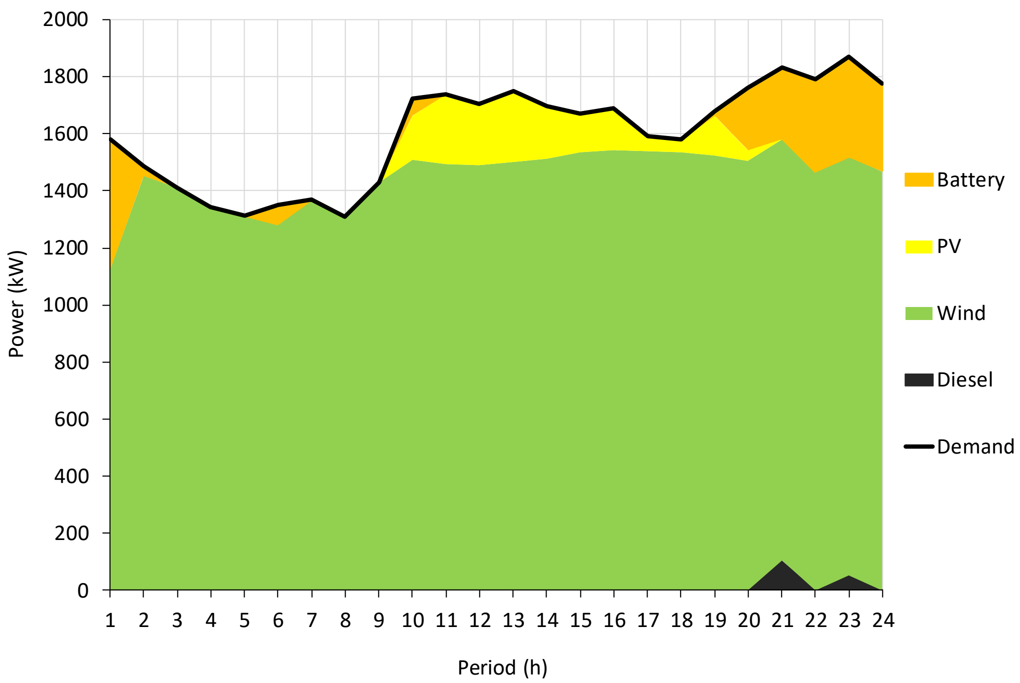


Figure 41 Optimal dispatch for the base case scenario

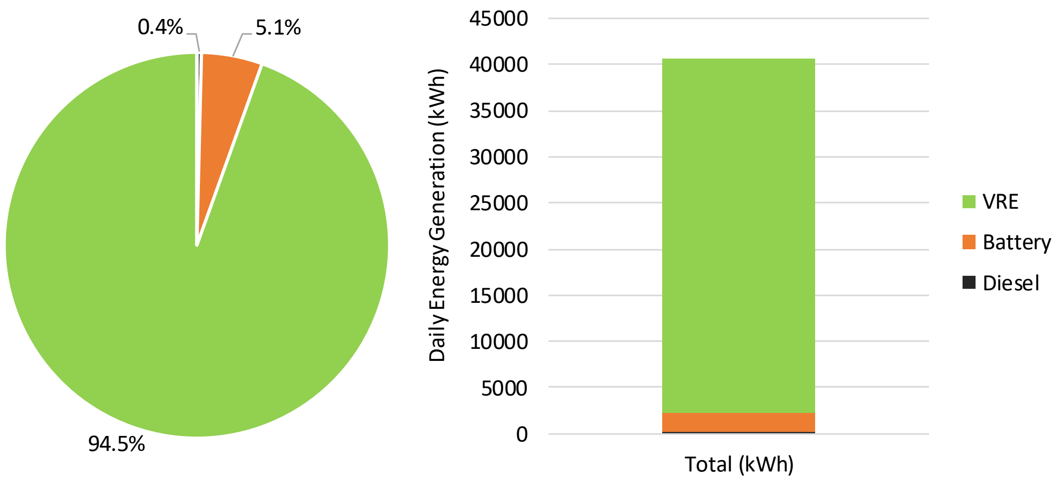


Figure 42 Total generation in the base case scenario

In Figure 41 it can be observed that there are not generation adequacy issues in the system, that is to say, there is enough generation capacity to cover the entire demand in the base case. Furthermore, the system can run almost entirely with renewable energy and the battery using just a 0.4% of diesel generation. As for the battery operation, Figure 43 shows the charging and discharging behavior of the battery throughout the day,while Figure 44 shows the sources from which the battery charges during the day

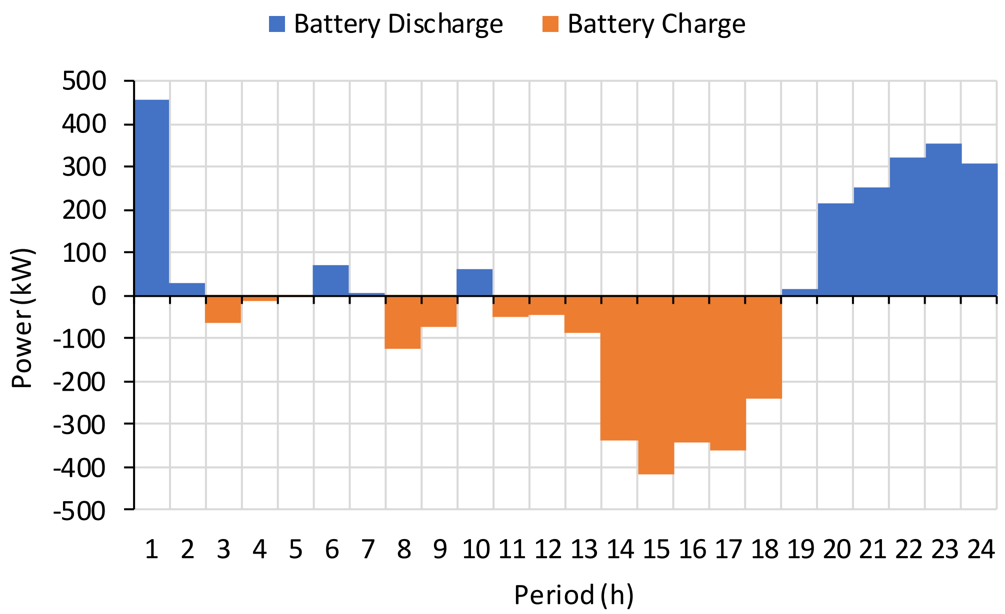


Figure 43 Battery charging and discharging profile in the base case scenario

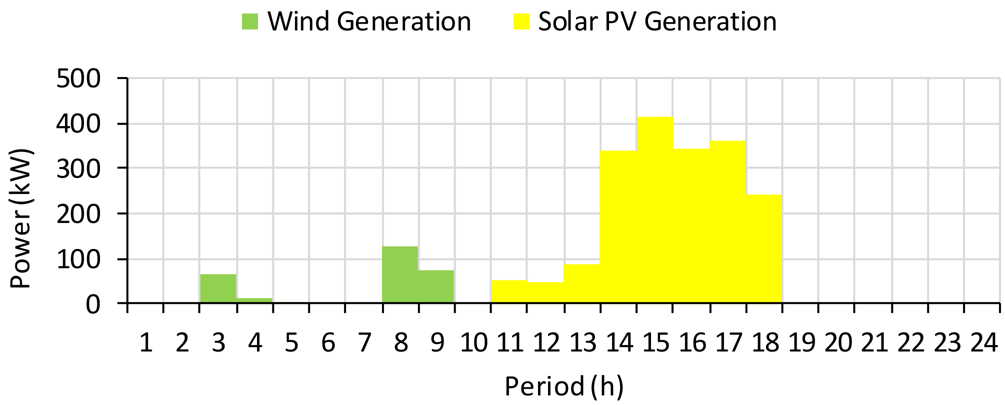


Figure 44 Resources feeding the battery over the 24 hours, in the base case.

As shown in Figure 43, the battery discharges at the beginning of the day not only to avoid diesel generation, but also to avoid unnecessary degradation costs and stay at a reasonable SoC. Then, it discharges at the end of the day to avoid excessive diesel generation being unable to shift entirely this costly generation. The battery charges using only renewable energy sources, whenever they are enough. This entails several benefits to the system (e.g. cost reduction, lower pollutant emissions, etc…) and suggests the usefulness of having a BS into a microgrid to support the RESs. There could be some particular cases in which the renewable energy available is higher than what the battery can absorb. In this case, if the wind turbine or PV panel has what is referred to as curtailment mechanism [184], the resource would be curtailed. This mechanism, however, is not within the scope ofthis dissertation and so was not considered in the simulations.

As for the total costs incurred in this scenario, these are shown in Table 5-4. It is possible to notice that the cost of diesel generation is the main component of total system costs.

Table 5-4 Total system costs in the base case scenario

€	Diesel Cost	Degradation Cost	Total Cost
Base Case	849.77	291.95	1141.72

The results in this scenario are average results and cannot be used to fully validate the operation of the system. For this reason, a complete set of demand and wind scenarios will be analyzed in the following sections.

5.2.2 Demand Scenarios

As presented in Section 4.4.2, the utility from Azores uses four different demand scenarios that correspond to the seasons of the year. From one season to another the demand varies significantly and, thus, it is convenient to study the influence of all these demand profiles on the microgrid optimal operation. The PV solar and wind profiles will be the average profiles as in the base case, but the demand profile will be changed in every simulation. From Figure 45 until Figure 48 the dispatch obtained for each season is shown in the following order: Spring, Summer, Autumn and Winter.

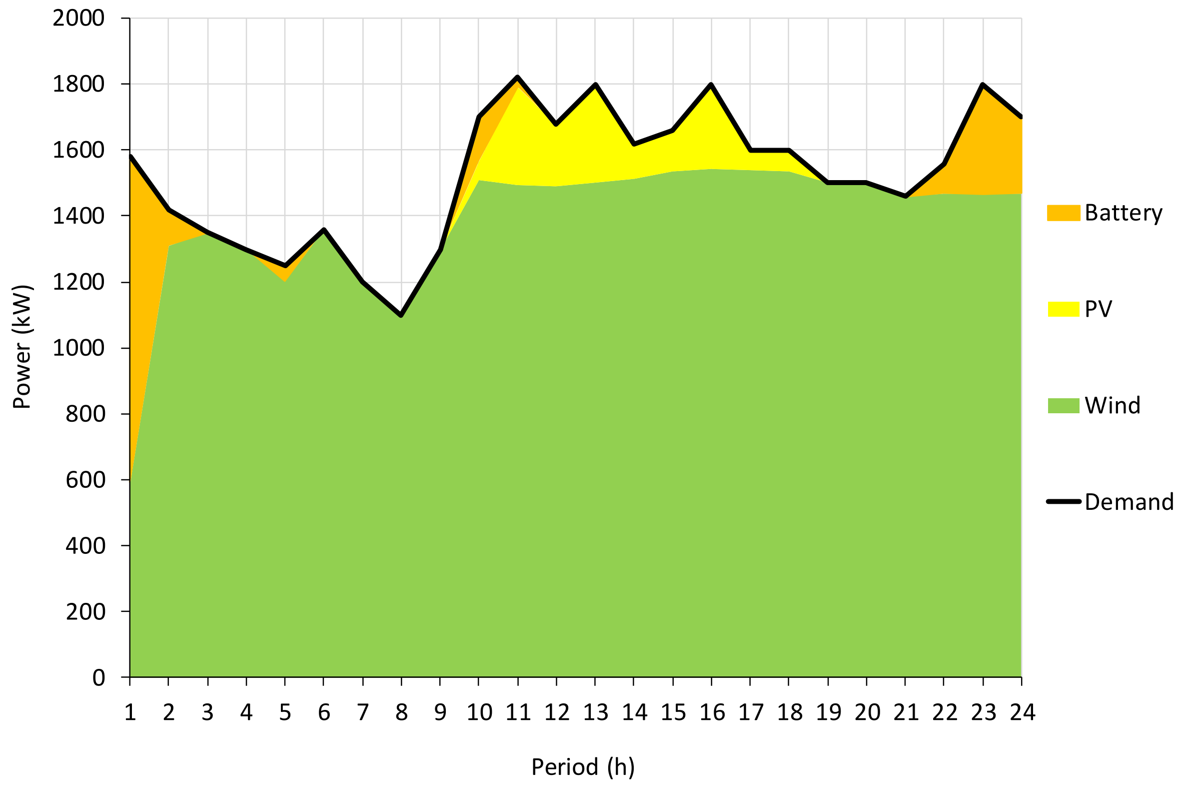


Figure 45 Optimal dispatch for the characteristic Spring demand in Graciosa

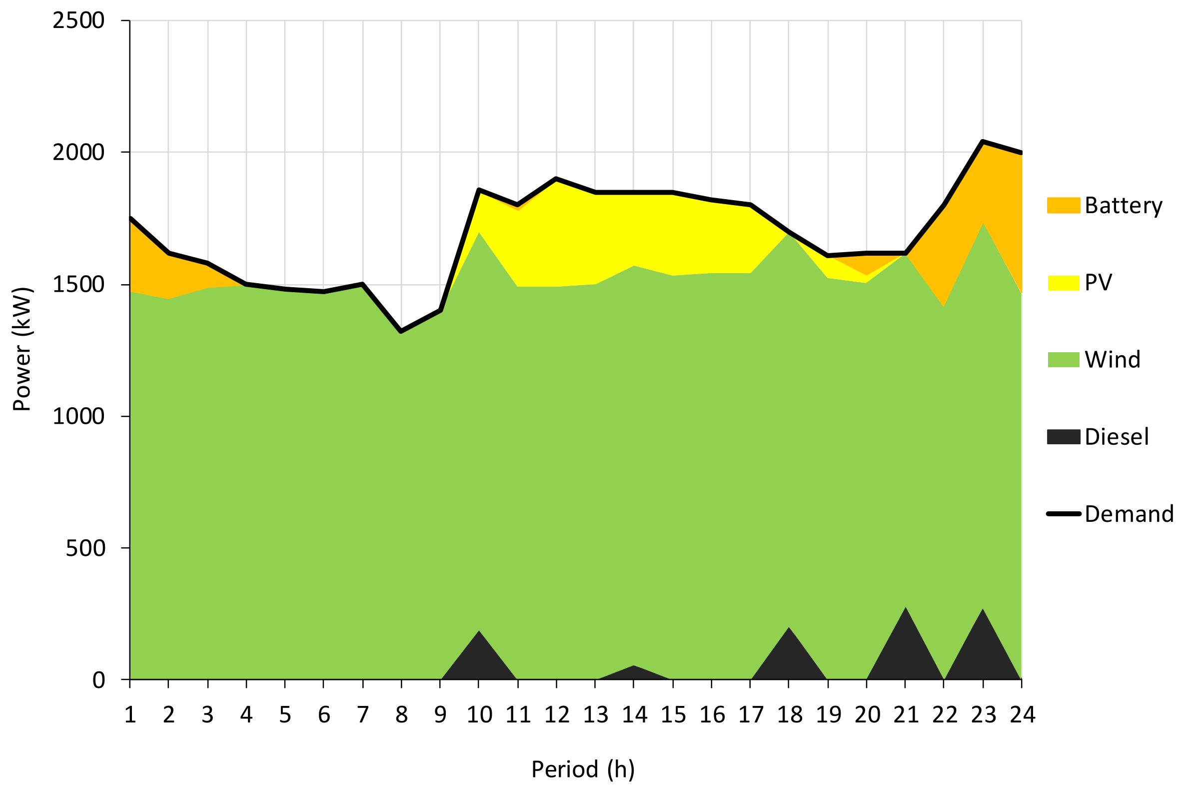


Figure 46 Optimal dispatch for the characteristic Summer demand in Graciosa

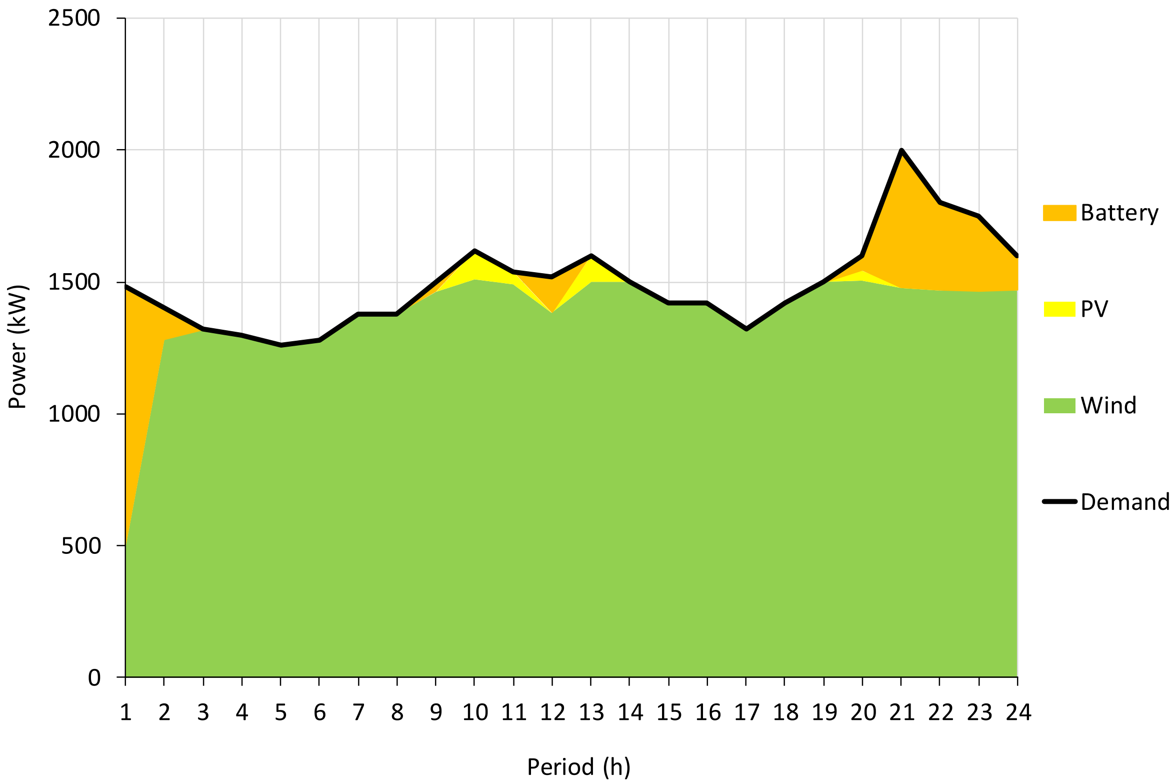


Figure 47 Optimal dispatch for the characteristic Autumn demand in Graciosa

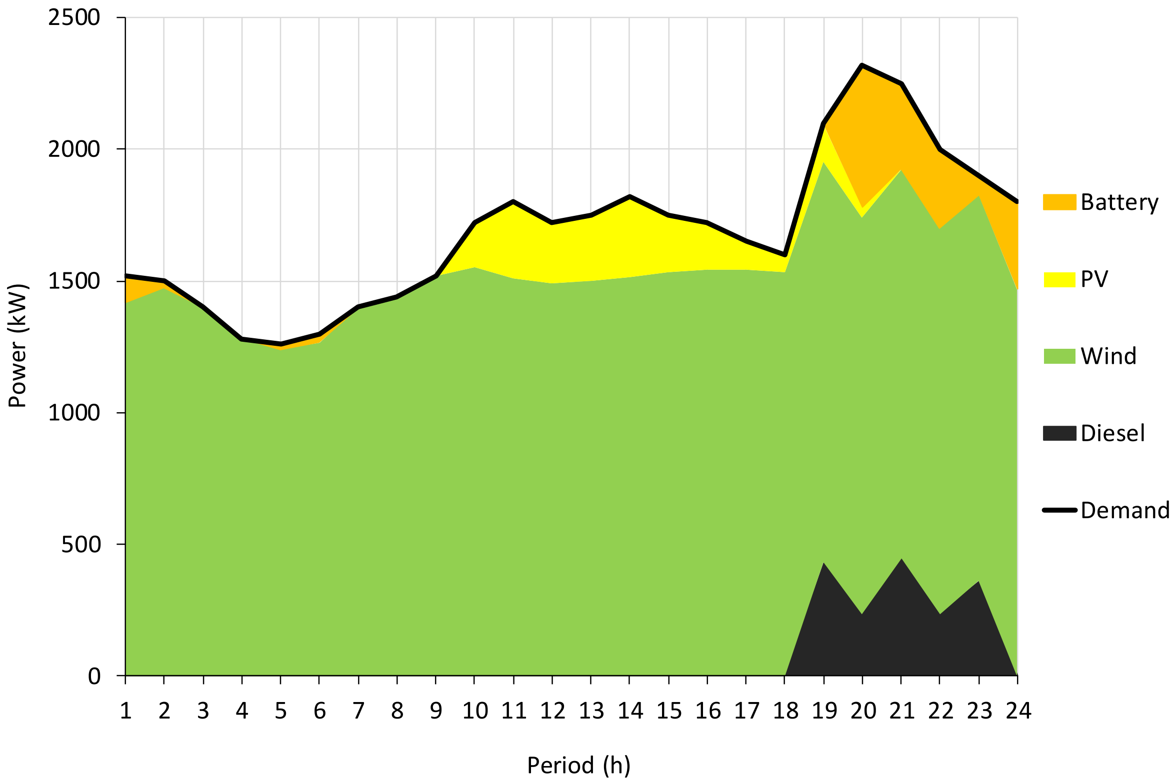


Figure 48 Optimal dispatch for the characteristic Winter demand in Graciosa

This leads to the total generation shown in Figure 49.

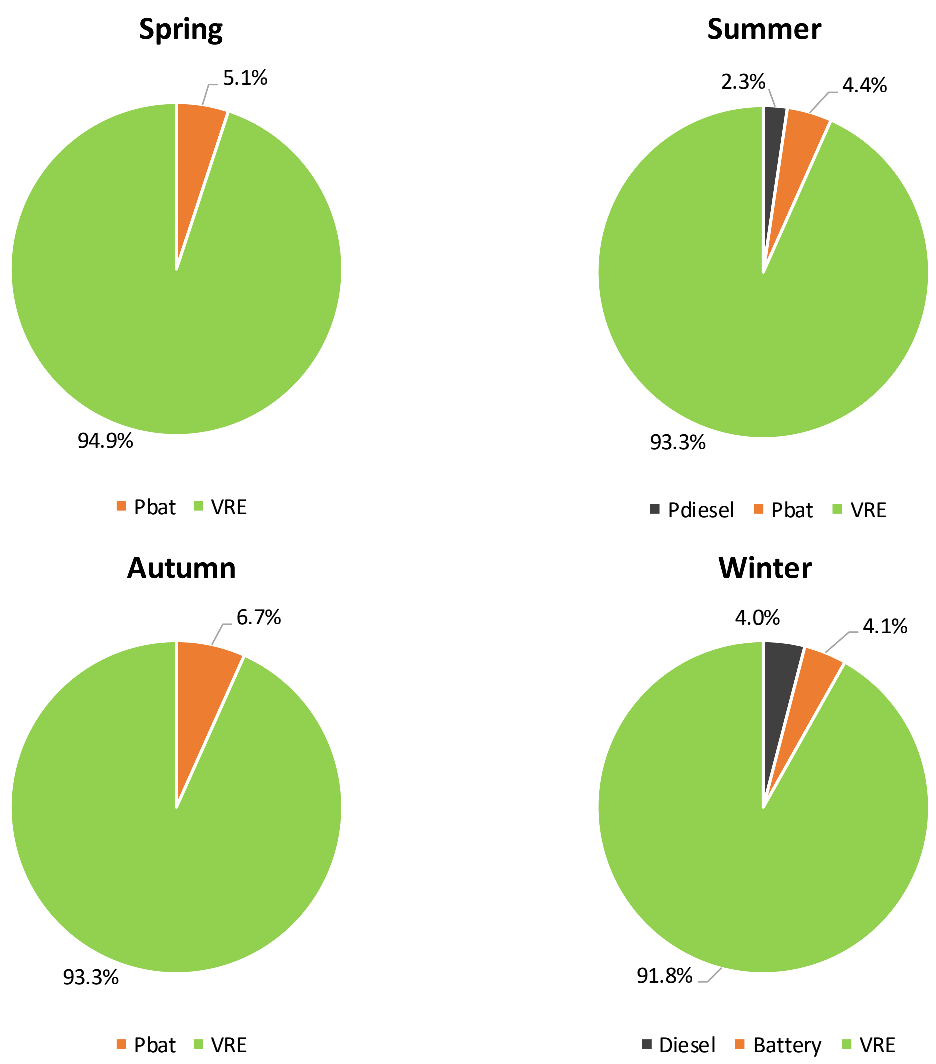


Figure 49 Total generation for each demand scenario in Graciosa

During spring and autumn, when the demand is lower, the system can run entirely with renewable energy using the battery storage system. However, in winter and summer, when the demand grows because of the tourism (summer) or because of the cold weather (winter), the diesel generator has to be sometimes dispatched, increasing the total system costs. This, justifies the permanence of the diesel generator in the island as a backup generator. As for the battery, the charging and discharging profiles for each season are shown in Figure 50.

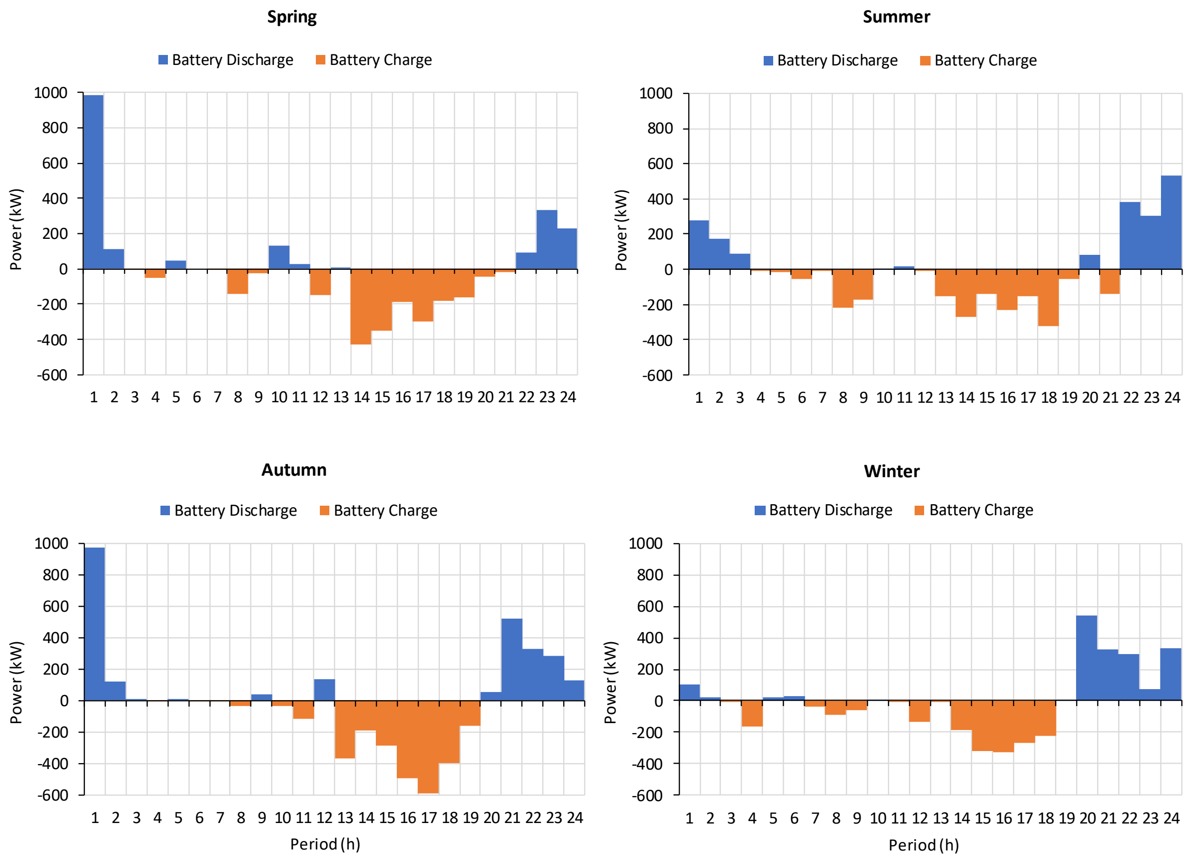


Figure 50 Battery charging and discharging profiles for each demand scenario

In the sample day taken for summer and winter, the battery has operate almost continuosly, to avoid as much diesel generation as possible. In these days, cyclic ageing is higher than in the sample days taken for spring and autumn, when the battery is not needed that much. In these two seasons, the total battery degradation is lower, not only because of the lower cyclic ageing but also because of lower calendric ageing (when the battery is idle is at a low SoC, reducing this type of degradation). The generation sources used by the battery to charge and store energy are solar PV and wind generation as in the base case scenario. Since this result is very similar to the previous section it will not be represented.

Apart from changes in the operation of the system and in the battery dispatch and degradation, there will be also cost differences in each scenario, caused mainly by the amount of diesel generation but also by the battery degradation costs incurred. Table 5-5 summarizes the total costs of the system in each demand scenario.

Table 5-5 Total system's costs in different demand scenarios

€	Spring	Summer	Autumn	Winter
Diesel Cost	0	1081.31	0	1379.44
Degradation Cost	252.05	282.24	267.74	306.72
Total Cost	**252.05**	**1363.56**	**267.74**	**1686.17**

The most expensive scenarios are those in which the diesel generator must be on in some periods: summer and winter scenarios. Here, the degradation costs are also higher because, as already explained, the battery must be continuously in operation to avoid excess diesel generation. Because of this, the battery increases its cycling rate and maintains a higher SoC suffering both cyclic and calendric degradation.

5.2.3 Wind Scenarios

Apart from demand scenarios, the most critical VRE scenarios and its influence on the system operability were also analyzed. In Graciosa there are 4.5MW of wind and 1MW of solar PV installed, making wind more influential than solar PV in the island. Apart from this, wind has typically a higher variability than solar PV, which makes even more interesting the analysis of wind scenarios in the island. For this reason, three different wind scenarios were analyzed while maintaining an average demand profile. These are: **highest energy scenario**, which uses the wind profile from the day with highest wind energy production**, lowest energy scenario**, which uses the wind profile from the day with the lowest wind energy production and **highest variability scenario**, which uses the wind profile from the day with higher wind variability. Using this data, the system dispatch in each scenario is shown from Figure 51 to Figure 53.

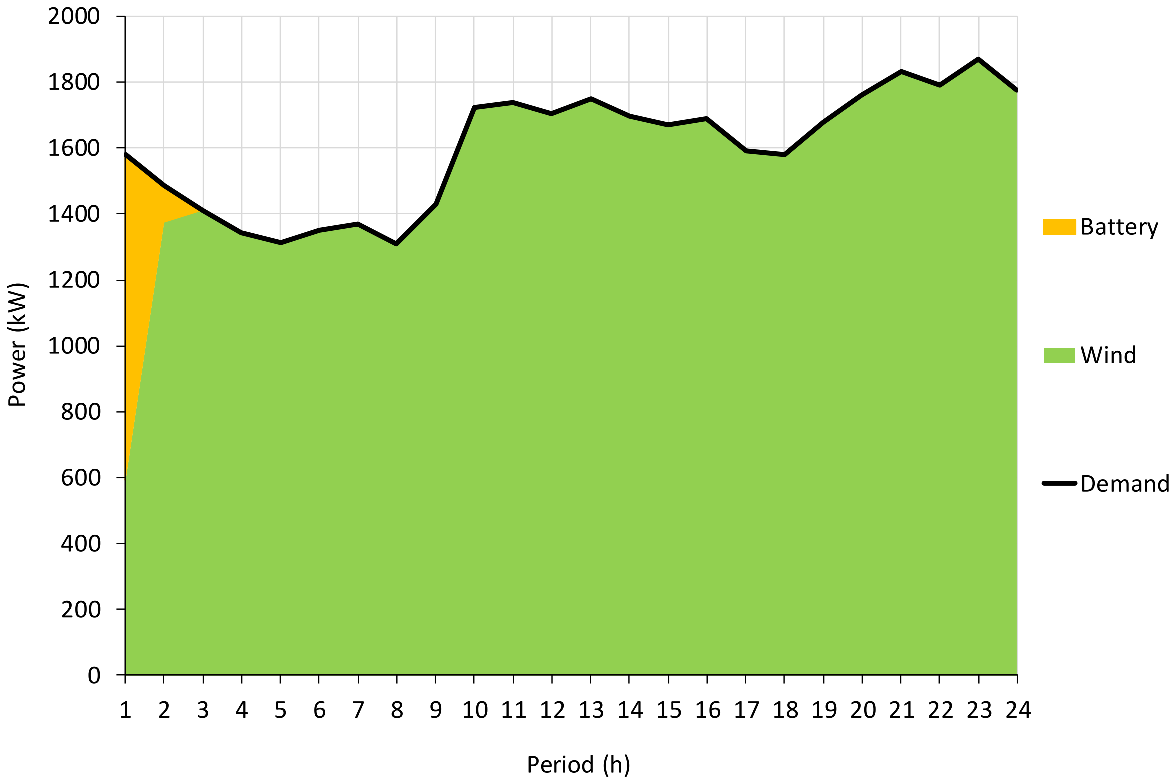


Figure 51 Optimal dispatch for the highest wind scenario

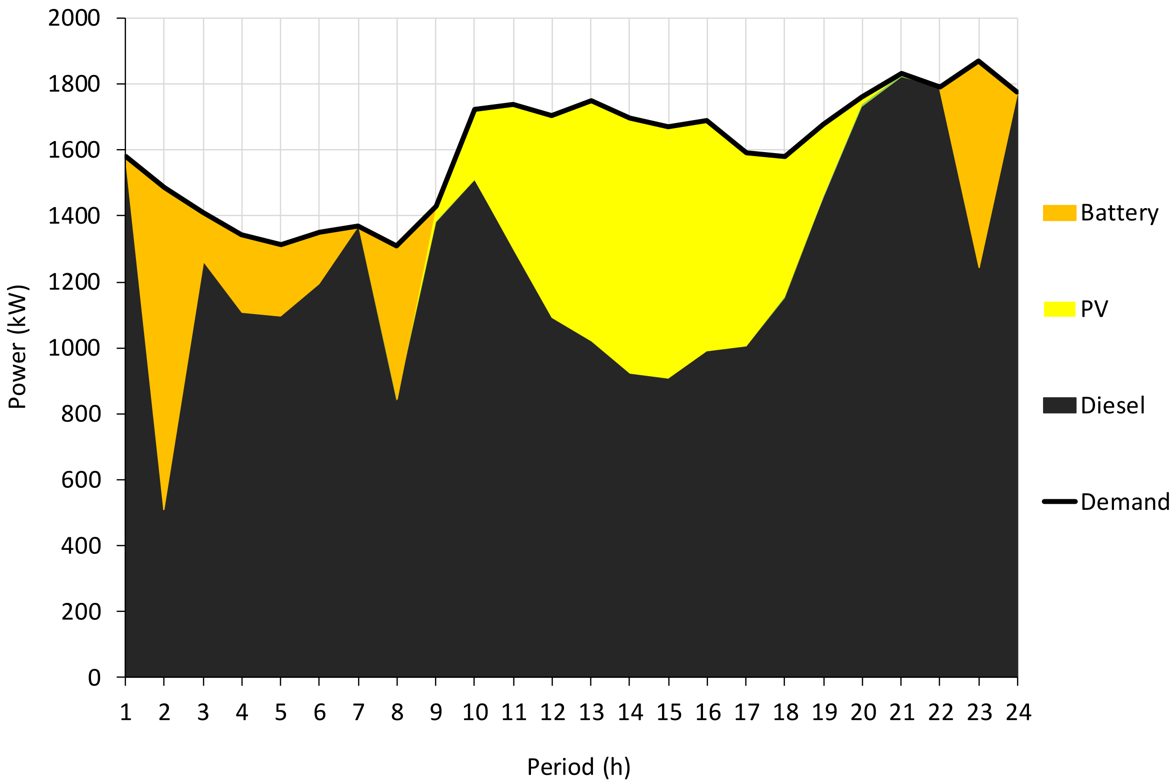


Figure 52 Optimal dispatch for the lowest wind scenario

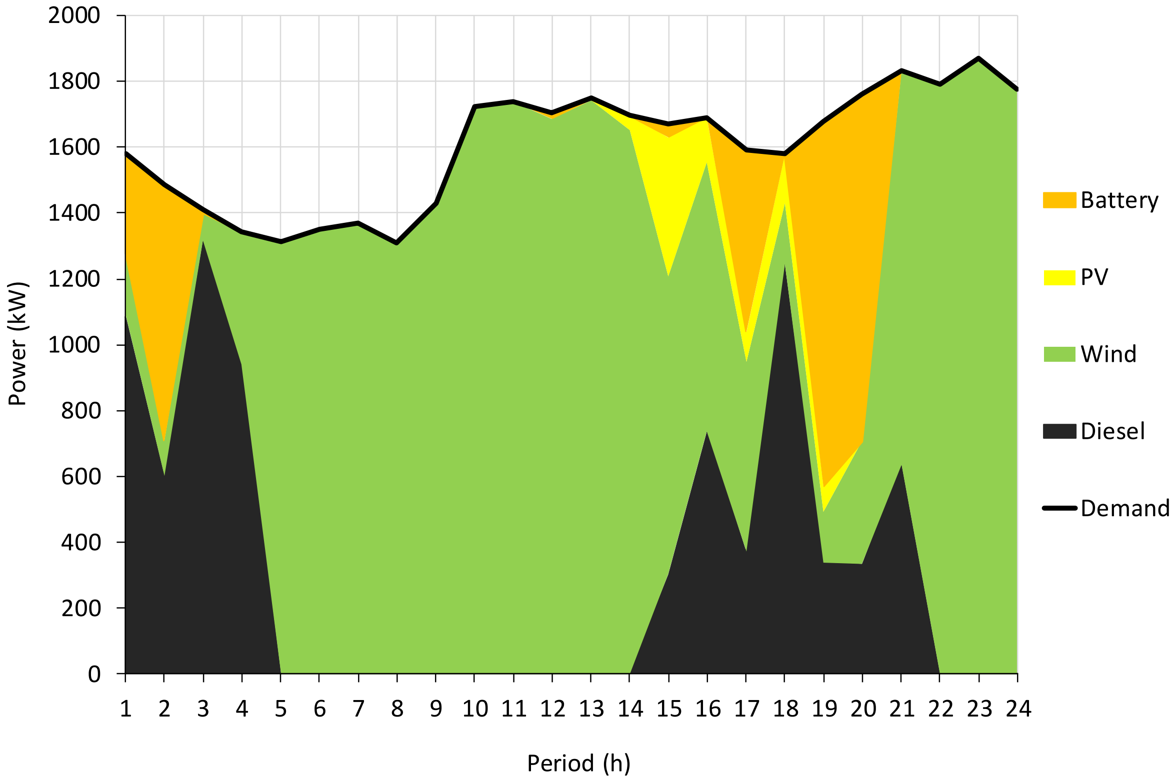


Figure 53 Optimal dispatch for the highest variability scenario

While the total generation is shown in Figure 54.

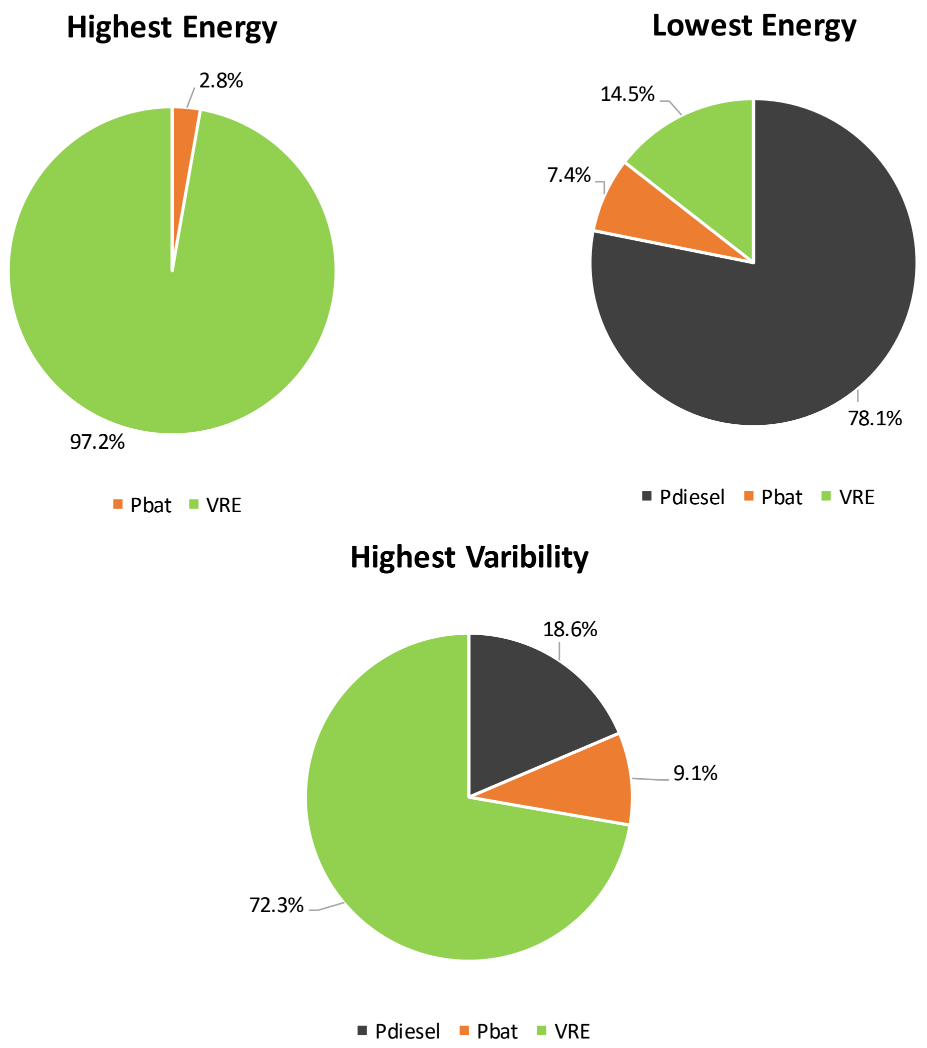


Figure 54 Total generation for different wind scenarios

First, in the highest wind energy scenario the wind profile is so flat because the wind speed in that location for that day was very constant according to MERRA-2 data provided by NASA. Thanks to this, in this scenario, the system can cover the demand only with renewable energy sources with no need of using diesel generator. The battery is just used in the first period to maintain a low SoC and minimize ageing as much as possible. In Figure 54, wind generation amounts for a 97.2% of the total while 2.8% of the generation produced by the battery storage. In the lowest wind energy scenario, the results are virtually the opposite to the highest wind energy scenario. Here, the production of wind is so low that the diesel generator must be generating all the time. This, results in an 83% of total diesel generation that will increase the cost of the system and the amount of pollutant emissions. As for the battery, it discharges in periods in which the highest costs of diesel generation are avoided. In the highest variability scenario there are periods with high wind penetration and periods with low wind penetration, and because of this the battery would be really useful for this scenario. To observe this, Figure 55 represents the charging and discharging battery profile of the highest variability scenario, while Figure 56 shows from where comes the generation stored by the battery in the simulation period.

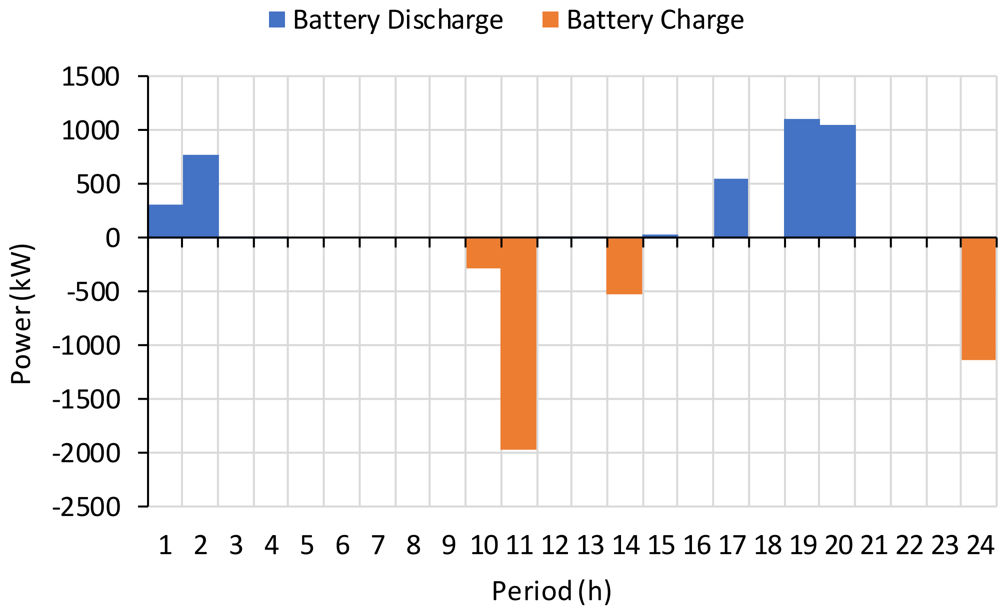


Figure 55 Battery charging and discharging profile for the highest wind variability scenario

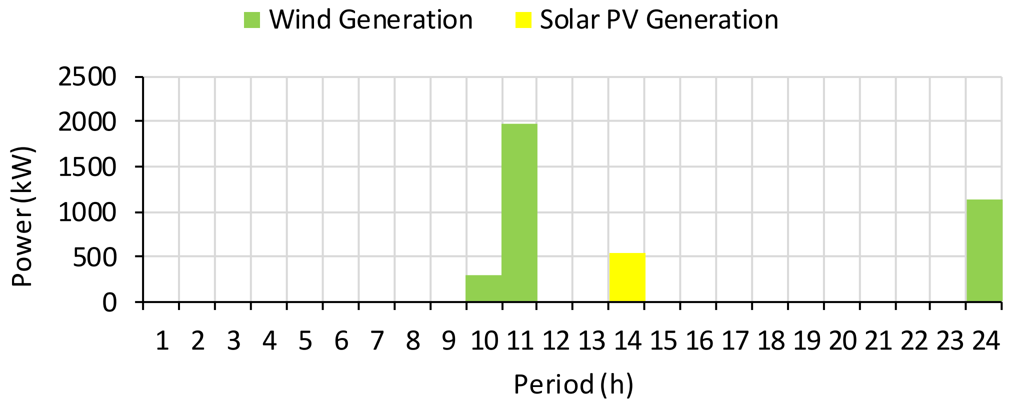


Figure 56 Origin from the energy stored by the battery in the base case scenario

The battery charges when the wind penetration is high absorbing zero cost energy that would be lost otherwise. Then, it discharges when wind generation is low, avoiding excessive diesel generation. Consequently, diesel generation amounts only for a 19% of the total generation.

Finally, Table 5-6 shows the total cost of the system in the wind scenarios analyzed and compares them to the base case scenario where the wind profile is the average.

Table 5-6 Total system's costs in different wind scenarios

€	Highest	Lowest	Variability	Base Case
Diesel Cost	0	27888.34	5114.12	849.77
Degradation Cost	233.79	236.44	299.06	291.95
Total Cost	**233.79**	**28124.78**	**5413.19**	**1141.72**

The costliest scenario is the lowest wind energy scenario due to the amount of diesel generation necessary to cover the entire demand. This cost is reduced in the highest variability scenario where the diesel generation is much lower, and it is minimized in the highest wind energy scenario, where there are only degradation costs from the battery because of calendric ageing (the battery is idle and there cannot be cyclic ageing in that scenario).

5.2.4 Impact of Battery Degradation in the simulation

In the economic dispatch problem formulation already presented, the most innovative part is the inclusion of battery degradation as a cost in the objective function. In the previous sections, the effect of modeling degradation in the economic dispatch problem has been briefly presented to explain, for instance, why the battery sometimes seeks for low SoC or why it minimizes cycling. In this section, the effect of degradation on the model will be explained more in depth by analyzing it in the base case scenario.

The battery installed in Graciosa is a lithium-ion battery manufactured by Leclanché with LTO as a cathode and LCO as an anode. In the website from Leclanché, their LTO battery datasheet can be easily obtained (Appendix C). Here it can be seen that the cyclic lifetime is 15.000 cycles at 100% DoD, and the nominal calendric lifetime is 20 years at 23ºC. Apart from that, the replacement costs of an LTO battery, which as already explained inn Section 4.2.2 are the same as the storage investment costs, are 1000€/kWh [6]. Using this data, the degradation model proposed in Section 4.2.1 can be modified to represent appropriately the battery installed in Graciosa.

First, the Wöhler curve was shifted upwards from 3000 cycles at 100% DoD to 15.000 cycles at 100% DoD to properly represent the cyclic ageing of the LTO battery. As for the calendric ageing, the temperature of the battery was considered constant (23ºC) and the input data was modified to have 20 years of calendric age at a reference temperature of 23ºC as well. Finally, the effect of the state of charge was maintained as in [170], this is, calendric ageing is increased by a high SoC. This is because when the SoC is high, Li-ion batteries suffer processes such as loss of active surface or electrolyte decomposition which produce ageing. The minimum SoC is set to 15% since it is not realistic to completely discharge the battery, while the maximum SoC was set to 90%.

Using this modified degradation model, the base case study was simulated, and the degradation related variables were analyzed together with the battery throughput. Figure 57 shows the charging and discharging profile, the SoC profile and the DoD variation for the base case scenario with and without degradation. In the case with no degradation considered, the replacement costs of the battery are set to 0€/kWh and therefore there is no implicit degradation in the model.

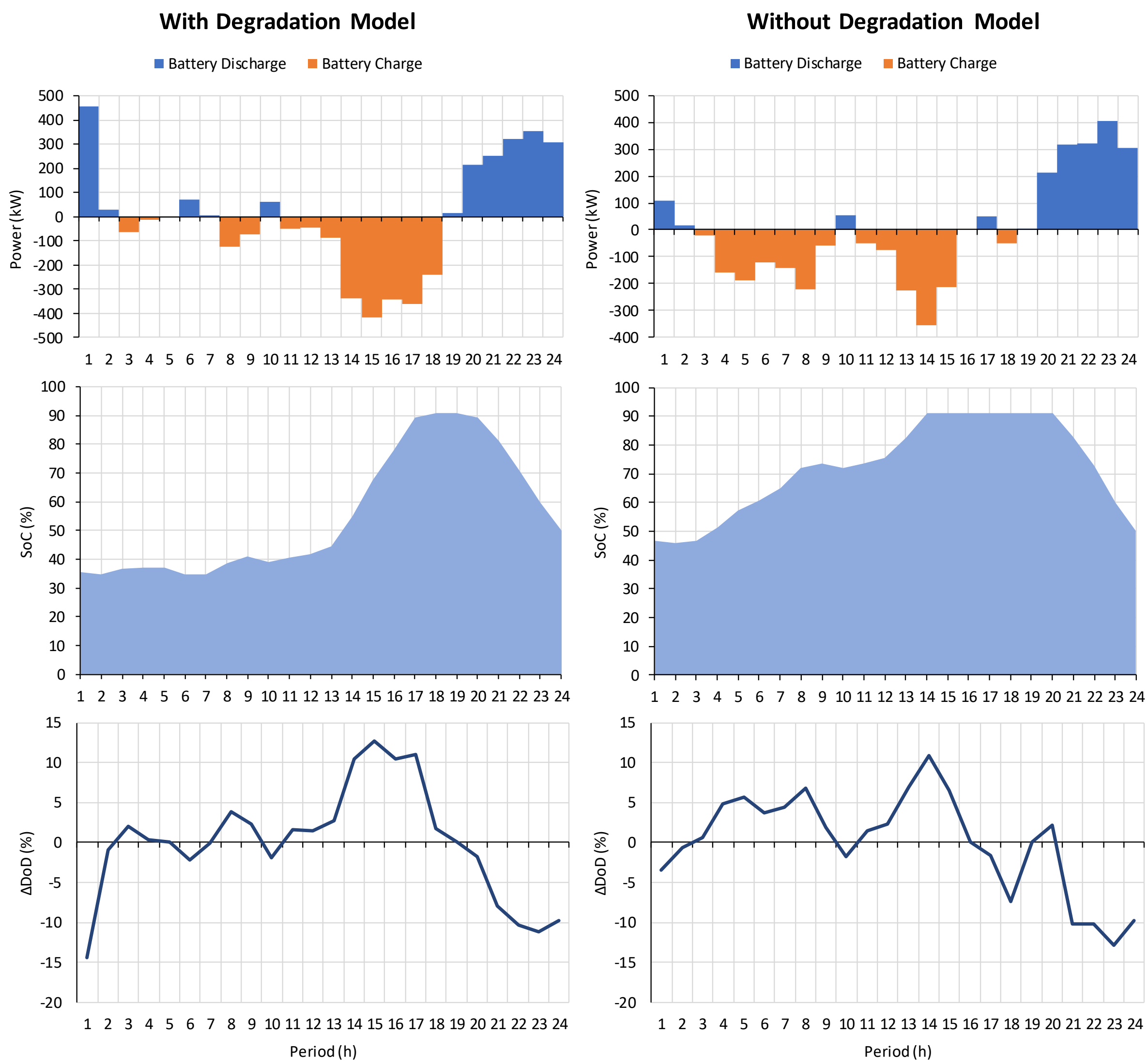


Figure 57 Charging and discharging profile, SoC and ΔDoD for the base case scenario with and without degradation

The calendric ageing is here dependent on the SoC. Although the temperature was also included in the degradation model, it was considered constant and its influence is therefore neglected. In Figure 57, it can be seen that by modeling degradation, the average SoC over the 24-hours is significantly higher than by not considering it. This would increase in the battery ageing speed, increasing replacement costs. On the other hand, cyclic ageing is dependent on the ΔDoD which is directly related with the battery cycles. In the ΔDoD plots it can be observed that in the magnitude of ΔDoD is similar in both scenarios, however, the average ΔDoD (in absolute value) is higher in the scenario that does not consider degradation, which will result in a higher number of cycles and therefore a higher cyclic ageing speed. This will also yield higher replacement costs.

To further analyze how modeling degradation in the economic dispatch problem affects the battery operation, Table 5-7 shows a comparison of scenarios with and without degradation in terms of costs and ageing. Ageing and degradation costs calculation in the simulation without degradation model was made 'a posteriori' using the SoC and ΔDoD time series given by the tool. The total system costs include this calculated degradation cost.

Table 5-7 Comparison of battery ageing for simulations with and without degradation model

Base Case	Average ΔDoD (%)	Average SoC (%)	Total Degradation (p.u.)	Degradation Cost (€)	Total System Costs (€)
With Degradation	5.05	54.92	9.12E-5	291.95	1141.72
Without Degradation	5.20	71.85	1.10E-4	354.53	1180.62

As already explained, the average ΔDoD per hour (related to the average cycles made by the battery per hour) is higher if degradation is not considered in the economic dispatch problem and so does the cyclic ageing. This also occurs with the SoC, being calendric ageing higher if degradation is not modelled. This causes that total system costs (accounting also for degradation costs) are higher in those scenarios in which degradation was not modeled. Thus, to operate the microgrid at the minimum possible cost and increase the life of the battery as much as possible it could be crucial to consider battery ageing as part of the simulation.

5.2.5 Measures to improve the microgrid performance

5.2.5.1 Implementation of a demand response program

To cope with flexibility issues that appear when the penetration of Variable Renewable Energy is high there are not only alternatives in the generation side (e.g. battery storage or flexible thermal generation) but also on the demand side. In this context, the concept of Demand Side Management (DSM) comes to play. In literature, demand side management is typically defined as “a portfolio of measures that improve the energy system in the side of consumption” [3]. Some of these measures could be the improvement of energy efficiency, time of use tariffs, demand response or spinning reserve provision. In this dissertation the focus will be on Demand Response (DR) which is defined as “changes in electric usage by demand-side resources from their normal consumption patterns in response to changes in the price of electricity over time” [185].

To model demand response the formulation presented in Section 4.1.2 will be used. This formulation is based on [186] and consists on limiting the amount of demand response per period to what is called a demand response band. This band depends on the contract established by the load user and the system operator for the controllability of flexible (controllable) loads. There are different ways to estimate a controllable load. In [186], for instance, the author uses Montecarlo and scenario reduction techniques to create an average profile. In this dissertation the estimation of controllable loads will be made based on assumptions on demand composition.

Figure 58 shows the composition of the demand in Graciosa island. Typically, industry, agriculture, public lighting and government buildings do not participate into demand response programs since they require a continuous supply of electricity. Non-domestic load includes shops, office buildings and non-residential areas and in this dissertation will not be considered. Thus, only domestic loads will be able participate into the demand response program from Graciosa.

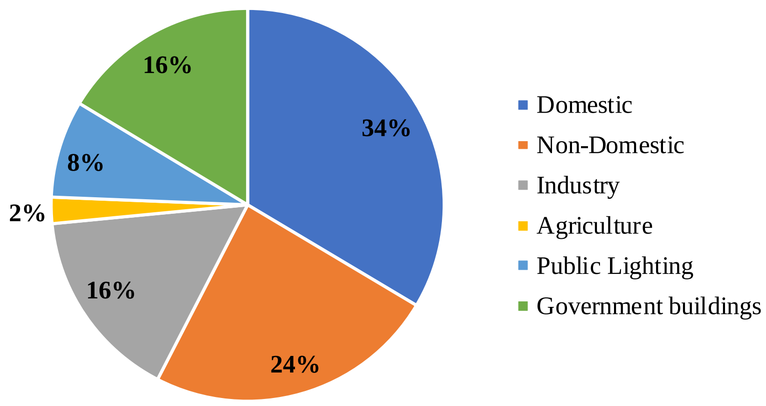


Figure 58 Composition of the load in Graciosa island

The composition of the demand shown in Figure 58 is the average composition of the demand in year 2015. Although the composition of the demand per hour will typically vary, the first assumption regarding demand response will be that the domestic demand is always a 34% of the total. The second assumption will be that every household (with a hired power of 6 kW) have electric heating/cooling (AC), electric water heating (EWI) and a refrigerator. These loads will be called thermal loads and will be the ones able to participate in the demand response program [187]. The electric heating/cooling will be working during the day (7 am-21 pm) and will be heating in winter/autumn and cooling in spring/summer with a power of 2kW. The EWI will be working during the night (22 pm – 6 am) and the refrigerator (0.2kW) will be working 24 hours. This call for two scenarios: day and night demand response.

During the day it will be assumed that 2.2kW (AC+refrigerator) out of 6kW will be controllable, which in percentage would be a 36.6%. Since the domestic loads are only a 34% the total controllable load during the day will be around 12.5%. Since it is not realistic to shed the total controllable load for the whole day only a 50% will be part of the demand response band yielding 6.25% of the total load.

During the night the 34% will correspond entirely to the EWI and the refrigerator and therefore will be directly the percentage of controllable load. Using a similar approach as above, the total demand response band during the night would be 17% of the total demand. Figure 59 shows the demand response band in Graciosa using the winter demand profile and the demand response band data already calculated.

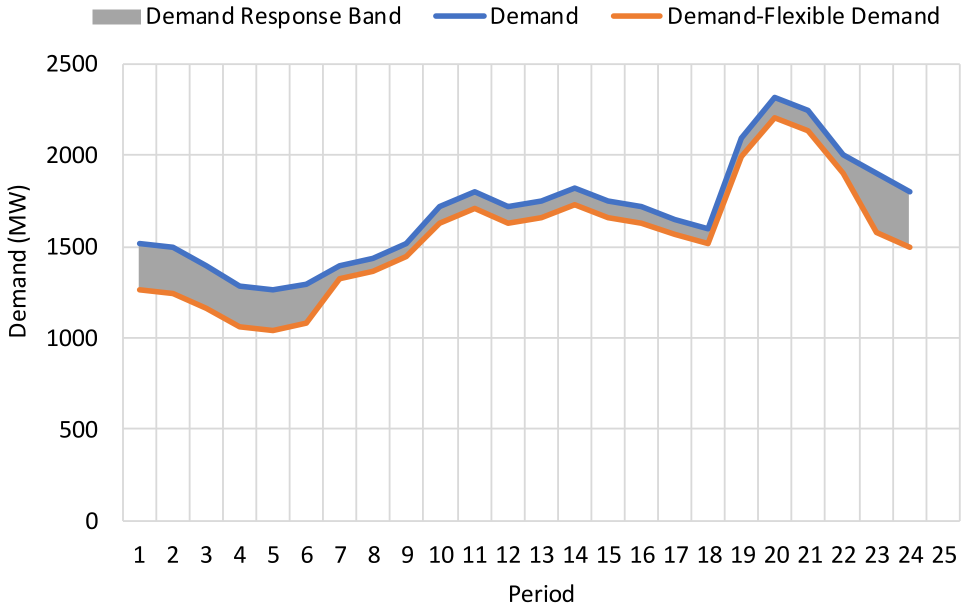


Figure 59 Demand response band in a typical winter day in Graciosa island

Once the demand response band has been estimated one of the previous scenarios can be analyzed including demand response. Since, demand response can be particularly usefu to avoid diesel generation, an interesting scenario to analyze would be the winter demand scenario, where there is a 4% of diesel generation that could be avoided. The optimal dispatch and total generation are shown in Figure 60 and Figure 61.

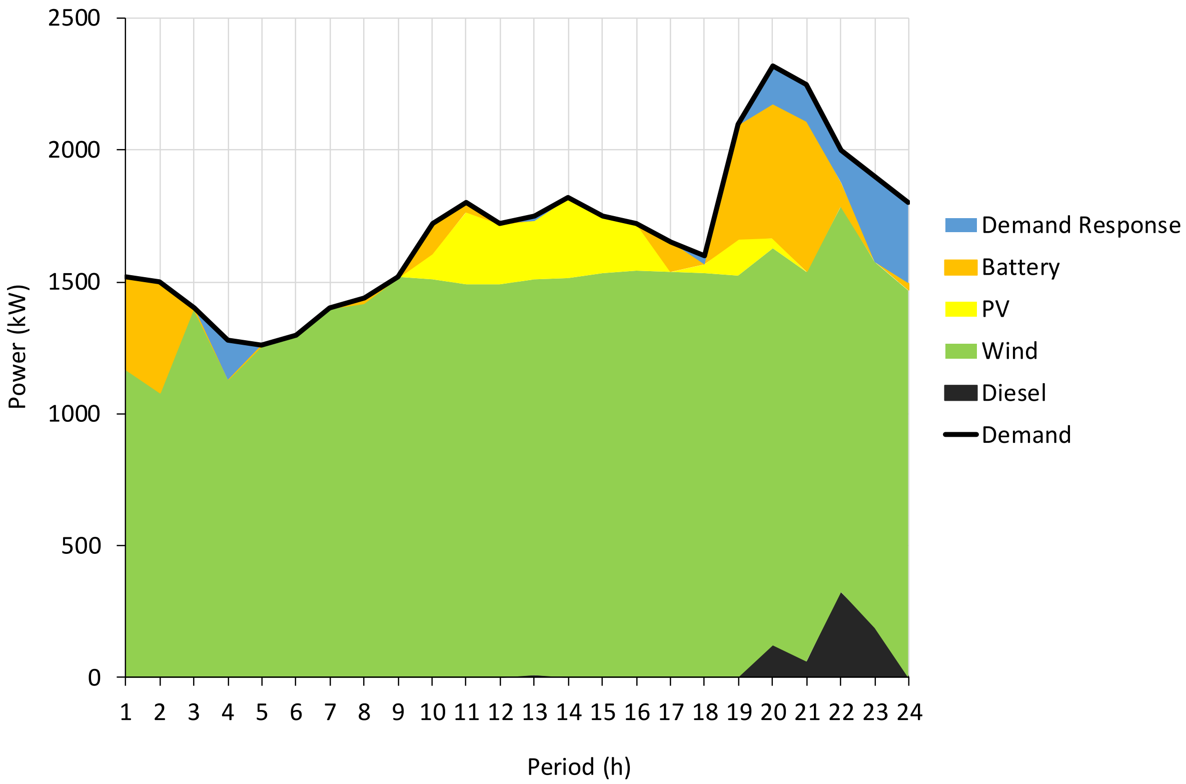


Figure 60 Dispatch and total generation in winter demand scenario with demand response

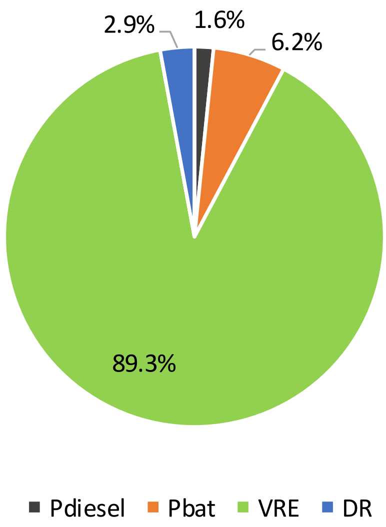


Figure 61 Total generation in the Demand Response scenario

In the normal winter demand scenario shown in Figure 45 and Figure 49 the diesel generation was a 4% of the total, something that produces an increase in total system costs. By adding a demand response program in the island, some diesel generation could be avoided, and total system costs can be reduced. In this case, demand response accounts for 2.9% of total generation while diesel decreases to only a 1.6%. Additionally, the battery throughput increases to a 6.2% of total generation. Finally, in terms of costs, Table 5-8 shows the difference in a scenario with and without demand response.

Table 5-8 Cost comparison of winter demand scenario with and without demand response

Winter Demand Scenario	Diesel Costs (€)	Degradation Cost (€)	DR Costs (€)	Total Costs
With Demand Response	1001.61	300.51	188.3	1631.10
Without Demand Response	1379.44	306.72	0	1686.17

This table shows that demand response can, not only avoid diesel costs but also battery degradation costs, increasing battery lifetime. It will be therefore advantageous to implement a demand response program into Graciosa island to reduce total system costs and to avoid unnecessary battery replacement costs.

5.2.5.2 Increase the size of the battery storage system

Another possible solution to increase the performance of the microgrid would be to increase the size of the battery storage system. The problem of this measure is that LTO batteries are the most expensive kind of Li-ion batteries because of the elevated price of titanium. This causes that the price per kWh of an LTO battery is around 1000€/kWh according to IRENA [6]. Apart from this, increasing the battery size would imply investment costs on new inverters and electrical equipment, operation and maintenance costs, replacement costs and in the case of Graciosa high transportation costs. Before applying this measure, a project valuation - as the one made by Younicos and Leclanché for Graciosa - would be needed, and the result would probably be not economically feasible.

To do a battery project valuation is not one of the goals of this dissertation, so it will be assumed that it is possible to double the size of the storage resulting in a 6400kWh battery with 12000kW of inverter power. This new battery storage system was included in scenarios in which it could be interesting to have a bigger battery storage system. At the end, the scenarios chosen were the winter demand scenarios and the highest wind variability scenario because a higher battery would be able to shift unnecessary VRE curtailment to periods with diesel generation, reducing system's costs. Figure 62 and Figure 63 show the dispatch in the winter demand and highest wind variability scenarios respectively, while XX shows the total generation in both scenarios.

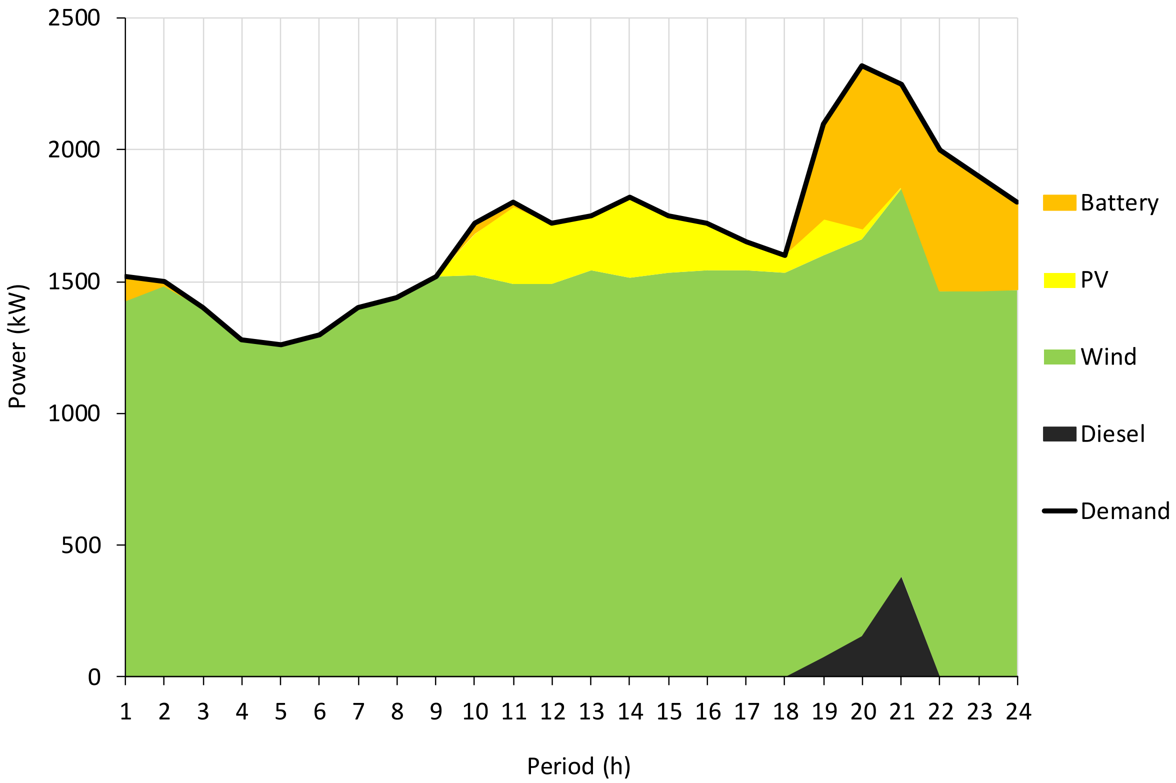


Figure 62 Optimal dispatch in the winter demand scenarios with doubled battery storage

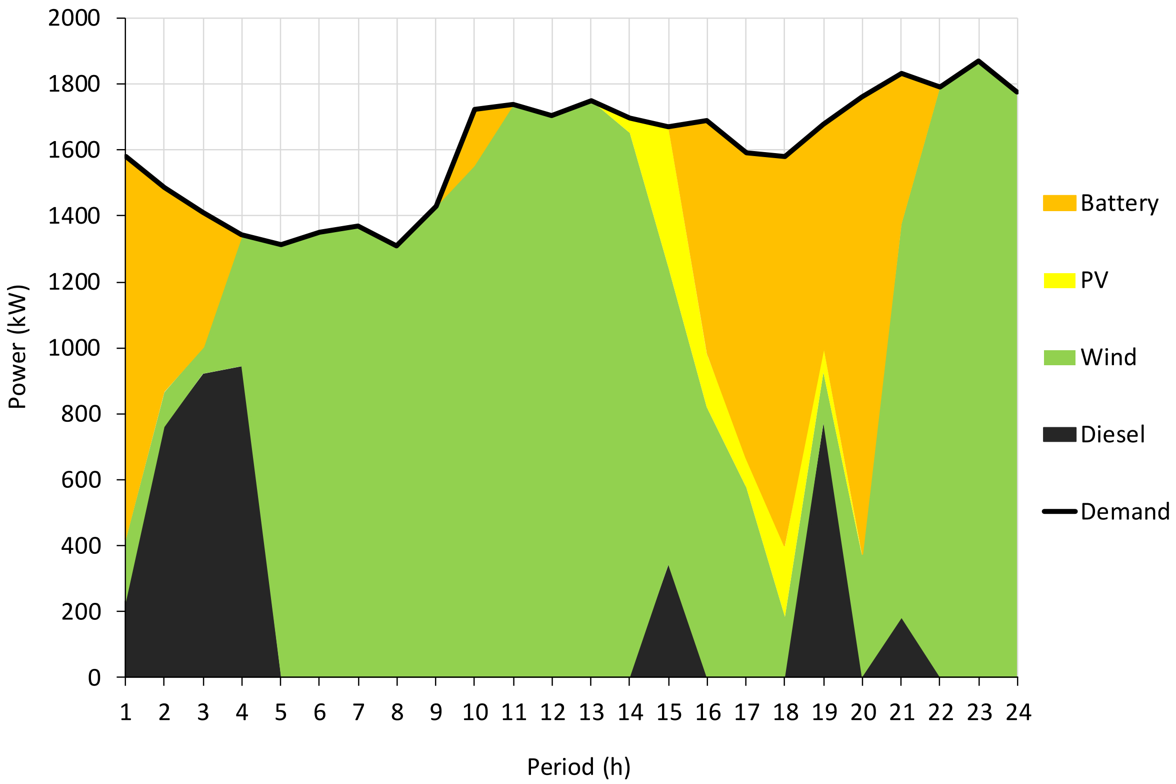


Figure 63 Optimal dispatch in the highest wind variability scenario with doubled battery storage

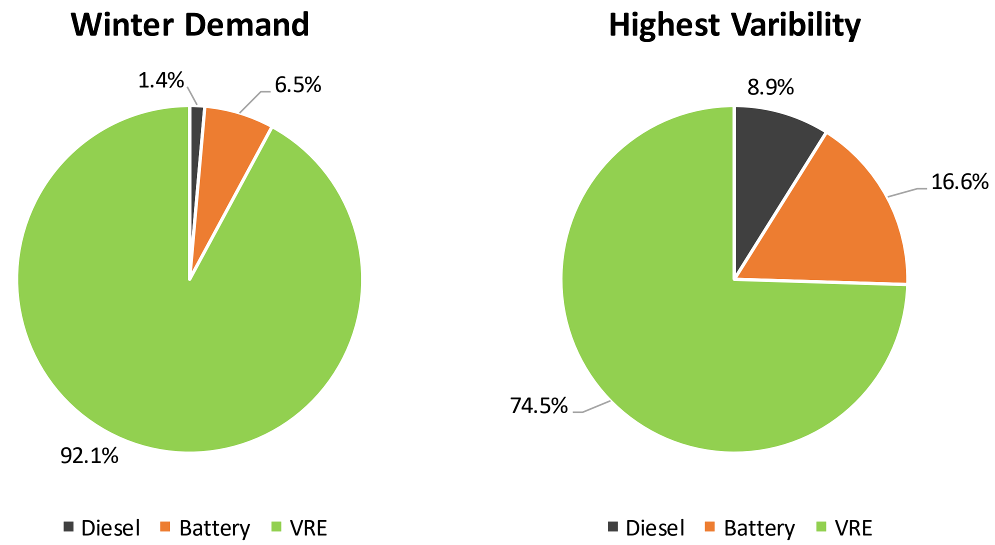


Figure 64 Total generation in the winter demand and highest wind variability scenarios with doubled battery storage

In the highest wind variability scenario, if the battery is doubled, the total diesel generation is more than halved, reducing from a 19% of total generation to only a 8.9%. The battery in this scenario has a higher capacity and therefore it can better support the wind power production when this is high, reducing VRE curtailment. In the winter demand scenario the effects of doubling the battery are not that significant and diesel is only reduced from a 4% of total generation to a 1%. In this scenario the battery also tries to benefit from excessive VRE generation to avoid diesel generation, however there is not that much VRE production as in the highest variability scenario to be absorbed and the effect of doubling the battery are limited. The dispatch of the battery as well as the origin of its stored energy is shown in Figure 65 for winter demand and highest variability scenarios.

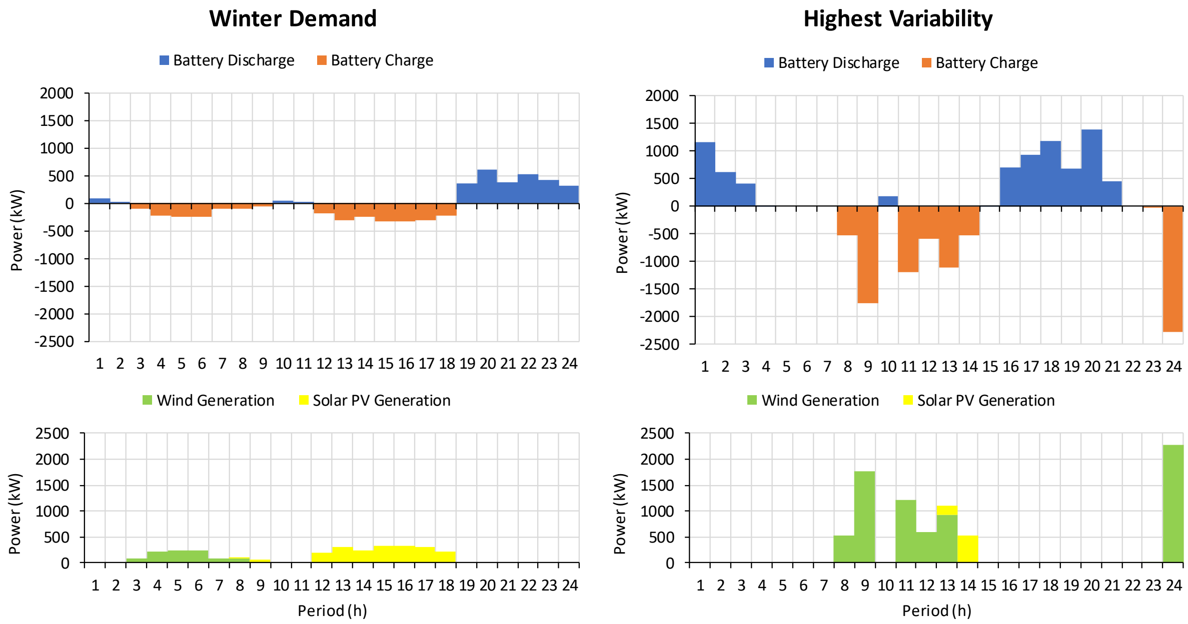


Figure 65 Battery charging and discharging profiles for the winter demand and the highest wind variability scenarios with doubled battery storage

Finally, in terms of costs both scenarios avoid diesel generation and therefore reduce the total system costs. Table 5-9 shows a comparion of total system costs in both analyzed scenarios with a single and a double battery storage system.

Table 5-9 Total system cost for the winter demand and highest wind variability scenarios with single and double battery storage

Winter Demand	**Diesel Costs (€)**	**Degradation Costs (€)**	**Total Costs (€)**
Single Battery	1379.44	306.72	1686.17
Double Battery	1003.43	599.44	1602.88
Highest Variability			
Single Battery	5114.12	299.06	5413.19
Double Battery	2811.20	576.56	3387.76

As already expected, in the highest variability scenario the cost reduction because of increasing the battery size is much significant than in the winter demand scenario, with 2000€ of savings per day. This is because, as already explained, in the highest variability scenario a bigger battery storage can help to reduce the total generation by shifting VRE production to those periods with no VRE, smoothing the wind profile. It can be therefore concluded that only for some scenarios it could be beneficial to have a bigger battery storage system, however, with this analysis it cannot be affirmed that a bigger battery would be profitable for Graciosa's power system. A long-term analysis with project valuation would be required for this purpose.

5.2.5.3 Future Work: Introduce Electric Vehicles and V2G

Batteries have typically two applications. They can be used as stationary storage like the one that have been used for this case study or they can be used for electromobility, which includes Electric Vehicles (EVs). Graciosa, due to its small land area, present a fantastic opportunity for EVs deployment.

EVs main application is electromobility, however their battery must be charged by connecting themselves to the grid. When this occurs, EVs can be included inside the demand response program if the charger is unidirectional or provide Vehicle-to-Grid (V2G) if the charger is bidirectional. The first approach would be formulated as the demand response presented in this dissertation. The input demand response band would vary but the analysis would be the same. A more interesting case to analyze would be the second approach, where vehicles can charge and discharge energy to the grid with what is known as V2G. The analysis of the EVs providing V2G into Graciosa's power system or in Smart Grids in general will be left as potential future work that would imply modifying the optimization tool to model also electric vehicles and, although a little bit futuristic and only interesting nowadays for research purposes, V2G.

6 Conclusions and Future Work

6.1 Conclusions

Electric power systems, that traditionally have had a centralized structure, where generation was sent via transmission and distribution to the customers, have been changing in the last decade to a more decentralized structure. The increasing penetration of renewable energy and specially of those known as Variable Renewable Energy (VRE) combined with the expansion of markets, the inclusion of new participants and the development of telecommunication technologies have led to a new grid paradigm: the Smart Grid.

Smart Grids are composed by a set of generation technologies called Distributed Energy Resources (DER) which are smaller generators which are typically connected to the distribution grid. Among these DER, energy storage systems and more specifically battery storage systems are gaining an increasing interest due to its capabilities to quickly respond to changes and cope with the flexibility challenges that VRE pose when their penetration is high. Battery storage have several advantages for the Smart Grid however it is a technology that suffers from ageing that depend on how it is operated. This makes very challenging its operation. Optimization techniques are then required to optimally operate a Smart Grid with battery storage systems.

Traditionally, classic optimization techniques have been used to solve the economic dispatch problem. Techniques such as linear programming or mixed integer programming have been widely used to solve this problem. However, in the economic dispatch of a Smart Grid, non-linearities or stochasticity appears, and these classic methods are no longer efficient. For this reason, this dissertation has proposed the development of a tool using modern optimization techniques such as Particle Swarm Optimization or Genetic Algorithm for the optimal operation of a Smart Grid. After a deep revision of these modern optimization techniques the tool was developed to be as generic as possible. The tool can solve an economic dispatch problem of a Smart Grid containing VRE, diesel generation, grid connection, demand response and batteries, where a non-linear degradation model was included to optimally account for the ageing during operation that this type of storage suffers. This is solved using either GA or PSO with the possibility of choosing many different variants.

To test the tool, a real islanded microgrid (which is a type of Smart Grid) case study was chosen: Graciosa in Azores (Portugal). Composed by only VRE, diesel and a LTO li-ion battery, this case study was considered suitable to test the capabilities of the tool. In this analysis, first a base case was solved to then compare it with different scenarios of demand and wind. Finally, a set of measures that could improve the performance of the system were presented. The main results of this analysis were that although Graciosa power system can sometimes operate using 100% of renewables, as it is intended, the diesel generator is still needed in many of the scenarios analyzed. To avoid excessive diesel generation, it could be beneficial to add a demand response program or increase the battery size if economically feasible. Apart from this, an analysis of how degradation modeling would affect operation was made and it was proven that the total ageing of the battery is significantly higher if this is not included in the economic dispatch problem.

Apart from that, and from a personal perspective, during the development of this dissertation I could observe that EAs are very powerful to solve problems that include non-linear equations such as the degradation model presented or the heat rate quadratic function of the diesel generation. In this sense, EAs overcome the capabilities of Mixed Integer Programming. However, I could also observe that EAs have some disadvantages. First, EAs are heuristic algorithms and therefore it is not possible to know with certainty if the solution found is the global optimum or not. Also, being heuristic implies that the performance of the algorithm highly depends on the user-defined parameters. Second, EAs lack of robustness, this is, there is not a general EA that will be able to perform better than other in a wide range of problems. For every specific problem a specific EA variation will perform better than others. In literature Genetic Algorithm has been proven to be the most robust EA that in average performs better, however for this specific problem solved we showed that PSO is a better choice. Finally, EAs are computationally intensive and using other optimization techniques could be more efficient.

As a final conclusion, I would like to say that if I had to solve an economic dispatch problem I would use the classic optimization techniques such as LP, MIP, NLP… which are widespread and more robust methods that in my opinion could be more adequate to solve an economic dispatch problem. However, in some specific problems the classic techniques could fail. In this case researchers should have a toolbox that goes beyond the classical optimization techniques and in this case, modern optimization techniques, that include GA and PSO, could help.

6.2 Future Work

This dissertation opens a range of possible future work. The optimization tool developed could be reused and further improved to include new evolutionary algorithms that could be interesting for power system optimization, to include new variants of the algorithm that could perform better than the implemented ones or to improve the tool interface and code. Apart from this, the tool includes a post-processing part in Matlab which was not fully developed since Excel was chosen for plotting in this dissertation. A potential future work would be advancing the post-processing module to create good looking Matlab plots automatically, so that a full Matlab-based platform could be used.

Regarding the formulation, it could be interesting to add an electric vehicles' module with vehicle-to-grid that also includes a battery degradation model to minimize the ageing of the battery. However, this would be possible and relevant after the implementation of bidirectional charging stations or for research purposes. Apart from this, new optimization problems could be added (e.g. model the nodes and lines to solve the optimal power flow).

Bibliography

[1] Laloux, D., & Rivier, M. (2013). Technology and operation of electric power systems. In Regulation of the Power Sector (pp. 1-46). Springer London.

[2] Farhangi, H. (2010). The path of the smart grid. IEEE power and energy magazine, 8(1).

[3] Palensky, P., & Dietrich, D. (2011). Demand side management: Demand response, intelligent energy systems, and smart loads. IEEE transactions on industrial informatics, 7(3), 381-388.

[4] EPRI (2017). Distributed Energy Resources. Electric Power Research Institute Website. Retrieved the 23rd November 2017 from http://www2.epri.com/Our-Work/Pages/Distributed-Electricity-Resources.aspx

[5] Perez-Arriaga, I. J., & Batlle, C. (2012). Impacts of intermittent renewables on electricity generation system operation. Economics of Energy & Environmental Policy, 1(2).

[6] IRENA (2017). Electricity storage and renewables: Costs and markets to 2030. The International Renewable Energy Agency, Abu Dhabi.

[7] Scrosati, B., Hassoun, J., & Sun, Y. K. (2011). Lithium-ion batteries. A look into the future. Energy & Environmental Science, 4(9), 3287-3295.

[8] Dunn, B., Kamath, H., & Tarascon, J. M. (2011). Electrical energy storage for the grid: a battery of choices. Science, 334(6058), 928-935.

[9] EASAC (2017). Valuing dedicated storage in electricity grids. European Academies' Science Advisory Council. Brussels, Belgium.

[10] Soliman, S. A. H., & Mantawy, A. A. H. (2011). Modern optimization techniques with applications in electric power systems. Springer Science & Business Media.

[11] Back, T. (1996). Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford university press.

[12] Celli, G., Ghiani, E., Mocci, S., & Pilo, F. (2005). A multiobjective evolutionary algorithm for the sizing and siting of distributed generation. IEEE Transactions on power systems, 20(2), 750-757.

[13] LI, Z. K., CHEN, X. Y., Yu, K., LIU, H. M., & Zhao, B. (2008). Hybrid particle swarm optimization for distribution network reconfiguration. Proceedings of the CSEE, 31, 35-41.

[14] Soares, J., Sousa, T., Morais, H., Vale, Z., & Faria, P. (2011, April). An optimal scheduling problem in distribution networks considering V2G. In Computational Intelligence Applications In Smart Grid (CIASG), 2011 IEEE Symposium on (pp. 1-8). IEEE.

[15] Pedrasa, M. A. A., Spooner, T. D., & MacGill, I. F. (2010). Coordinated scheduling of residential distributed energy resources to optimize smart home energy services. IEEE Transactions on Smart Grid, 1(2), 134-143.

[16] Darwin, C. (1859). On the origin of species. Lulu. Com

[17] Simon, D. (2013). Evolutionary optimization algorithms. John Wiley & Sons.

[18] Melanie, M. (1999). An introduction to genetic algorithms. Cambridge, Massachusetts London, England, Fifth printing, 3, 62-75.

[19] Carr, J. (2014). An introduction to genetic algorithms. Senior Project, 1-40.

[20] Barricelli, N. (1954). Esempi numerici di processi di evoluzione. Methodos, 6:45-68. The English translation of the title is Numerical models of evolutionary processes.

[21] Holland, J. H. (1992). Genetic algorithms. Scientific american, 267(1), 66-73

[22] Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). Artificial intelligence through simulated evolution.

[23] Jefferson, D., Collins, R., Cooper, C., Dyer, M., Korf, M. F. R., Taylor, C., and Wang, A. 1991. Evolution as a theme in artificial life: The genesys/tracker system. In Langton, C. et al., editors, Proceedings of Artificial Life II, Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley

[24] Rechenberg, I. (1965). Cybernetic solution path of an experimental problem.

[25] Rechenberg, I. (1973). Evolutionstrategie (Evolution Strategy). Frommann-Holzboog, Stuttgart.

[26] Hansen, N., Arnold, D. V., & Auger, A. (2015). Evolution strategies. In Springer handbook of computational intelligence (pp. 871-898). Springer Berlin Heidelberg.

[27] Back, T., Hoffmeister, F., & Schwefel, H. P. (1991, July). A survey of evolution strategies. In Proceedings of the fourth international conference on genetic algorithms (Vol. 2, No. 9). Morgan Kaufmann Publishers, San Mateo.

[28] Koza, J. R. (1992). Genetic programming: on the programming of computers by means of natural selection (Vol. 1). MIT press

[29] Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE transactions on Evolutionary Computation, 6(1), 58-73.

[30] Eberhart, R. C., & Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In Evolutionary Computation, 2000. Proceedings of the 2000 Congress on (Vol. 1, pp. 84-88). IEEE.

[31] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. science, 220(4598), 671-680.

[32] Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. IEEE computational intelligence magazine, 1(4), 28-39.

[33] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4), 341-359.

[34] Bengoetxea, E. (2002). Inexact Graph Matchig using Estimation of Distribution Algortihms. PhD Thesis, Chapter 4,43-76

[35] Reynolds, R. G. (1994, February). An introduction to cultural algorithms. In Proceedings of the third annual conference on evolutionary programming (Vol. 131139). Singapore.

[36] Tizhoosh, H. R. (2005, November). Opposition-based learning: a new scheme for machine intelligence. In Computational intelligence for modelling, control and automation, 2005 and international conference on intelligent agents, web technologies and internet commerce, international conference on (Vol. 1, pp. 695-701). IEEE.

[37] Glover, F. (1989). Tabu search—part I. ORSA Journal on computing, 1(3), 190-206.

[38] Li, X. L. (2003). A new intelligent optimization-artificial fish swarm algorithm. Doctor thesis, Zhejiang University of Zhejiang, China

[39] He, S., Wu, Q. H., & Saunders, J. R. (2009). Group search optimizer: an optimization algorithm inspired by animal searching behavior. IEEE transactions on evolutionary computation, 13(5), 973-990

[40] Eusuff, M., Lansey, K., & Pasha, F. (2006). Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. Engineering optimization, 38(2), 129-154

[41] Yang, X. S. (2010). Firefly algorithm. Engineering optimization, 221-230.

[42] Passino, K. M. (2012). Bacterial foraging optimization. Innovations and Developments of Swarm Intelligence Applications, 219.

[43] Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of global optimization, 39(3), 459-471.

[44] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. Information sciences, 179(13), 2232-2248.

[45] Newton, I., Bernoulli, D., MacLaurin, C., & Euler, L. (1833). Philosophiae naturalis principia mathematica (Vol. 1). excudit G. Brookman; impensis TT et J. Tegg, Londini.

[46] Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. simulation, 76(2), 60-68

[47] Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-Aided Design, 43(3), 303-315.

[48] Ray, T., & Liew, K. M. (2003). Society and civilization: An optimization algorithm based on the simulation of social behavior. IEEE Transactions on Evolutionary Computation, 7(4), 386-396.

[49] Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: charged system search. Acta Mechanica, 213(3), 267-289.

[50] Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. Ecological informatics, 1(4), 355-366.

[51] Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on (pp. 210-214). IEEE.

[52] Shah-Hosseini, Hamed. "The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm." International Journal of Bio-Inspired Computation 1.1-2 (2009): 71-79.

[53] Bishop, J. M. (1989, October). Stochastic searching networks. In Artificial Neural Networks, 1989., First IEE International Conference on (Conf. Publ. No. 313) (pp. 329-331). IET.

[54] Kjellström, G. (1991). On the efficiency of Gaussian adaptation. Journal of Optimization Theory and Applications, 71(3), 589-597.

[55] Erol, Osman K., and Ibrahim Eksin. "A new optimization method: big bang–big crunch." Advances in Engineering Software 37.2 (2006): 106-111.

[56] Atashpaz-Gargari, Esmaeil, and Caro Lucas. "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition." Evolutionary computation, 2007. CEC 2007. IEEE Congress on. IEEE, 2007.

[57] Joslin, David E., and David P. Clements. "Squeaky wheel optimization." Journal of Artificial Intelligence Research 10 (1999): 353-373.

[58] O'Neil, Michael, and Conor Ryan. "Grammatical evolution." Grammatical Evolution. Springer US, 2003. 33-47.

[59] Krishnanand, K. N., and Debasish Ghose. "Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions." Swarm intelligence 3.2 (2009): 87-124.

[60] Lam, Albert YS, and Victor OK Li. "Chemical-reaction-inspired metaheuristic for optimization." IEEE Transactions on Evolutionary Computation 14.3 (2010): 381-399.

[61] Gandomi, Amir Hossein, and Amir Hossein Alavi. "Krill herd: a new bio-inspired optimization algorithm." Communications in Nonlinear Science and Numerical Simulation 17.12 (2012): 4831-4845.

[62] Yang, Xin-She. "A new metaheuristic bat-inspired algorithm." Nature inspired cooperative strategies for optimization (NICSO 2010) (2010): 65-74.

[63] Dueck, Gunter, and Tobias Scheuer. "Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing." Journal of computational physics 90.1 (1990): 161-175.

[64] Dueck, Gunter. "New optimization heuristics: The great deluge algorithm and the record-to-record travel." Journal of Computational physics 104.1 (1993): 86-92.

[65] Muller, Sibylle D., et al. "Optimization based on bacterial chemotaxis." IEEE transactions on Evolutionary Computation 6.1 (2002): 16-29.

[66] Ahrari, Ali, and Ali A. Atai. "Grenade explosion method—a novel tool for optimization of multimodal functions." Applied Soft Computing 10.4 (2010): 1132-1140.

[67] Jabr, R. A., Coonick, A. H., & Cory, B. J. (2000). A homogeneous linear programming algorithm for the security constrained economic dispatch problem. IEEE Transactions on power systems, 15(3), 930-936.

[68] Granville, S. (1994). Optimal reactive dispatch through interior point methods. IEEE Transactions on power systems, 9(1), 136-146.

[69] Gan, D., Thomas, R. J., & Zimmerman, R. D. (2000). Stability-constrained optimal power flow. IEEE Transactions on Power Systems, 15(2), 535-540.

[70] Clímaco, J., Henggeler Antunes, C., Gomes Martins, A., & Traca Almeida, A. (1995). A multiple objective linear programming model for power generation expansion planning. International Journal of Energy Research, 19(5), 419-432.

[71] Carrión, M., & Arroyo, J. M. (2006). A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. IEEE Transactions on power systems, 21(3), 1371-1378.

[72] Alizadeh, B., & Jadid, S. (2011). Reliability constrained coordination of generation and transmission expansion planning in power systems using mixed integer programming. IET generation, transmission & distribution, 5(9), 948-960.

[73] Gönen, T., & Foote, B. L. (1981, March). Distribution-system planning using mixed-integer programming. In IEE Proceedings C (Generation, Transmission and Distribution) (Vol. 128, No. 2, pp. 70-79). IET Digital Library.

[74] Ramos, A., Perez-Arriaga, I. J., & Bogas, J. (1989). A nonlinear programming approach to optimal static generation expansion planning. IEEE Transactions on Power Systems, 4(3), 1140-1146.

[75] Wei, H., Sasaki, H., Kubokawa, J., & Yokoyama, R. (1998). An interior point nonlinear programming for optimal power flow problems with a novel data structure. IEEE Transactions on Power Systems, 13(3), 870-877.

[76] Gallego, R. A., Monticelli, A. J., & Romero, R. (2001). Optimal capacitor placement in radial distribution networks. IEEE Transactions on Power Systems, 16(4), 630-637.

[77] Frangioni, A., & Gentile, C. (2006). Solving nonlinear single-unit commitment problems with ramping constraints. Operations Research, 54(4), 767-775.

[78] Snyder, W. L., Powell, H. D., & Rayburn, J. C. (1987). Dynamic programming approach to unit commitment. IEEE Transactions on Power Systems, 2(2), 339-348.

[79] Ferrero, R. W., Rivera, J. F., & Shahidehpour, S. M. (1998). A dynamic programming two-stage algorithm for long-term hydrothermal scheduling of multireservoir systems. IEEE Transactions on Power Systems, 13(4), 1534-1540.

[80] Shapiro, A. (2011). Analysis of stochastic dual dynamic programming method. European Journal of Operational Research, 209(1), 63-72.

[81] Del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J. C., & Harley, R. G. (2008). Particle swarm optimization: basic concepts, variants and applications in power systems. IEEE Transactions on evolutionary computation, 12(2), 171-195.

[82] Chen, P. H., & Chang, H. C. (1995). Large-scale economic dispatch by genetic algorithm. IEEE transactions on power systems, 10(4), 1919-1926.

[83] Gaing, Z. L. (2003). Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE transactions on power systems, 18(3), 1187-1195

[84] Sinha, N., Chakrabarti, R., & Chattopadhyay, P. K. (2003). Evolutionary programming techniques for economic load dispatch. IEEE Transactions on evolutionary computation, 7(1), 83-94.

[85] Juste, K. A., Kita, H., Tanaka, E., & Hasegawa, J. (1999). An evolutionary programming solution to the unit commitment problem. IEEE Transactions on Power Systems, 14(4), 1452-1459.

[86] Hou, Y. H., Wu, Y. W., Lu, L. J., & Xiong, X. Y. (2002, October). Generalized ant colony optimization for economic dispatch of power systems. In Power System Technology, 2002. Proceedings. PowerCon 2002. International Conference on (Vol. 1, pp. 225-229). IEEE.

[87] Park, J. B., Park, Y. M., Won, J. R., & Lee, K. Y. (2000). An improved genetic algorithm for generation expansion planning. IEEE Transactions on Power Systems, 15(3), 916-922.Dommel, H. W., & Tinney, W. F. (1968). Optimal power flow solutions. IEEE Transactions on power apparatus and systems, (10), 1866-1876.

[88] Kannan, S., Slochanal, S. M. R., Subbaraj, P., & Padhy, N. P. (2004). Application of particle swarm optimization technique and its variants to generation expansion planning problem. Electric Power Systems Research, 70(3), 203-210.

[89] Meza, J. L. C., Yildirim, M. B., & Masud, A. S. (2009). A multiobjective evolutionary programming algorithm and its applications to power generation expansion planning. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 39(5), 1086-1096.

[90] Carpentier, J. (1962). Contribution to the economic dispatch problem. Bulletin de la Societe Francoise des Electriciens, 3(8), 431-447.

[91] Bakirtzis, A. G., Biskas, P. N., Zoumas, C. E., & Petridis, V. (2002). Optimal power flow by enhanced genetic algorithm. IEEE Transactions on power Systems, 17(2), 229-236.

[92] Bouktir, T., & Slimani, L. (2005). Optimal power flow of the algerian electrical network using an ant colony optimization method. Leonardo Journal of Sciences, 6, 43-57.

[93] Rarick, R., Simon, D., Villaseca, F. E., & Vyakaranam, B. (2009, October). Biogeography-based optimization and the solution of the power flow problem. In Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on (pp. 1003-1008). IEEE.

[94] Yuryevich, J., & Wong, K. P. (1999). Evolutionary programming based optimal power flow algorithm. IEEE transactions on Power Systems, 14(4), 1245-1250.

[95] da Silva, E. L., Gil, H. A., & Areiza, J. M. (1999, July). Transmission network expansion planning under an improved genetic algorithm. In Power Industry Computer Applications, 1999. PICA'99. Proceedings of the 21st 1999 IEEE International Conference (pp. 315-321). IEEE.

[96] Da Rocha, M. C., & Saraiva, J. T. (2011, August). Discrete evolutionary particle swarm optimization for multiyear transmission expansion planning. In Proceedings of the 17th Power Systems Computation Conference PSCC (pp. 1-8).

[97] Romero, R., Gallego, R. A., & Monticelli, A. (1995, May). Transmission system expansion planning by simulated annealing. In Power Industry Computer Application Conference, 1995. Conference Proceedings., 1995 IEEE (pp. 278-283). IEEE.

[98] Da Silva, E. L., Ortiz, J. A., De Oliveira, G. C., & Binato, S. (2001). Transmission network expansion planning under a tabu search approach. IEEE Transactions on Power Systems, 16(1), 62-68.

[99] Iba, K. (1994). Reactive power optimization by genetic algorithm. IEEE Transactions on Power Systems, 9(2), 685-692.

[100] Zhao, B., Guo, C. X., & Cao, Y. J. (2005). A multiagent-based particle swarm optimization approach for optimal reactive power dispatch. IEEE transactions on power systems, 20(2), 1070-1078.

[101] El Ela, A. A., Abido, M. A., & Spea, S. R. (2011). Differential evolution algorithm for optimal reactive power dispatch. Electric Power Systems Research, 81(2), 458-464.

[102] Roy, P. K., Ghoshal, S. P., & Thakur, S. S. (2012). Optimal var control for improvements in voltage profiles and for real power loss minimization using biogeography based optimization. International Journal of Electrical Power & Energy Systems, 43(1), 830-838.

[103] Higorani, N. G., & Gyugyi, L. (2000). Understanding FACTS Devices.

[104] Gerbex, S., Cherkaoui, R., & Germond, A. J. (2001). Optimal location of multi-type FACTS devices in a power system by means of genetic algorithms. IEEE transactions on power systems, 16(3), 537-544.

[105] Saravanan, M., Slochanal, S. M. R., Venkatesh, P., & Abraham, P. S. (2005, November). Application of PSO technique for optimal location of FACTS devices considering system loadability and cost of installation. In Power Engineering Conference, 2005. IPEC 2005. The 7th International (pp. 716-721). IEEE.

[106] Mori, H., & Goto, Y. (2000). A parallel tabu search based method for determining optimal allocation of FACTS in power systems. In Power System Technology, 2000. Proceedings. PowerCon 2000. International Conference on (Vol. 2, pp. 1077-1082). IEEE.

[107] Farhoodnea, M., Mohamed, A., Shareef, H., & Zayandehroodi, H. (2013, June). Optimum D-STATCOM placement using firefly algorithm for power quality enhancement. In Power Engineering and Optimization Conference (PEOCO), 2013 IEEE 7th International (pp. 98-102). IEEE.

[108] Zhu, J. Z. (2002). Optimal reconfiguration of electrical distribution network using the refined genetic algorithm. Electric Power Systems Research, 62(1), 37-42.

[109] LI, Z. K., CHEN, X. Y., Yu, K., LIU, H. M., & Zhao, B. (2008). Hybrid particle swarm optimization for distribution network reconfiguration. Proceedings of the CSEE, 31, 35-41.

[110] Parada, V., Ferland, J. A., Arias, M., & Daniels, K. (2004). Optimization of electrical distribution feeders using simulated annealing. IEEE Transactions on Power Delivery, 19(3), 1135-1141.

[111] Su, C. T., & Lee, C. S. (2003). Network reconfiguration of distribution systems using improved mixed-integer hybrid differential evolution. IEEE Transactions on Power Delivery, 18(3), 1022-1027.

[112] Ackermann, T., Andersson, G., & Söder, L. (2001). Distributed generation: a definition. Electric power systems research, 57(3), 195-204.

[113] Celli, G., Ghiani, E., Mocci, S., & Pilo, F. (2005). A multiobjective evolutionary algorithm for the sizing and siting of distributed generation. IEEE Transactions on power systems, 20(2), 750-757.

[114] Krueasuk, W., & Ongsakul, W. (2006, December). Optimal placement of distributed generation using particle swarm optimization. In Proceedings of Power Engineering Conference in Australasian Universities, Australia.

[115] Falaghi, H., & Haghifam, M. R. (2007, July). ACO based algorithm for distributed generation sources allocation and sizing in distribution systems. In Power Tech, 2007 IEEE Lausanne (pp. 555-560). IEEE.

[116] Abu-Mouti, F. S., & El-Hawary, M. E. (2011). Optimal distributed generation allocation and sizing in distribution systems via artificial bee colony algorithm. IEEE transactions on power delivery, 26(4), 2090-2101.

[117] Ghoshal, S. P. (2004). Application of GA/GA-SA based fuzzy automatic generation control of a multi-area thermal generating system. Electric Power Systems Research, 70(2), 115-127.

[118] Ghoshal, S. P. (2004). Optimizations of PID gains by particle swarm optimizations in fuzzy based automatic generation control. Electric Power Systems Research, 72(3), 203-212.

[119] Nanda, J., Mishra, S., & Saikia, L. C. (2009). Maiden application of bacterial foraging-based optimization technique in multiarea automatic generation control. IEEE Transactions on power systems, 24(2), 602-609.

[120] Barisal, A. K. (2015). Comparative performance analysis of teaching learning based optimization for automatic load frequency control of multi-source power systems. International Journal of Electrical Power & Energy Systems, 66, 67-77.

[121] Bishey R. (2017). Smart Grid Security Market: Technological Progress in Energy & Power Industry trends 2025. Retrieved the 8th of March 2018 from http://www.editiontruth.com/smart-grid-security-market-technological-progress-energy-power-industry-trends-2025/

[122] Fang, X., Misra, S., Xue, G., & Yang, D. (2012). Smart grid—The new and improved power grid: A survey. IEEE communications surveys & tutorials, 14(4), 944-980.

[123] Metke, A. R., & Ekl, R. L. (2010). Security technology for smart grid networks. IEEE Transactions on Smart Grid, 1(1), 99-107.

[124] Gungor, V. C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., & Hancke, G. P. (2011). Smart grid technologies: Communication technologies and standards. IEEE transactions on Industrial informatics, 7(4), 529-539.

[125] Hochgraf, C., Tripathi, R., & Herzberg, S. (2010, October). Smart grid charger for electric vehicles using existing cellular networks and SMS text messages. In Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on (pp. 167-172). IEEE.

[126] Yu, R., Zhang, Y., Gjessing, S., Yuen, C., Xie, S., & Guizani, M. (2011). Cognitive radio based hierarchical communications infrastructure for smart grid. IEEE network, 25(5).

[127] Gungor, V. C., Lu, B., & Hancke, G. P. (2010). Opportunities and challenges of wireless sensor networks in smart grid. IEEE transactions on industrial electronics, 57(10), 3557-3564.

[128] Deep, U. D., Petersen, B. R., & Meng, J. (2009). A smart microcontroller-based iridium satellite-communication architecture for a remote renewable energy source. IEEE Transactions on Power Delivery, 24(4), 1869-1875.

[129] Britz, D. M., & Miller, R. R. (2007, June). Mesh free space optical systems: A method to improve broadband neighborhood area network backhaul. In Local & Metropolitan Area Networks, 2007. LANMAN 2007. 15th IEEE Workshop on (pp. 37-42). IEEE.

[130] McGranaghan, M., & Goodman, F. (2005, June). Technical and system requirements for advanced distribution automation. In 18th International Conference and Exhibition on Electricity Distribution (Vol. 5, p. 93).

[131] Barmada, S., Musolino, A., Raugi, M., Rizzo, R., & Tucci, M. (2011). A wavelet based method for the analysis of impulsive noise due to switch commutations in power line communication (PLC) systems. IEEE Transactions on Smart Grid, 2(1), 92-101.

[132] Wang, W., & Lu, Z. (2013). Cyber security in the Smart Grid: Survey and challenges. Computer Networks, 57(5), 1344-1371.

[133] Li, F., Qiao, W., Sun, H., Wan, H., Wang, J., Xia, Y., ... & Zhang, P. (2010). Smart transmission grid: Vision and framework. IEEE transactions on Smart Grid, 1(2), 168-177.

[134] M5BAT (2016). M5BAT: A unique hybrid battery storage system. Retrieved February 12th 2017 from http://m5bat.de/en-gb/

[135] The Guardian (2017). South Australia turns on Tesla's 100MW battery: 'History on the making'. Retrieved February 12th 2017 from: https://www.theguardian.com/australia-news/2017/dec/01/south-australia-turns-on-teslas-100mw-battery-history-in-the-making

[136] Díaz-González, F., Sumper, A., Gomis-Bellmunt, O., & Villafáfila-Robles, R. (2012). A review of energy storage technologies for wind power applications. Renewable and sustainable energy reviews, 16(4), 2154-2171.

[137] Akhil, A. A., Huff, G., Currier, A. B., Kaun, B. C., Rastler, D. M., Chen, S. B., ... & Gauntlett, W. D. (2013). DOE/EPRI 2013 electricity storage handbook in collaboration with NRECA. Albuquerque, NM: Sandia National Laboratories.

[138] Mahlia, T. M. I., Saktisahdan, T. J., Jannifar, A., Hasan, M. H., & Matseelar, H. S. C. (2014). A review of available methods and development on energy storage; technology update. Renewable and Sustainable Energy Reviews, 33, 532-545.

[139] Battery University (2017). Lithium-based batteries. Retrieved 18th February 2018 from http://batteryuniversity.com/learn/article/lithium_based_batteries

[140] Battery University (2017). BU-205: Types of Lithium-ion. Retrieved 14th February 2018 from http://batteryuniversity.com/learn/article/types_of_lithium_ion

[141] Vetter, J., Novák, P., Wagner, M. R., Veit, C., Möller, K. C., Besenhard, J. O., ... & Hammouche, A. (2005). Ageing mechanisms in lithium-ion batteries. Journal of power sources, 147(1-2), 269-281.

[142] Barré, A., Deguilhem, B., Grolleau, S., Gérard, M., Suard, F., & Riu, D. (2013). A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. Journal of Power Sources, 241, 680-689.

[143] Newcastle University (n.d.). Roulette Wheel Selection. Retrieved the 8th of March 2018 from http://www.edc.ncl.ac.uk/highlight/rhjanuary2007g02.php

[144] Baker, J. E. (1987, July). Reducing bias and inefficiency in the selection algorithm. In Proceedings of the second international conference on genetic algorithms (pp. 14-21).

[145] Thrum S. & Teichman A. (n.d.). Probabilistic Robotics: Monte Carlos Localization. Retrieved the 8th of March 2018 from http://slideplayer.com/slide/7619607/

[146] Miller, B. L., & Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. Complex systems, 9(3), 193-212.

[147] Syswerda, G. (1989). Uniform crossover in genetic algorithms. In Proceedings of the third international conference on Genetic algorithms (pp. 2-9). Morgan Kaufmann Publishers

[148] Yalcinoz, T., Altun, H., & Uzam, M. (2001). Economic dispatch solution using a genetic algorithm based on arithmetic crossover. In Power tech proceedings, 2001 IEEE Porto (Vol. 2, pp. 4-pp). IEEE.

[149] Herrera, F., Lozano, M., Pérez, E., Sánchez, A. M., & Villar, P. (2002, November). Multiple crossover per couple with selection of the two best offspring: an experimental study with the BLX-α crossover operator for real-coded genetic algorithms. In Ibero-American Conference on Artificial Intelligence (pp. 392-401). Springer, Berlin, Heidelberg.

[150] Herrera, F., Lozano, M., & Verdegay, J. L. (1998). Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. Artificial intelligence review, 12(4), 265-319.

[151] Deb, K., & Beyer, H. G. (1999). Self-adaptive genetic algorithms with simulated binary crossover. Secretary of the SFB 531.

[152] Wright, A. H. (1991). Genetic algorithms for real parameter optimization. In Foundations of genetic algorithms (Vol. 1, pp. 205-218). Elsevier.

[153] Deb, K., & Agrawal, R. B. (1994). Simulated binary crossover for continuous search space. Complex Systems, 9(3), 1-15.

[154] Damousis, I. G., Bakirtzis, A. G., & Dokopoulos, P. S. (2003). Network-constrained economic dispatch using real-coded genetic algorithm. IEEE Transactions on power systems, 18(1), 198-205.

[155] Herrera, F., Lozano, M., & Verdegay, J. L. (1997). Fuzzy connectives based crossover operators to model genetic algorithms population diversity. Fuzzy Sets and Systems, 92(1), 21-30.

[156] Michalewicz, Z. (1996). GAs: Why Do They Work?. In Genetic Algorithms+ Data Structures= Evolution Programs (pp. 45-55). Springer, Berlin, Heidelberg

[157] Shi, Y. (2001). Particle swarm optimization: developments, applications and resources. In evolutionary computation, 2001. Proceedings of the 2001 Congress on (Vol. 1, pp. 81-86). IEEE.

[158] Shi, Y., & Eberhart, R. (1998, May). A modified particle swarm optimizer. In Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on (pp. 69-73). IEEE.

[159] Shi, Y., & Eberhart, R. C. (1998, March). Parameter selection in particle swarm optimization. In International conference on evolutionary programming (pp. 591-600). Springer, Berlin, Heidelberg.

[160] Kennedy, J. (2011). Particle swarm optimization. In Encyclopedia of machine learning (pp. 760-766). Springer US.

[161] Ratnaweera, A., Halgamuge, S. K., & Watson, H. C. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Transactions on evolutionary computation, 8(3), 240-255.

[162] Hu, M., Wu, T., & Weir, J. D. (2012). An intelligent augmentation of particle swarm optimization with multiple adaptive methods. Information Sciences, 213, 68-83

[163] Burger, S., Chaves-Ávila, J. P., Batlle, C., & Pérez-Arriaga, I. J. (2016). The value of aggregators in electricity systems. MIT working paper.

[164] Ikäheimo J, Evens C, Kärkkäinen S. DER Aggregator Business: the Finnish Case. 2010

[165] García González, J. (2010). Unit Commitment of Generation Plants. Electric Power System Modeling for a Low-Carbon Economy. Technologica Research Institute (IIT) at Universidad Pontificia de Comillas, Madrid, 2010.

[166] Widodo, A., Shim, M. C., Caesarendra, W., & Yang, B. S. (2011). Intelligent prognostics for battery health monitoring based on sample entropy. Expert Systems with Applications, 38(9), 11763-11769.

[167] Spotnitz, R. (2003). Simulation of capacity fade in lithium-ion batteries. Journal of Power Sources, 113(1), 72-80.

[168] Sauer, D. U., & Wenzl, H. (2008). Comparison of different approaches for lifetime prediction of electrochemical systems—Using lead-acid batteries as example. Journal of Power sources, 176(2), 534-546.

[169] DiOrio, N., Dobos, A., Janzou, S., Nelson, A., & Lundstrom, B. (2015). Technoeconomic modeling of battery energy storage in SAM (No. NREL/TP-6A20-64641). National Renewable Energy Lab.(NREL), Golden, CO (United States).

[170] Magnor, D., Gerschler, J. B., Ecker, M., Merk, P., & Sauer, D. U. (2009, September). Concept of a battery aging model for lithium-ion batteries considering the lifetime dependency on the operation strategy. In European Photovoltaic Solar Energy Conference (Vol. 24).

[171] Lunz, B., Walz, H., & Sauer, D. U. (2011, September). Optimizing vehicle-to-grid charging strategies using genetic algorithms under the consideration of battery aging. In Vehicle Power and Propulsion Conference (VPPC), 2011 IEEE (pp. 1-7). IEEE.

[172] Badey, Q., Cherouvrier, G., Reynier, Y., Duffault, J. M., & Franger, S. (2011). Ageing forecast of lithium-ion batteries for electric and hybrid vehicles. Curr. Top. Electrochem, 16, 65-79.

[173] Díaz-Gónzalez, F., Sumper, A., Gomis-Bellmunt, O. (2016, May). Cost Models and Economic Analysis. In Energy Storage in Power Systems (pp. 143-162). John Wiley & Sons, Ltd, 2016.

[174] MathWorks (n.d.) App Designer. Retrieved from https://de.mathworks.com/help/matlab/app-designer.html the 23rd of January 2018

[175] Serviço Regional de Estatística dos Açores (2016). Os Açores em Números 2016. Vice-Presidência do Governo, Emprego e Competitividade Empresarial, Angra do Heroísmo (Azores), 2016.

[176] Electricidade dos Açores (2015). Caracterizaçao das redes de transporte e Distribuiçao de energia elétrica da regiao autónoma dos Açores. Ilha Graciosa pgs.109-130

[177] Leclanché (n.d.). Graciosa | Island Microgrid System. Retrieved the 20th February 2018 from http://www.leclanche.com/markets-solutions/graciosa-project/

[178] Younicos (n.d.). Graciosa. Adios Diesel. Ahoy Storage. Retrieved the 20th February 2018 from https://www.younicos.com/case-studies/graciosa/

[179] Electricidade dos Açores (2017). Pocura e Oferta de Energia Eléctrica – dezembro 2017. Ilha Graciosa pg.13

[180] Stenzel, P., Schreiber, A., Marx, J., Wulf, C., Schreieder, M., & Stephan, L. (2017). Renewable energies for Graciosa Island, Azores–Life Cycle Assessment of electricity generation. Energy Procedia, 135, 62-74.

[181] Desarrollo Normativo de la Ley del Sector Eléctrico. (2008). In: 1st ed. Madrid: CNE, vol.10, pp.126-138

[182] Renewables.ninja (n.d.) Homepage. Retrieved from https://www.renewables.ninja/ the 21st of January 201

[183] Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. IEEE transactions on evolutionary computation, 1(1), 67-82.

[184] Lew, D., Bird, L., Milligan, M., Speer, B., Wang, X., Carlini, E. M., ... & Orths, A. (2013, September). Wind and solar curtailment. In International Workshop on Large-Scale Integration of Wind Power Into Power Systems.

[185] FERC (2017). Reports on Demand Response and Advanced Metering. Retrieved the 12th March 2018 from https://www.ferc.gov/industries/electric/indus-act/demand-response/dem-res-adv-metering.asp

[186] Battistelli, C., Agalgaonkar, Y. P., & Pal, B. C. (2017). Probabilistic Dispatch of Remote Hybrid Microgrids Including Battery Storage and Load Management. IEEE Transactions on Smart Grid, 8(3), 1305-1317.

[187] Heleno, M., Matos, M. A., Lopes, J. P., & Iria, J. P. (2014, March). Estimating the flexible residential load using appliances availability. In Power Engineering and Optimization Conference (PEOCO), 2014 IEEE 8th International (pp. 352-357). IEEE.

Appendix

A. Symbols and Abbreviations

If not stated otherwise, the most common symbols and abbreviations in this thesis are defined as follows:

EA	Evolutionary Algorithm
GA	Genetic Algorithm
EP	Evolutionary Programming
ES	Evolutionary Strategies
GP	Genetic Programming
SA	Simulated Annealing
ACO	Ant Colony Optimization
PSO	Particle Swarm Optimization
DE	Differential Evolution
EDA	Estimation of Distribution Algorithm
BBO	Biogeography-Based Optimization
CA	Cultural Algorithm
OBL	Opposition-Based Learning
TS	Tabu Search
AFSA	Artificial Fish Swarm Algorithm
GSO	Group Search Optimized
SFLA	Shuffled Frog Leaping Algorithm
FA	Firefly Algorithm
BFOA	Bacteria Foraging Optimization Algorithm
ABC	Artificial Bee Colony Optimization
GSA	Gravitational Search Algorithm
HS	Harmony Search
TLBO	Teaching Learning Based Optimization
SG	Smart Grid
DG	Distributed Generation
DER	Distributed Energy Resources
RES	Renewable Energy Sources
VRE	Variable Renewable Energy
DR	Demand Response
DSM	Demand Side Management
ESS	Energy Storage Systems
BS	Battery System
EV	Electric Vehicle

V2G	Vehicle to Grid
LP	Linear Programming
MIP	Mixed Integer Programming
NLP	Non-Linear Programming
QP	Quadratic Programming
DP	Dynamic Programming

B. Summary of Scenarios

Scenario	Definition
Base Case	**Demand Profile:** Average **Wind Profile:** Average **PV Profile:** Average **Battery:** LTO Leclanché as it is in Graciosa **Degradation Model:** Yes **Others:** -
Spring Demand	**Demand Profile:** Spring characteristic profile **Wind Profile:** Average **PV Profile:** Average **Battery:** LTO Leclanché as it is in Graciosa **Degradation Model:** Yes **Others:** -
Summer Demandt	**Demand Profile:** Summer characteristic profile **Wind Profile:** Average **PV Profile:** Average **Battery:** LTO Leclanché as it is in Graciosa **Degradation Model:** Yes **Others:** -
Winter Demand	**Demand Profile:** Winter characteristic profile **Wind Profile:** Average **PV Profile:** Average **Battery:** LTO Leclanché as it is in Graciosa **Degradation Model:** Yes **Others:** -
Autumn Demand	**Demand Profile:** Autumn characteristic profile **Wind Profile:** Average **PV Profile:** Average **Battery:** LTO Leclanché as it is in Graciosa **Degradation Model:** Yes **Others:** -
Highest Wind Energy	**Demand Profile:** Average **Wind Profile:** Day with highest wind energy produced in year considered **PV Profile:** PV profile from that day **Battery:** LTO Leclanché as it is in Graciosa **Degradation Model:** Yes **Others:** -
Lowest Wind Energy	**Demand Profile:** Average **Wind Profile:** Day with lowest wind energy produced in year considered **PV Profile:** PV profile from that day **Battery:** LTO Leclanché as it is in Graciosa **Degradation Model:** Yes **Others:** -
Highest Wind Variability	**Demand Profile:** Average

	Wind Profile: Day with highest wind variability in the year considered **PV Profile:** PV profile from that day **Battery:** LTO Leclanché as it is in Graciosa **Degradation Model:** Yes **Others:** -
Base Case No Deg	**Demand Profile:** Average **Wind Profile:** Average **PV Profile:** Average **Battery:** LTO Leclanché as it is in Graciosa **Degradation Model:** No **Others:** -
Demand Response	**Demand Profile:** Winter Demand **Wind Profile:** Average **PV Profile:** Average **Battery:** LTO Leclanché as it is in Graciosa **Degradation Model:** Yes **Others:** Demand Response
Doubled Battery	**Demand Profile:** Winter Demand and Average Demand **Wind Profile:** Average and Highest Variability **PV Profile:** Average and Highest Variability **Battery: 2x**LTO Leclanché as it is in Graciosa multiplied **Degradation Model:** Yes **Others:** -

C. Leclanché LTO Battery Datasheet

Example of product: TiRack 63
Unbeatable facts of technology

Security	Embedded String-Controller External / internal interfaces Protokoll ads-tec	Integrated Ethernet / CAN Master or slave
String configuration	String format Number of battery modules connected in series Number of string controllers per triple rack Number of string controllers per double rack Number of string shutoff modules »bipolar«	19" triple rack 15 1 - 1
Connection values	Battery voltage »empty« Battery voltage »full«	510V DC 810V DC
Currents	Operating current (up to 2C charging / 2C discharging) Maximum string current	180A 300A
Battery system	Cell chemistry Cell capacity Nominal system capacity Specified cycles 1C/1C @23°C at 100% DOD Expected calendar lifespan	Lithium-titanate 30Ah 63kWh 15.000 20 years
Environmental conditions	Temperature range (long life < 2C) Protection class Humidity	10 to 30 °C IP20 < 90%, non-condensing
Guarantee	Limitation period for claims due to defects	24 months
Function and durability	In combination with a Big-LinX service contract	Up to 10 years
Standards	EMV: EN 61000-6-2:2006; EN 61000-6-3:2007 + A1:2011; EN 61000-6-4:2007 + A1:2011; EN 55024:2010; EN 55022:2010; EN 61000-4-2:2009; EN 61000-4-3:2006 + A1:2008 + A2:2010; EN 61000-4-4:2004 + A1:2010; EN 61000-4-5:2006; EN 61000-4-6:2009; EN 61000-4-11:2004; EN 55016-2-1:2009 + A1:2011; EN 55016-2-3:2010 + A1:2010	
	Safety (functional and electrical): EN 61010-1:2010; EN 50272-2:2001	
	Transport: UN38.3 Transport directive for lithium batteries	
	Isolation and fire protection: DIN EN 60664-1, VDE 0110-1, DIN VDE 0298-4, VDE 0471 DIN EN 60695-11-10 und -20	

Retrieved from: http://www.leclanche.com/technology-products/products/titanate-racks/