

Proyecto Fin de Máster Máster en Ingeniería Industrial

Operación Óptima de Redes Inteligentes con gran penetración de Generación Renovable

Autor: Alfonso Aycart Lazo

Tutor: Alejandro Marano Marcolini

**Dpto. Ingeniería Eléctrica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla**

Sevilla, 2019



Proyecto Fin de Máster
Máster en Ingeniería Industrial

Operación Óptima de Redes Inteligentes con gran penetración de Generación Renovable

Autor:

Alfonso Aycart Lazo

Tutor:

Alejandro Marano Marcolini

Profesor Contratado Doctor

Dpto. Ingeniería Eléctrica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019

Proyecto Fin de Máster: Operación Óptima de Redes Inteligentes con gran penetración de
Generación Renovable

Autor: Alfonso Aycart Lazo
Tutor: Alejandro Marano Marcolini

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

Me gustaría comenzar agradeciéndole a Alejandro Marano, mi tutor en este Trabajo de Fin de Máster, su disponibilidad total y su ayuda para conseguir llevar a buen puerto este proyecto. Ha sabido guiarme incluso a distancia y le estoy muy agradecido por ello.

Igualmente, me gustaría agradecer a mis supervisores en Energías De Portugal por ver siempre con buenos ojos que realizase este proyecto al mismo tiempo que desarrollaba mi actividad profesional allí y por las facilidades que me han puesto a la hora de volver a Sevilla para presentarlo. Se lo agradezco particularmente a Mariana Gonçalves.

Por supuesto, mis compañeros y amigos del Trainee Program de EDP merecen una mención especial por hacerme más llevaderos estos últimos meses de trabajo con su presencia.

Por último, y como no podría ser de otra manera, se lo agradezco enormemente a mis padres. Me han apoyado a lo largo de mis años de universidad y sin ellos nunca habría podido tener tantas experiencias de estudio en el extranjero. Si no fuese por su constante insistencia, seguramente este proyecto seguiría sin estar terminado. Gracias.

Alfonso Aycart Lazo
Estudiante de Máster de Ingeniería Industrial

Sevilla, 2019

Resumen

En la actualidad, el sector energético está experimentando importantes cambios. La liberalización de los mercados de energía, los intentos de dar respuesta a los problemas asociados al cambio climático y los avances tecnológicos son solo parte de los factores que lo hacen evolucionar. Pueden distinguirse tendencias comunes en el sector: la reducción del uso de combustibles fósiles y la descarbonización, la electrificación de los consumos, el aumento de producción de energías renovables, la aplicación de nuevas tecnologías como la inteligencia artificial, la creciente sofisticación del perfil de los consumidores y el desarrollo de sistemas almacenamiento y de vehículos eléctricos son solo algunos ejemplos.

El sector energético es reconocido globalmente como un elemento esencial en la búsqueda de un futuro más sostenible. Una forma que este tiene de dar respuesta a los desafíos que se plantean actualmente es el desarrollo y la implementación de redes de distribución de electricidad inteligentes, las conocidas como *smart grids*. El conjunto de tecnologías que las forman puede ayudar a optimizar los recursos reduciendo los costes asociados, a la vez que se consigue integrar eficazmente los nuevos sistemas energéticos tales como la generación descentralizada o los sistemas de almacenamiento.

En el presente proyecto se ha realizado un programa informático basado en Python que recrea el comportamiento de una red de distribución de media tensión. Se ha creado un modelo de una red de referencia en Europa que incluye diversas cargas y generadores, sistemas de producción eléctrica renovables y sistemas de almacenamiento de energía, entre otros elementos. Igualmente, se ha configurado un problema de optimización matemática MINLP que actúa como sistema de gestión de energía para resolver el problema de Asignación de Unidades y de Despacho Económico en la red. El objetivo de este programa es hallar para un periodo de 24 horas los valores óptimos de las variables de la red de forma que se reduzcan los costes y se mantengan las condiciones operacionales deseadas. Se incluye un modelo estocástico para gestionar las incertidumbres asociadas a los precios del mercado eléctrico y a la generación renovable. El modelo se ha programado en Pyomo y se utiliza el solver BONMIN (al que se accede a través del servidor NEOS) para resolver el problema de optimización.

Los diversos casos estudiados presentan las funcionalidades del programa. El modelo permite las desconexión de las cargas industriales para adaptar la demanda e incluye la posibilidad de establecer niveles de reserva de potencia y de limitar la potencia importada por la red. Un caso en el que existe una gran producción de energías renovables es igualmente estudiado. En todos estos casos el programa ha hallado soluciones óptimas y las variables siguen el comportamiento esperado.

La solución del problema estocástico asociado se ha realizado mediante una aproximación del problema relajado NLP. A pesar de ello, la solución es similar a la del problema determinista equivalente.

El modelo podría ampliarse en el futuro para incluir un enlace entre las dos subredes que componen la red o simular la presencia de vehículos eléctricos, entre otras posibilidades.

Abstract

The energy sector is currently undergoing important changes. Energy market liberalisation, the attempts to fight climate change and the technological advances are only a small part of the factors that push its development. Certain common trends can be identified in the market: decarbonisation and fossil fuel reduction, electrification of energy consumption, increase in renewable energy generation, the use of new technologies such as artificial intelligence, customer's progressive sophistication and the development of energy storage systems and electric vehicles are just a few examples.

The energy sector is globally recognised as an essential element in the search of a more sustainable future. One of its tools to face the current challenges is the development and the implementation of smart electricity distribution networks, also known as smart grids. The technologies around which they are built can help optimising resources and reducing associated costs, as well as effectively integrate the new energy systems such as distributed generation or energy storage systems.

In this project a computer program that models the behaviour of medium voltage distribution network has been created in Python. The network modelled is a reference network for Europe that includes different loads and generators, renewable energy generators and energy storage systems, amongst other elements. It has also been defined a MINLP optimization problem that acts as an energy management system for solving the Unit Commitment and Economic Dispatch problems. The aim of the program is to find the optimal values of the variables in order to minimize costs and maintain the operational conditions in the network for a 24 hour period. It includes a stochastic model used to manage the uncertainties associated to energy market prices and renewable generation. The model has been coded in Pyomo and the solver BONMIN is accessed through the server NEOS to solve the optimization problem.

The different cases analysed present the various functionalities of the program. The model allows load curtailment in industries to adjust demand and it includes the possibility of establishing power reserves and limiting the power imported by the network. A case with increased renewable energy generation is also studied. In all these configurations, the program has found optimal solutions and the variables follow the expected behaviour.

The stochastic problem has been solved through an approximation of the relaxed NLP problem. In spite of this, the solution found is close to that of the equivalent deterministic problem.

The model could be further developed to include a link between the two separate networks or to simulate the presence of electric vehicles, amongst other possibilities.

Índice Abreviado

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
<i>Notación</i>	XI
<i>Acrónimos</i>	XIII
1 Introducción	1
1.1 Transición Energética	2
1.2 Energías Renovables	4
1.3 Redes de Distribución Eléctrica Inteligentes	6
1.4 Gestión de la Demanda Energética	10
1.5 Presentación del Problema	11
2 Metodología	13
2.1 Proceso de Investigación y Análisis	13
2.2 Problemas de Optimización Matemática	14
2.3 Pyomo	16
3 Caso de Estudio	21
3.1 Red de Distribución de Referencia	21
3.2 Problema de Optimización	25
3.3 Estructura del Programa	34
3.4 Importación de Datos	36
3.5 Estructura y Datos del Problema Estocástico	39
4 Resultados del Modelo	43
4.1 Caso Base	43
4.2 Limitación de Potencia Importada	45
4.3 Limitación de Potencia Importada y Coste de Desconexión Reducido	47
4.4 Alta Generación Renovable	48
4.5 Restricción de Reservas	49
4.6 Resultados Estocásticos	51
5 Conclusiones y Líneas Futuras	53

Apéndice A	Modelo Determinista	55
Apéndice B	Estructura del problema estocástico	83
	<i>Índice de Figuras</i>	95
	<i>Índice de Tablas</i>	97
	<i>Índice de Códigos</i>	99
	<i>Bibliografía</i>	101

Índice

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
<i>Notación</i>	XI
<i>Acrónimos</i>	XIII
1 Introducción	1
1.1 Transición Energética	2
1.2 Energías Renovables	4
1.3 Redes de Distribución Eléctrica Inteligentes	6
1.4 Gestión de la Demanda Energética	10
1.5 Presentación del Problema	11
1.5.1 Alcance y objetivos del proyecto	12
2 Metodología	13
2.1 Proceso de Investigación y Análisis	13
2.1.1 Documentación	13
2.1.2 Modelización	13
2.1.3 Obtención de datos	14
2.1.4 Análisis de resultados	14
2.2 Problemas de Optimización Matemática	14
2.2.1 Algoritmos de resolución de problemas MINLP	16
2.3 Pyomo	16
2.3.1 Elementos de Pyomo	17
2.3.2 Solvers	19
2.3.3 Extensiones: PySP para modelización estocástica	20
3 Caso de Estudio	21
3.1 Red de Distribución de Referencia	21
3.1.1 Líneas de la red	21
3.1.2 Cargas de la red	22
3.1.3 Generadores de la red	23
3.2 Problema de Optimización	25
3.2.1 Asignación de unidades y Despacho económico	26
3.2.2 Formulación del problema	27
Sets	27

Parámetros	27
Variables	28
Función objetivo	28
Restricciones	28
Restricciones auxiliares	30
Formulación completa	32
Problema estocástico	32
3.2.3 Recapitulación	34
3.3 Estructura del Programa	34
3.4 Importación de Datos	36
3.4.1 Datos deterministas	36
Datos de la red	37
Costes	37
Generación renovable	37
Baterías	38
3.5 Estructura y Datos del Problema Estocástico	39
3.5.1 Escenarios de precios	39
3.5.2 Escenarios de generación eólica	39
3.5.3 Escenarios de generación fotovoltaica	41
3.5.4 Estructura del problema estocástico	41
4 Resultados del Modelo	43
4.1 Caso Base	43
4.2 Limitación de Potencia Importada	45
4.3 Limitación de Potencia Importada y Coste de Desconexión Reducido	47
4.4 Alta Generación Renovable	48
4.5 Restricción de Reservas	49
4.6 Resultados Estocásticos	51
5 Conclusiones y Líneas Futuras	53
Apéndice A Modelo Determinista	55
Apéndice B Estructura del problema estocástico	83
<i>Índice de Figuras</i>	95
<i>Índice de Tablas</i>	97
<i>Índice de Códigos</i>	99
<i>Bibliografía</i>	101

Notación

$^{\circ}\text{C}$	Grados Celsius
CO_2	Dióxido de carbono
L	Longitud
R	Resistencia eléctrica
X	Reactancia
X_{sh}	Reactancia shunt
C	Capacidad eléctrica
n	Prefijo que indica un factor de 10^{-9}
k	Prefijo que indica un factor de 10^3
M	Prefijo que indica un factor de 10^6
G	Prefijo que indica un factor de 10^9
T	Prefijo que indica un factor de 10^{12}
F	Faradios
h	Horas
Hz	Hercios
m	Metro
Ω	Ohmios
p.u.	Sistema por unidad
V	Voltio
VA	Voltiamperio
W	Vatio
Wh	Vatio hora
s.a.	Sujeto a
\mathbb{R}	Cuerpo de los números reales
\mathbb{C}	Cuerpo de los números complejos
\mathbb{Z}	Cuerpo de los números enteros
sen	Función seno
cos	Función coseno
\leq	Menor o igual
\geq	Mayor o igual
Δ	Incremento
\in	Pertenece a
\forall	Para todo
$\sum_{i=1}^n$	Sumatorio para todo i desde 1 hasta n
$\cos \varphi$	Factor de potencia
c^T	Vector de costes traspuesto

t	Índice de las horas
i	Índice de los nodos
g	Índice de los generadores
pv	Índice de los generadores fotovoltaicos
w	Índices de los aerogeneradores
bs	Índices de las baterías
ni	Índices de las cargas residenciales
ind	Índices de las cargas industriales
C_g^A	Coste fijo del generador g
C_g^B	Coste proporcional a la potencia del generador g
C_{ind}	Coste de desconexión de la industria ind
P_{max}	Potencia activa máxima
Q_{max}	Potencia reactiva máxima
S_{max}	Potencia aparente máxima
$p_{g,t}$	Potencia activa del generador g en la hora t
$q_{g,t}$	Potencia reactiva del generador g en la hora t
$pe_{ind,t}$	Potencia activa máxima posible de la industria ind para la hora t
$qe_{ind,t}$	Potencia reactiva máxima posible de la industria ind para la hora t
mp_t	Potencia total importada por la red en la hora t
$\eta_{c,bs}$	Rendimiento de carga de la batería bs
$\eta_{d,bs}$	Rendimiento de descarga de la batería bs
$soc_{bs,t}$	Estado de carga de la batería bs en la hora t
G_{ij}	Susceptancia de la línea conectando los nodos i y j
B_{ij}	Conductancia de la línea conectando los nodos i y j
ur_t	Reserva de subida de potencia a mantener en la hora t
dr_t	Reserva de bajada de potencia a mantener en la hora t
ϕ_t	Porcentaje máximo de potencia reactiva importada respecto a la potencia activa importada
π_t	Precio de la electricidad importada en la hora t
$u_{g,t}$	Estado del generador g en la hora t
$ls_{ind,t}$	Porcentaje de conexión de la industria
$V_{i,t} \angle \theta_{i,t}$	Fasor del voltaje en el nodo i para la hora t ; amplitud del voltaje V y ángulo θ
$p \in \Omega_p$	Escenario de precios p dentro del conjunto de escenarios de precios
μ_p	Probabilidad de ocurrencia del escenario p
$s \in \Omega_s$	Escenario de generación solar s dentro del conjunto de escenarios generación solar
μ_s	Probabilidad de ocurrencia del escenario s
$v \in \Omega_v$	Escenario de generación eólica v dentro del conjunto de escenarios generación eólica
μ_v	Probabilidad de ocurrencia del escenario v

Acrónimos

AMI	Infraestructura de Medida Avanzada (Advanced Metering Infrastructure)
BB	Algoritmo de Ramificación y Acotamiento (Branch-and-Bound)
BS	Sistema de Baterías
CCS	Captura y Almacenamiento de Carbono (Carbon Capture and Storage)
CIGRE	Consejo Internacional de Grandes Redes Eléctricas
CHP	Cogeneración (Combined Heat and Power)
CHPD	Cogeneración basada en Diesel
CHPFC	Cogeneración basada en Pila de Combustible
DC	Corriente Continua
DSM	Gestión de la Demanda (Demand-Side Management)
DSO	Operador de la Red de Distribución (Distribution System Operator)
ED	Problema de Despacho Económico (Economic Dispatch problem)
EDP	Energías De Portugal
EE	Eficiencia Energética
EMS	Sistema de Gestión de Energía (Energy Management System)
ETSI	Escuela Técnica Superior de Ingeniería
FC	Pila de combustible (Fuel Cell)
IPCC	Grupo Intergubernamental de Expertos sobre el Cambio Climático (Intergovernmental Panel on Climate Change)
LP	Programación Lineal
MINLP	Programación Entera Mixta No Lineal (Mixed-Integer Non-Linear Programming)
MIP	Programación Entera Mixta Lineal (Mixed-Integer Programming)
NLP	Programación No Lineal (Non-Linear Programming)
OA	Algoritmo de Aproximación Exterior (Outer-Approximation algorithm)
OCDE	Organización para la Cooperación y el Desarrollo Económicos

PCC	Punto de Acoplamiento Común (Point of Common Coupling)
PIB	Producto Interior Bruto
PV	Panel Fotovoltaico
QP	Programación Cuadrática (Quadratic Programming)
SG	Redes de distribución inteligentes (Smart Grids)
TOU	Basado en el tiempo de utilización (Time-Of-Use)
UC	Problema de Asignación de Unidades (Unit Commitment problem)
UE	Unión Europea

1 Introducción

En los últimos años, conceptos como cambio climático, desarrollo sostenible y ecologismo han ganado relevancia en los debates sociales y políticos, tanto a nivel local como internacional.

De igual manera, la situación demográfica actual se ve marcada por el crecimiento de la población mundial y el consecuente aumento del consumo eléctrico, en 2017 la demanda eléctrica global creció un 3,1 % (frente al 2,1 % de la demanda energética). El aumento del consumo se produce especialmente en los países en vías de desarrollo como China e India, representando estos dos países el 70 % de dicho crecimiento [18]. Esta tendencia se prevé que continúe, con el consumo de China prácticamente triplicando al de la Unión Europea en 2040 de mantenerse las condiciones actuales (ver figura 1.2) [21].

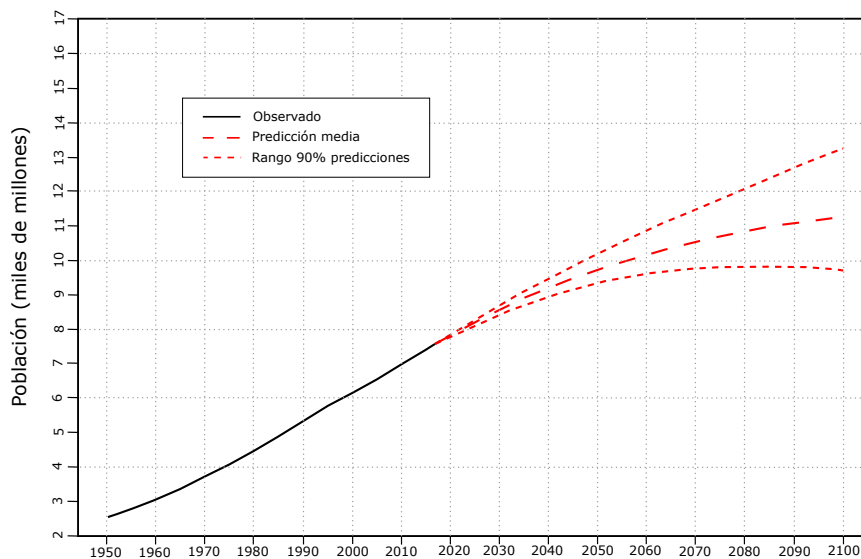


Figura 1.1 Predicciones de crecimiento de la población mundial [13].

En este contexto, se suceden numerosos estudios que advierten sobre las consecuencias que puede tener el cambio climático en nuestro planeta e instan a la puesta en marcha de medidas concretas para paliar sus efectos adversos [5] (ver figura 1.3). Todo ello ha desembocado en una apuesta a nivel internacional por la sostenibilidad y la lucha contra el calentamiento global, como se aprecia en el gran peso que han ganado estos aspectos en los Objetivos de Desarrollo Sostenible [12][46] y en la celebración de numerosas cumbres globales en las que se intenta con un éxito moderado establecer acuerdos y protocolos [2]. Del mismo modo, esta inercia reguladora ocasiona que cada vez más medidas ecológicas sean asimiladas por las sociedades occidentales: la economía circular y

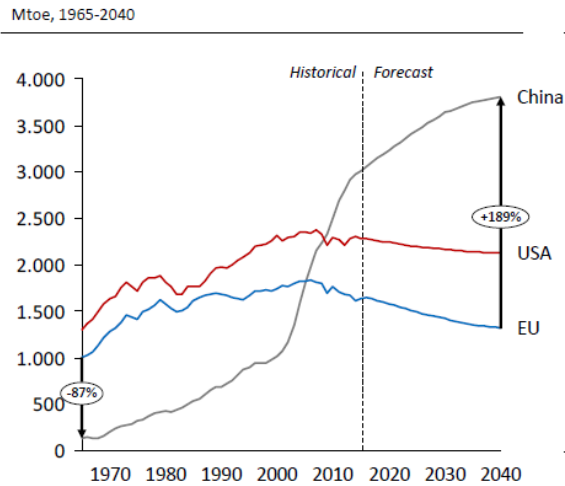


Figura 1.2 Predicciones de crecimiento de la demanda de energía primaria [21].

el reciclaje, el impulso de pequeñas comunidades sostenibles o incluso el cambio en los hábitos de alimentación por razones medioambientales han cobrado relevancia en los últimos años [8][22]. Han llamado particularmente la atención las manifestaciones de estudiantes contra el cambio climático que tuvieron lugar el día 15 de mayo de 2019 internacionalmente y que denotan una importante preocupación por parte de estos temas de una gran parte de la sociedad [3].

Sin embargo, estas iniciativas se demuestran insuficientes a día de hoy para cumplir los objetivos marcados con el fin de limitar el ascenso de la temperatura del planeta por encima de 2 °C (o 1,5 °C como especifica el último informe del IPCC [5]).

Se plantea por lo tanto necesario el desarrollo de los sistemas eléctricos con la inclusión de generación renovable, la descentralización y el auto-consumo, así como la integración en la red de nuevos actores como los vehículos eléctricos. Para optimizar estos recursos será indispensable el uso de la tecnología. Igualmente, la proliferación de sensores y de herramientas como la inteligencia artificial y el análisis de datos masivos supondrán un gran paso en este sentido [54]. Es en este espacio donde las redes de distribución eléctrica inteligentes se vuelven más relevantes y necesarias.

1.1 Transición Energética

Las formas de energía convencional son generalmente finitas, contaminantes (causando incluso problemas de salud [16], que son aún más notorios en países como China) o generan residuos, además de ocasionar el conocido efecto invernadero que provoca un aumento de la temperatura del planeta. Con el objetivo de reducir estos efectos del cambio climático, numerosos países están desarrollando hojas de ruta a tal efecto. Esta adaptación de los modelos económicos, de consumo y de generación de energía es lo que se denomina como transición energética.

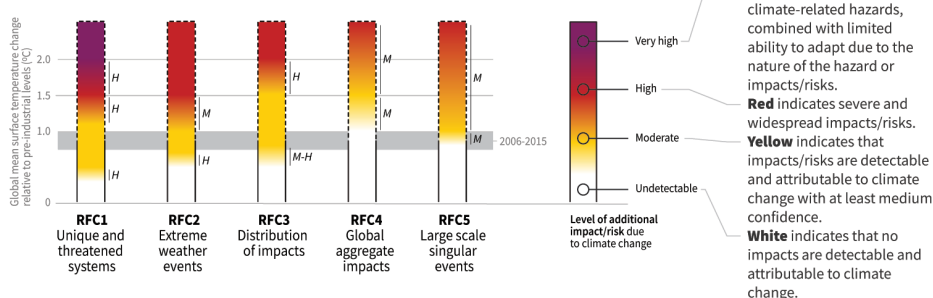
Las tendencias que se observan en el panorama energético global actual, y de una forma más acentuada en los países de la OCDE, y que se prevén ganen mayor importancia en el futuro son las siguientes:

- Aumento de la electrificación, ocasionado por el aumento de la tecnología tanto en los hogares como en la industria y por la descarbonización de la economía con el objetivo de reducir las emisiones de gases de efecto invernadero.
- Inversiones en la generación de energías renovables y acelerado desarrollo tecnológico de las mismas, especialmente de la energía eólica, solar y de las técnicas de almacenamiento de energía.

How the level of global warming affects impacts and/or risks associated with the Reasons for Concern (RFCs) and selected natural, managed and human systems

Five Reasons For Concern (RFCs) illustrate the impacts and risks of different levels of global warming for people, economies and ecosystems across sectors and regions.

Impacts and risks associated with the Reasons for Concern (RFCs)



Impacts and risks for selected natural, managed and human systems

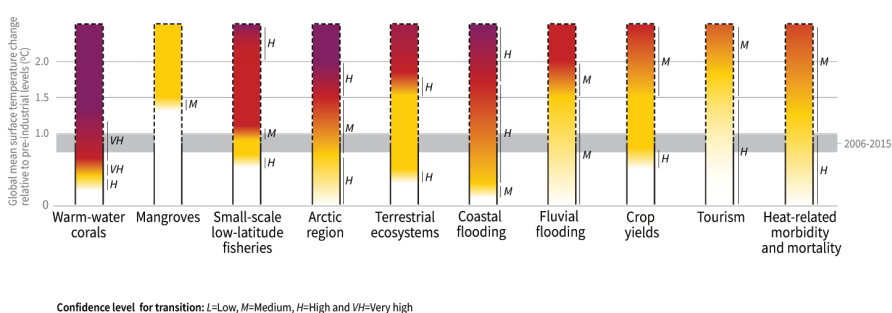


Figura 1.3 Riesgos e impactos del cambio climático [36].

- Mayor interés por medidas de eficiencia energética y reducción del consumo.
- Digitalización y optimización de los procesos de producción y gestión de la energía.
- Aumento de la movilidad eléctrica, especialmente en zonas urbanas.
- Sofisticación de los consumidores finales de energía, pivotando hacia un perfil de *prosumidor*.

Estas características están alineadas con lo que se consideran los tres pilares de la transición energética, los cuales ya se reconocían al principio de la década [7]:

- Reducción del consumo energético: es necesario actuar en los tres principales sectores consumidores de energía (residencial, industrial y transportes). Deben promoverse medidas de eficiencia energética (equipos más eficientes en la industria y el sector residencial), sistemas de recuperación de energía, el aislamiento térmico de edificios y el cambio hacia la movilidad eléctrica (ver figura 1.4). La promoción de modelos de producción y consumo basados en la economía circular también se muestra crucial a efectos de la reducción y revalorización de los residuos.
- Electrificación de la economía, con especial atención al desarrollo de energías renovables y sistemas de almacenamiento de energía. También se plantea necesario invertir en el desarrollo de tecnologías relacionadas con la captura y el almacenamiento de CO_2 (CCS), con el fin de

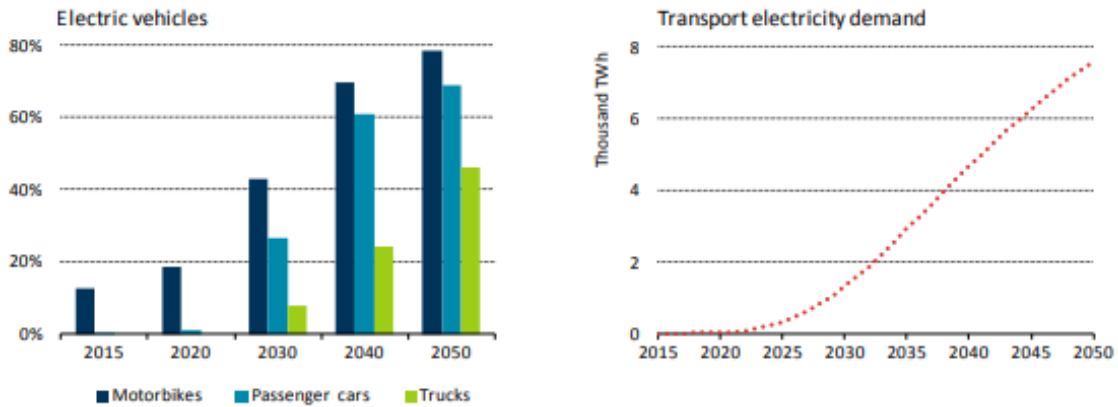


Figura 1.4 Evolución necesaria del vehículo eléctrico para alcanzar el objetivo de 2 °C según [17].

superar los desafíos técnicos que existen actualmente y reducir sus costes [48]. Sin embargo, recientes estudios alertan del limitado alcance que las tecnologías CCS pueden tener en el objetivo de limitar el aumento de la temperatura del planeta [20].

- Desarrollo de las infraestructuras eléctricas y de las interconexiones entre países. A estos efectos, la promoción de las redes de distribución eléctricas inteligentes permitirá optimizar el uso de los recursos energéticos e integrar al consumidor en la cadena.

Siguiendo este razonamiento, la Comisión Europea ha establecido una hoja de ruta con la que pretende definir visiones comunes para estados e industria y fijar objetivos específicos en cuanto a mitigación del cambio climático. Así, para el año 2030, se espera conseguir un aumento de la eficiencia energética en la UE del 23,5 %, que el 32 % de la energía final consumida proceda de fuentes renovables y que las emisiones de CO_2 se reduzcan en un 40 % respecto a los niveles de 1990 (actualmente se proyecta que este número aumente hasta el 60 % en 2050) [19]. Dichos objetivos deben ser compartidos por los estados miembros, los cuales tienen que implementar medidas concretas para alcanzarlos. Igualmente, se han llevado a cabo acciones concretas a nivel europeo, como el establecimiento de un mercado de carbono internacional que ocasiona un aumento del precio del CO_2 que hace más atractiva la inversión en tecnologías limpias.

Todas estas medidas de desarrollo sostenible ocasionan importantes cambios en la industria y especialmente en el sector energético. Una de las consecuencias relevantes es el desacople que se produce entre el PIB y el consumo de energía en los países de la OCDE.

1.2 Energías Renovables

Según lo explicado anteriormente, la generación de energía renovable se contorna como un elemento clave para la transición energética mundial. Estas fuentes de energía son virtualmente inagotables al provenir de medios naturales con capacidad de regeneración, y generalmente tienen un impacto limitado en el medio ambiente. Asimismo, debido a su carácter autóctono, no generan dependencia de potencias extranjeras como suele ocurrir con los combustibles fósiles.

Estas fuentes de energía pueden clasificarse según su origen:

- **Hidráulica:** consiste en aprovechar la energía cinética y potencial de las corrientes y saltos de agua para generar electricidad mediante el funcionamiento de turbinas. Tiene un rendimiento energético muy elevado, el precio de la energía es reducido, su vida útil es muy larga y es una fuente de energía limpia. Igualmente, los embalses permiten el almacenamiento de agua para abastecimiento humano y se utilizan como método de regulación del caudal de los ríos. También es posible transformar las grandes centrales hidráulicas en reversibles mediante la

implementación de sistemas de bombeo, los cuales pueden ser usados como almacenamiento energético.

Sin embargo, esta tecnología puede causar importantes daños ambientales a los ecosistemas, especialmente debido a la construcción de grandes embalses que inundan importantes extensiones de terreno.

- **Eólica:** se aprovecha la energía cinética del viento para mover turbinas y generar electricidad. Estas tecnologías pueden ser instaladas en zonas muy variadas con escaso impacto medioambiental. En los últimos años su coste de instalación se ha reducido al mismo tiempo que aumentaban los rendimientos, lo que las ha llevado a ser competitivas con respecto a las fuentes convencionales [32].
- **Solar:** la fuente de energía en este caso es la radiación electromagnética del Sol. Esta radiación se captura mediante diversas tecnologías, entre las que se incluyen los paneles fotovoltaicos (energía solar fotovoltaica) y los colectores solar (energía solar térmica). Según el medio utilizado, el resultado puede ser electricidad o energía térmica. La producción a partir de esta fuente es muy predecible, y los precios se han reducido enormemente en la última década debido a los avances tecnológicos y las economías de escala, por lo que han llegado a alcanzar la paridad de red en algunas regiones [32]. Las expectativas de evolución de esta tecnología han sido por lo tanto constantemente revisadas al alza.

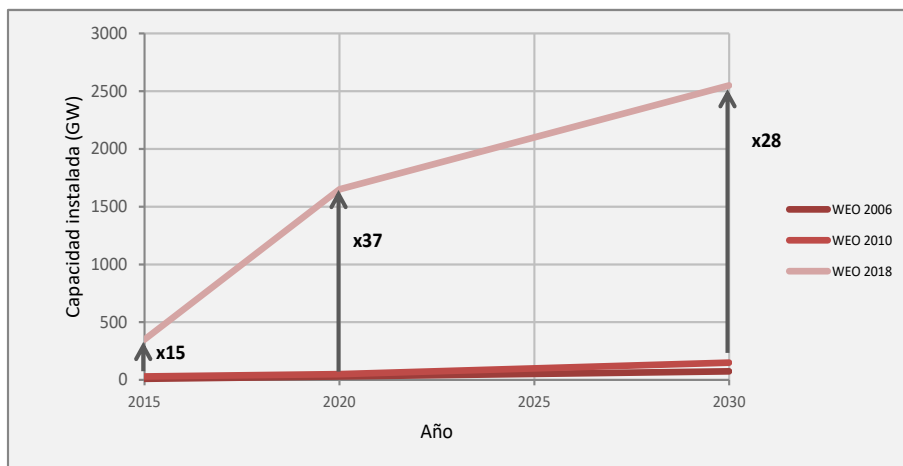


Figura 1.5 Actualización de las previsiones de capacidad instalada de energía solar [21].

- **Geotérmica:** consiste en extraer el calor natural del interior de la tierra. Existen reservorios de variada temperatura y profundidad, por lo que estas tecnologías se clasifican entre alta, media, baja y muy baja temperatura. Producen mínimos residuos y son una medida de gran ahorro energético. Por otro lado, pueden causar contaminación de aguas y tan solo está disponible en determinados lugares, aunque en aquellos en los que los recursos son de fácil acceso su coste es muy reducido.
- **Marina:** puede considerarse como un tipo de energía hidráulica, aunque la anterior suele estar asociada con el aprovechamiento de los recursos fluviales y esta hace referencia exclusivamente a los mares y océanos. Hay variadas tecnologías aunque en general se encuentran en fase de desarrollo tecnológico (excepto la energía mareomotriz, que técnicamente no difiere mucho de la energía hidráulica convencional). Puede obtenerse energía de las olas, de las mareas, de las corrientes oceánicas, de la diferencia térmica entre las masas de agua a distintas profundidades y del gradiente de salinidad. Estas fuentes de energía muestran unos recursos potenciales muy importantes, siendo la energía térmica oceánica la de mayores recursos con 44,000 TWh/año (aproximadamente el doble del consumo energético mundial por año) [38].

- **Biomasa:** se utiliza la materia orgánica procedente de algún proceso biológico. Puede aprovecharse directamente mediante la combustión, o transformarse en otras sustancias que pueden ser utilizadas posteriormente, como los biocarburantes. Estos últimos se emplean como aditivos o sustitutivos de los combustibles convencionales. Pese a que estas fuentes de energía generan emisiones de CO_2 durante la combustión, son consideradas neutras respecto al carbono, ya que las emisiones generadas son compensadas por aquellas que fueron captadas por las plantas durante la fotosíntesis. Sin embargo, pueden tener un gran impacto ambiental si se dedican importantes masas forestales para la producción de combustible.

El desarrollo de estas tecnologías se han incrementado mucho en los últimos años, como ya se ha mencionado anteriormente. En la figura 1.5 se puede apreciar como se han actualizado las previsiones de la energía solar con el transcurso de los años. Las expectativas de instalación de energía eólica era mucho mayores inicialmente. Sin embargo, y pese a haber sido triplicadas del año 2010 al 2018, han sido sobrepasadas por las de energía solar [21].

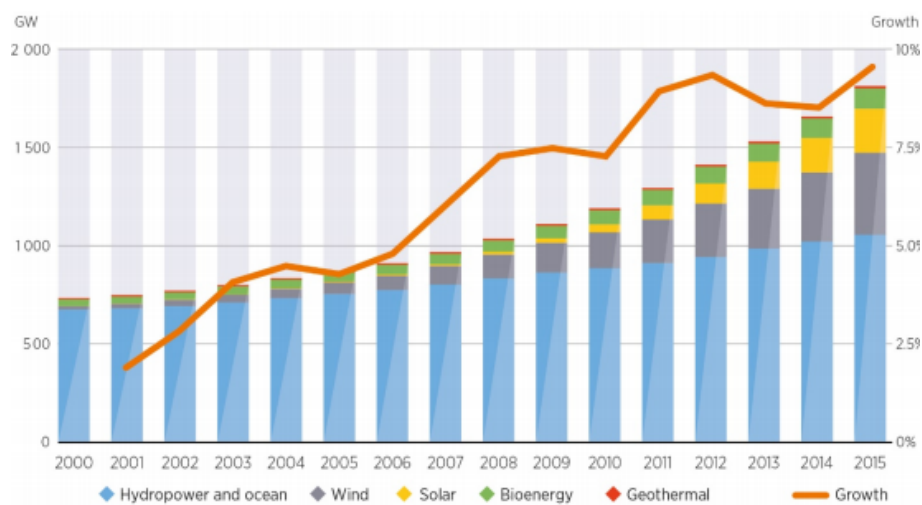


Figura 1.6 Evolución de la potencia renovable instalada a nivel mundial [17].

Las consecuencias directas de esta evolución son diversas. El perfil de generación de electricidad se verá modificado. La carga de base será muy inferior por el aumento del auto-consumo y durante las horas centrales del día la demanda será asegurada principalmente mediante la energía solar, hidráulica y eólica. Estas últimas también estarán presente durante los otros periodos del día, acopladas con generación a partir de biomasa y las centrales de gas más eficientes para responder a los picos de demanda. Todo esto se complementa con sistemas de almacenamiento de energía y de generación descentralizada en los hogares, lo que plantea importantes retos en el sector energético. La naturaleza imprevisible de alguna de estas fuentes de energía va a ocasionar un cambio en la manera de gestionar la red eléctrica.

Las llamadas redes de distribución eléctrica inteligentes (*smart grids*) se revelan así como el elemento clave para el futuro de los sistemas eléctricos.

1.3 Redes de Distribución Eléctrica Inteligentes

Los sistemas eléctricos pueden dividirse en tres partes diferenciadas: la generación, el transporte y la distribución.

La generación de energía se produce generalmente en grandes centrales, de donde parte la electricidad hacia los centros de consumo mediante la red de transporte de alta tensión (voltajes de 220 kV y superiores). Posteriormente, la tensión se reduce para distribuir la electricidad en

los medios rurales (media tensión) y urbanos (baja tensión) de forma que llegue directamente al consumidor. La medición del consumo se hace mediante contadores en los puntos de conexión. En este proyecto se tratarán las redes de distribución, concretamente las consideradas redes inteligentes.

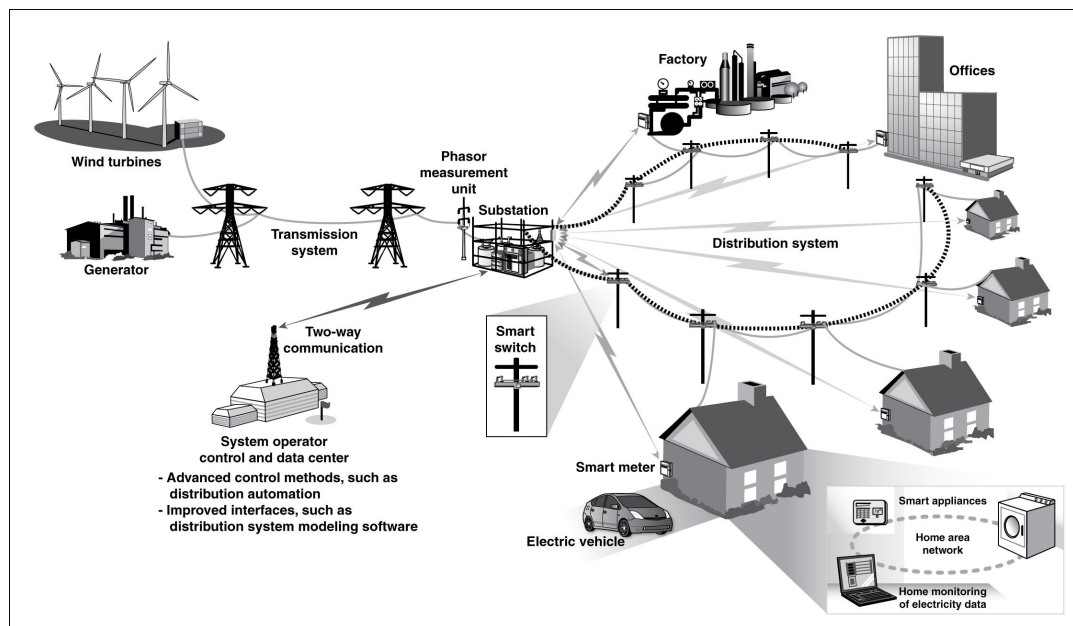


Figura 1.7 Esquema general de una red de distribución eléctrica inteligente [15].

Las redes de distribución eléctrica inteligentes no son una tecnología en sí mismas, sino que deben considerarse más bien como una combinación de tecnologías de diferentes ámbitos, incluidas la electrónica de potencia, las telecomunicaciones y la gestión. Las redes suelen contar con sistemas de generación distribuida y de almacenamiento energético, con líneas de transporte y distribución de electricidad, con un conjunto de sensores (contadores inteligentes) que almacenan y transmiten la información y con un sistema de control que determina las acciones necesarias en base a los datos recibidos. Estas redes pueden ser explicadas mediante una división de sus componentes en capas (ver figura 1.8):

- **Infraestructuras:** está compuesta por los sistemas de distribución de electricidad. Incluye subestaciones, líneas eléctricas, transformadores, sistemas de protección y sensores entre otros elementos.
- **Equipos:** aquí se cuentan los medidores inteligentes instalados en los puntos de consumo, los sistemas de almacenamiento y de generación distribuida (paneles solares, turbinas eólicas, etc.), las cargas eléctricas de la red y otros elementos como pueden ser los vehículos eléctricos.
- **Comunicación:** en esta capa se incluyen los sistemas de información necesarios para transmitir y archivar los datos recopilados por los equipos.
- **Sistemas de control:** estos son los elementos que reciben toda la información de la red, la analizan y determinan las acciones necesarias para mantener las condiciones requeridas y optimizar su funcionamiento. Son el cerebro de la red metafóricamente hablando. Pueden incluir sistemas para controlar las cargas eléctricas que puedan ser conectadas o desconectadas a voluntad, así como para controlar la generación distribuida y los elementos de protección de la red.

Puede reconocerse una última capa que va ganando peso y se espera cobre mayor importancia en el futuro, la de los **Servicios**. En el contexto actual, el consumo en los hogares se caracteriza por

una creciente electrificación y por la proliferación de aparatos y electrodomésticos inteligentes, conectados a internet y que permiten su control de forma remota. Existen incluso compañías eléctricas que han diversificado su oferta para incluir sistemas de control de casas inteligentes, acoplándolos en ocasiones con ofertas de movilidad eléctrica. Es el caso del gigante francés *Électricité De France*, que lanzó en 2016 la subsidiaria *Sowee* para abordar este nuevo mercado [11]. Estos cambios conllevan una progresiva sofisticación del perfil del consumidor, por la cual este tiene mayor control y puede influir en el sistema eléctrico con sus acciones, el llamado *prosumidor*. También comienzan a surgir nuevas posibilidades de colaboración entre consumidores, como la creación de mercados de energía locales, el comercio de energía entre particulares (*peer-to-peer*) o las plantas de generación virtuales [37]. Así, la retransmisión de los datos de consumo en tiempo real a los sistemas de control se revela como un elemento crucial en el funcionamiento de las nuevas redes.

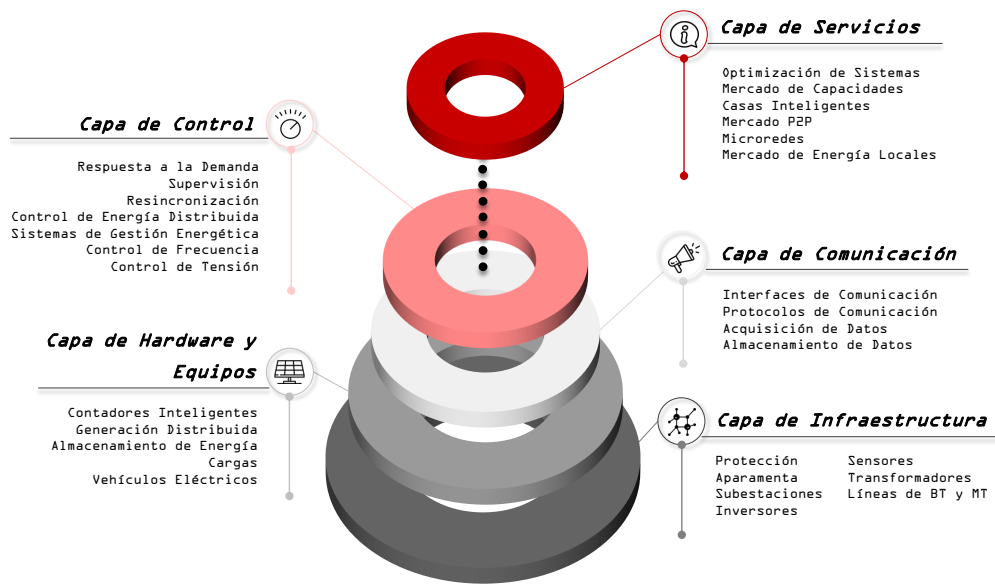


Figura 1.8 Representación de los elementos de una red de distribución eléctrica inteligente (fuente: elaboración propia).

El elemento definitorio común de las redes de distribución eléctricas inteligentes es, por lo tanto, el tratamiento digital de los datos: la gestión de la información es esencial en las redes eléctricas y la digitalización de los datos ha permitido ampliar la información disponible sobre el estado de las mismas, así como reducir los tiempos de procesamiento y respuesta de cara a su control [26].

Convencionalmente, los sistemas eléctricos se han caracterizado por su centralización con respecto a las grandes centrales de generación. Estas se situaban habitualmente cerca de reservas de combustibles fósiles (o en lugares a los que pudieran ser fácilmente transportados), salvo por las centrales hidráulicas, que condicionaron igualmente la configuración de las redes de transporte según su localización. La capacidad de la red está en estos casos generalmente sobredimensionada para asegurar el suministro durante los picos de demanda, lo cual genera unos costes importantes a las compañías productoras que acabaron revirtiendo en un incremento de las tarifas. Como respuesta a este problema, se comenzaron a desarrollar en la última década del siglo XX los sistemas de infraestructura de medida avanzada (AMI por sus siglas en inglés): se implementaron medidores inteligentes que permitían a los consumidores conocer y gestionar en mayor medida su consumo eléctrico [43].

Las posibilidades que se abren ante esta tecnología son múltiples, siendo una de ellas la gestión de la demanda eléctrica (*Demand-Side Management*). Contando con un sistema de control inteligente

que reciba toda la información de la red en tiempo real y tenga la capacidad de ajustar el consumo de ciertos aparatos conectados en las propias residencias, es posible optimizar enormemente la potencia necesaria. En momentos de bajo consumo y precio de la electricidad, se puede usar la energía de la red para cargar sistemas de baterías o cargar vehículos eléctricos. Al contrario, durante los picos de demanda sería rentable utilizar la energía de los sistemas de almacenamiento o incluso reducir las cargas que no sean indispensables. De esta forma se consigue una curva de consumo más estable, reduciendo los picos de demanda y por lo tanto la capacidad máxima necesaria en la red.

Los beneficios de este tipo de sistema son variados, aunque generalmente se consigue una red más flexible, accesible, fiable y económica [26]:

- Se sitúa al consumidor en el centro. Los flujos de potencia dejan de ser unidireccionales (de las centrales al consumidor) y pasan a ser bidireccionales gracias a la generación distribuida, los sistemas de almacenamiento de energía, etc. De igual manera, es posible la regulación de la demanda eléctrica mediante incentivos y tarifas especiales para los consumidores ya que estos tienen un mayor acceso a la información de su consumo (ver 1.4).
- Se aumenta el acceso a la red de los usuarios. Tanto las fuentes de energía renovable como otras formas de generación distribuida a nivel local permiten favorecer la descentralización y el auto-consumo.
- Se facilitan las tareas de operación y mantenimiento. El análisis de datos en tiempo real favorece el diagnóstico de la red y las actividades de control de emergencia. También se aumentan las capacidades de predicción, tanto de fallos como de consumo, lo que aumenta la seguridad y la calidad del suministro eléctrico.
- La optimización de los recursos permite reducir costes y se aumenta la competencia en los mercados liberalizados actuales.
- Se reducen las emisiones contaminantes de forma directa (por la reducción de pérdidas en la red) e indirecta (por la inclusión de formas de generación renovables que con anterioridad eran difícilmente integrables debido a su variabilidad, así como por la optimización de la demanda).
- La inclusión de nuevos actores como los vehículos eléctricos permite utilizarlos como sistemas adicionales de almacenamiento de energía.

En la actualidad hay distintos proyectos de mayor o menor envergadura, aunque muchos países apuestan por estos sistemas inteligentes para modernizar la distribución eléctrica [27]. En Portugal, la compañía EDP Distribuição ha implementado en la ciudad de Évora un proyecto de ciudad inteligente que incluye redes de distribución inteligentes [28]. Aun así, esta tecnología aún plantea importantes desafíos para su integración [26]:

- Es importante conseguir la implicación directa del consumidor para conseguir un mayor impacto favorable.
- Puede resultar en una complicación de las tarifas (tarifas variables) que reducirían la claridad de las mismas creando así desconfianza en los consumidores
- Surgen problemas en cuanto a la privacidad y el tratamiento de los datos personales de los clientes.
- Son necesarias importantes medidas de ciberseguridad debido al carácter estratégico de los sistemas.
- Las inversiones iniciales son grandes.
- A fin de aumentar la utilidad de las *smart grids*, la proliferación del internet de las cosas (aparatos eléctricos con capacidad con conexión a internet) se revela crucial para que las cargas

puedan ser controladas por el operador del sistema. Esto crea igualmente preocupaciones sobre la capacidad que tendrán las compañías operadoras para controlar aparatos privados.

1.4 Gestión de la Demanda Energética

La gestión de la demanda eléctrica, conocida como *demand-side management* o *demand-side response*, es el conjunto de medidas que pueden llevarse a cabo para optimizar los sistemas eléctricos desde el punto de vista de los consumidores.

La idea de utilizar la demanda como un grado de libertad adicional en la optimización de los sistemas eléctricos no es novedosa; sin embargo, el desarrollo de los sistemas de información y control y la proliferación del internet de las cosas en los últimos años la ha convertido en una realidad plausible. El aumento de la generación distribuida también supone una posibilidad de control suplementaria. Estas tecnologías son consideradas como un forma de reducir costes, puesto que, frente a la creciente electrificación, es menos costoso estabilizar la curva de consumo mediante la reducción y el control de la demanda que construir nuevas centrales eléctricas para aumentar la capacidad de generación del sistema.

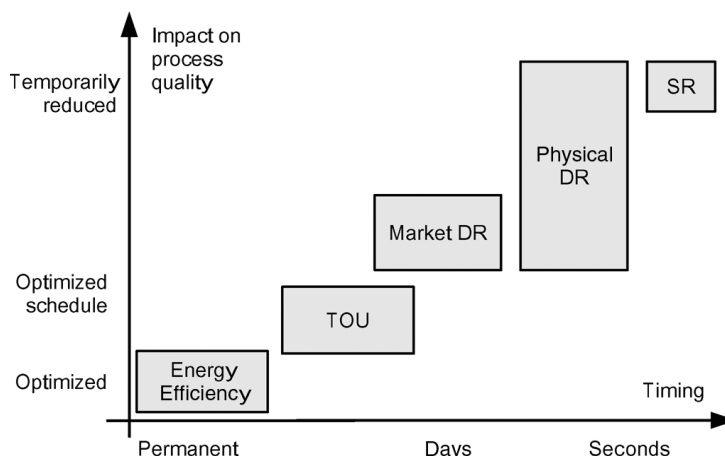


Figura 1.9 Categorías de DSM [39].

Atendiendo a su capacidad para impactar al sistema eléctrico y la duración temporal de las mismas, las principales tecnologías de DSM pueden ser categorizadas de la siguiente manera [39] (ver figura 1.9):

- **Eficiencia Energética (EE):** son las medidas destinadas a reducir la cantidad de energía necesaria para proporcionar servicios y productos. Pueden abarcar desde el diseño de electrodomésticos o sistemas de ventilación con un menor consumo eléctrico hasta el aislamiento térmico de edificios o la optimización de procesos industriales. Estas medidas son de carácter permanente, por lo que son aquellas que producen un mayor impacto en el sistema al reducir el consumo en la raíz del mismo (ver figura 1.10).
- **Time-Of-Use (TOU):** se refiere al establecimiento de tarifas eléctricas en las que la energía tiene un precio diferente según la hora del día en la que se consuma (por ejemplo un precio superior para las horas de mayor demanda e inferior durante la noche). Esto supone una adaptación de los contratos de los clientes y requiere un compromiso por su parte para que sus efectos sean considerables.

- **Gestión dinámica de la demanda:** supone crear incentivos o penalizaciones para que los clientes modifiquen sus hábitos de consumo. Aquellas basadas en incentivos son variadas: pueden establecerse contratos por los cuales los clientes que permitan al operador de red controlar sus cargas tengan una compensación económica, los operadores pueden mandar señales de emergencia de consumo excesivo para que los consumidores voluntariamente reduzcan su consumo, se pueden crear sistemas de subasta mediante los que los clientes puedan pujar para reducir su consumo durante ciertos periodos a precios más competitivos o mercados de capacidad en los que los consumidores se comprometen a conectar sus generadores a la red cuando sea necesario, entre otros ejemplos. Otras medidas serían las basadas en el tiempo: aumentar precios de la energía puntualmente durante los picos de consumo o facturar al cliente según el precio de mercado de la electricidad en tiempo real son algunos de estos ejemplos.

Este tipo de medida se pueden dividir igualmente entre aquellas basadas en el mercado, como el ajuste de precios según el consumo, y las que están determinadas por las necesidades físicas de la red, como las señales de emergencia.

- **Reservas Rotativas:** son medidas para controlar la frecuencia y el voltaje de la red en cortos periodos de tiempo. Esta función se atribuye históricamente a los generadores, pero la reducción de las cargas puede tener el mismo efecto en el sistema al ser reservas rotativas "negativas". Un ejemplo sería que los aparatos conectados redujesen su potencia cuando la frecuencia de la red disminuya.

Entre todas estas medidas y tecnologías, las de eficiencia energética son las que tienen un impacto más positivo en la red debido a su carácter permanente. Es en el ámbito del resto de medidas que las redes de distribución inteligentes ganan relevancia, ya que en lugar de reducir el consumo, lo desplazan en el tiempo para adaptar mejor la demanda a la producción. Esto puede apreciarse claramente en la figura 1.10. Las medidas de DSM dinámicas (representadas como DR en la imagen) pueden incluso tener un efecto rebote que produzca picos de consumo aún mayores debido a la necesidad de los aparatos de utilizar una potencia mayor para recuperar sus funcionalidades óptimas tras ser desconectados (un termo de agua eléctrico, por ejemplo).

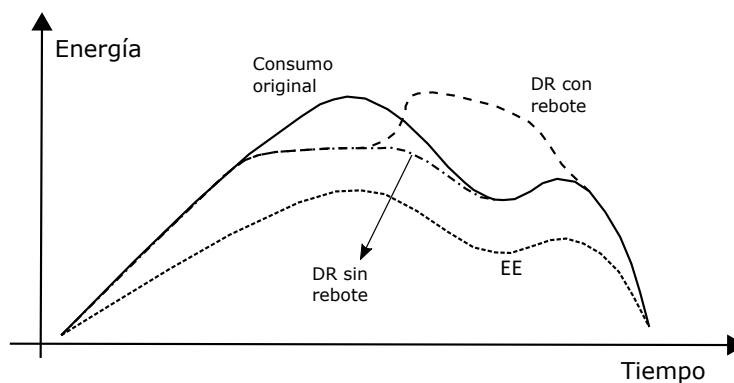


Figura 1.10 Impacto de diferentes tecnologías en el consumo eléctrico diario [39].

1.5 Presentación del Problema

La finalidad de este proyecto es crear un programa informático en lenguaje Python que permita estudiar y optimizar el comportamiento de una red de distribución inteligente de media tensión.

Como se ha comentado anteriormente en 1.3, las redes de distribución inteligentes están experimentando un auge gracias a sus importantes beneficios. Con el aumento de las energías renovables

(generalmente conectadas directamente a las redes de distribución a nivel local) se plantea igualmente un cambio en el comportamiento de las redes que es interesante analizar. Por otro lado, el lenguaje de programación Python es un lenguaje muy versátil, de código abierto, con una sintaxis sencilla e intuitiva y permite modelizar sistemas grandes y complejos gracias a sus características de lenguaje orientado a objetos. Concretamente, la modelización se hará con Pyomo (Python Optimization Modeling Objects), una colección de paquetes de Python utilizado para la formulación de problemas de optimización matemática [30].

Se utilizará un red de referencia creada por el Grupo de Trabajo C6.04.02 de CIGRE. El objetivo de este grupo de trabajo fue establecer un red de referencia común para el estudio de la integración de sistemas de generación distribuida [40]. La red de distribución de media tensión deriva de una red rural alemana que alimenta un pequeño núcleo urbano y las áreas rurales de alrededor. Está conformada por un total de 14 nodos repartidos en dos subredes con un voltaje de 20 kV que son alimentadas por sendos transformadores de 110/20 kV. La red cuenta con fuentes de generación distribuida (tanto renovable como convencional), sistemas de almacenamiento de energía y diversas cargas residenciales e industriales.

1.5.1 Alcance y objetivos del proyecto

En el ámbito de este proyecto debe crearse un modelo de la red y de un sistema de gestión de la energía y analizar los resultados derivados de dicho modelo.

La primera parte consiste en modelizar todos los atributos de la red en Pyomo para representarla de la forma más veraz posible. Se creará igualmente un problema de optimización matemática (siendo en este caso de tipo mixto y no lineal: MINLP; ver 2.2) que permita minimizar los costes asociados a la red en función de una serie de restricciones. Estas incluyen, entre otras, el control del voltaje en los nodos y de la potencia aparente las líneas, el balance de los flujos de potencia y el mantenimiento de reservas destinadas al control de frecuencia. Este tipo de problema se llama convencionalmente de Asignación de Unidades (*Unit Commitment*) si se utilizan escenarios estocásticos basados en predicciones para definir el orden en el que las centrales se pondrán en funcionamiento y de Despacho Económico (*Economic Dispatch*) si los datos son deterministas y se pretende conocer la potencia que cada unidad deberá producir para suplir la demanda. Se puede encontrar una explicación más detallada de los dos tipos de problema en el apartado 3.2.1.

Posteriormente se analizarán los datos extraídos del modelo. Se estudiarán tanto los resultados de un modelo determinista como de un modelo estocástico. Para este último se crearán diferentes escenarios de precios y de generación renovable a partir de datos históricos. Se compararan ambos modelos y se analizarán elementos como el uso de baterías, la desconexión de las cargas no esenciales o la influencia de restricciones auxiliares.

En los siguientes apartados se explicará la metodología seguida durante el proyecto, incluida una descripción de la herramienta Pyomo, y se presentará en mayor detalle la red estudiada, así como el problema de optimización a resolver y los datos deterministas y estocásticos utilizados. Posteriormente, se incluye un análisis de los resultados y una conclusión del proyecto con posibles ampliaciones a considerar. En los anexos puede encontrarse el código completo creado en Python.

2 Metodología

En este apartado se ofrece una descripción del proceso seguido y las herramientas utilizadas durante la realización del proyecto, especialmente Pyomo, así como una explicación de los problemas de optimización matemática.

2.1 Proceso de Investigación y Análisis

El proceso seguido para realizar este proyecto comienza con la documentación y la investigación sobre el sujeto en cuestión: las redes inteligentes, su modelización y la integración de renovables en las mismas. Tras este paso, se ha procedido a modelar la red y el problema de optimización, el cual ha sido resuelto con los datos obtenidos. Por último se han analizado los resultados para verificar el correcto funcionamiento del modelo desarrollado.

2.1.1 Documentación

La primera etapa del proyecto ha estado claramente marcada por la investigación y la búsqueda de documentos bibliográficos sobre las redes de distribución eléctrica, las *Smart Grids*, los problemas de asignación de unidades (UC) y de despacho económico (ED) y la modelización y resolución de problemas de optimización matemática. Igualmente, se ha buscado información relativa al cambio climático, al contexto energético mundial, a la proliferación de energías renovables, a los desafíos asociados a ellas y los vectores de innovación que comienzan a surgir en el mercado.

Para comprender el concepto de red inteligente y la tecnología asociada se ha recurrido a artículos mencionados anteriormente como [26] y [27]. Del mismo modo, se han estudiado artículos que hacen uso de la red de distribución de CIGRE para diversos estudios, especialmente [45], aunque también [23] donde se explican los distintos niveles de control presentes en una *smart grid*. Ha sido también de gran ayuda el artículo [35] donde se sientan las bases del problema de optimización matemática a resolver, así como el libro [47] donde se pueden encontrar ejemplos de problemas similares modelizados con el software GAMS. Igualmente ha sido necesario el estudio de la documentación relativa a Pyomo para la programación del código en Python [30] [10].

2.1.2 Modelización

La parte del proyecto dedicada a la modelización ha sido la más importante y laboriosa. Esta puede dividirse en dos bloques: la modelización de la red de referencia y la creación y resolución del problema de optimización. Esta última parte representa la creación de un sistema de gestión de energía en la red (*Energy Management System*).

Inicialmente se creó un modelo reducido de la red y del problema de optimización. Se utilizaron datos deterministas e instrumentales para una única hora y un número reducido de nodos y se limitó

el número de restricciones del problema. Progresivamente se fueron añadiendo diferentes cargas y generadores para estudiar si su comportamiento era el esperado hasta terminar con un modelo determinista completo en el que ya se tenían en cuenta la totalidad de las restricciones, incluidas las auxiliares. Por último, se creó un modelo estocástico partiendo del determinista. En él se incluyen varios escenarios, cada uno con una probabilidad asociada, según el precio del mercado eléctrico y la generación fotovoltaica y eólica.

2.1.3 Obtención de datos

La obtención de los datos necesarios ha sido un proceso que se ha llevado en paralelo a la creación del modelo y al análisis de los resultados preliminares, puesto que ha sido necesario modificarlos continuamente para comprobar la validez de los mismos.

Los datos relativos a la red se han extraído del artículo donde se detallan las características de la red de referencia de media tensión en Europa [40]. La información sobre el precio del mercado eléctrico para todo un año puede encontrarse en la página web del operador del mercado ibérico [9] y los datos de viento y radiación solar se han tomado de bases de datos abiertas [6].

Para la creación de los escenarios estocásticos relativos al precio y la generación renovable se han utilizado los datos de todos los días laborales de un año completo. Se ha recurrido a la agrupación de los mismos en cuatro grupos diferenciados según sus similitudes en valor y forma de la curva mediante un programa basado en lenguaje R. La probabilidad de cada escenario se ha determinado según el número de días perteneciente a cada grupo.

2.1.4 Análisis de resultados

En esta sección se estudiarán los datos producidos mediante la resolución del problema de optimización realizado anteriormente. Se hará una comparación entre los resultados de los modelos determinista y estocástico durante un periodo de 24 horas.

Los indicadores a analizar son los siguientes:

- Coste total de la red
- Voltaje en los nodos
- Potencia importada de la red y potencia producida por los generadores y baterías
- Estado de los generadores en los distintos momentos del día (encendido o apagado)
- Consumo de las cargas no esenciales

Se estudiará el efecto de diversos elementos, como la inclusión de restricciones auxiliares (mantener ciertos niveles de reservas de energía o conseguir un voltaje determinado en el Punto de Acoplamiento Común o PCC), la influencia de la desconexión de cargas no esenciales (la industria en este caso) o un aumento considerable de la generación renovable, siempre verificando que se respeten las restricciones esenciales de voltaje y balances de potencia en los nodos. Por otro lado, se presentará el resultado del modelo estocástico que tiene en cuenta la incertidumbres asociadas a la generación renovable y al precio del mercado eléctrico.

2.2 Problemas de Optimización Matemática

Una gran parte del proyecto consiste en la definición y resolución de un problema de optimización matemática. La optimización matemática es el conjunto de técnicas que permiten encontrar el valor óptimo de una función de acuerdo con un criterio establecido y de forma que los valores cumplan con unas restricciones definidas. Para un problema de minimización:

Dada una función $f : A \rightarrow \mathbb{R}$, buscar un elemento x_0 en A tal que $f(x_0) \leq f(x)$ para todo x en A .

Un problema de optimización consta por lo tanto de una función a optimizar (función objetivo) formada por variables (son los valores que deben ser optimizados) y por parámetros (son predefinidos y se mantienen constantes), y por una serie de ecuaciones e inecuaciones que también combinan variables y parámetros y que componen las restricciones.

Según el grado de las ecuaciones o inecuaciones que componen las restricciones y la función objetivo, los problemas pueden ser lineales, cuadráticos o no lineales. Si las variables de la función objetivo solo pueden tomar valores enteros, la programación será de tipo entera, y si se incluyen tanto variables enteras como continuas, el tipo de problema será de programación mixta. Se puede encontrar una descripción de los principales problemas de optimización en la tabla 2.1.

Tabla 2.1 Problemas de optimización.

Tipo de Problema	Descripción Matemática
LP (Programación Lineal)	$\min c^T x$ $s.a. Ax = b$ $x \geq 0$ $x \in \mathbb{R}^n, c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$
QP (Programación Cuadrática)	$\min c^T x + \frac{1}{2} x^T Q x$ $s.a. Ax = b$ $x \geq 0$ $x \in \mathbb{R}^n, c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}$ $Q \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^m$
NLP (Programación No Lineal)	$\min f(x)$ $s.a. g(x) = 0$ $h(x) \leq 0$ $x^L \leq x \leq x^U$ $x \in \mathbb{R}^n$ $f : \mathbb{R}^n \rightarrow \mathbb{R}$ $g, h : \mathbb{R}^n \rightarrow \mathbb{R}^m$
MIP (Programación lineal Entera Mixta)	$\min c^T x + d^T y$ $s.a. Ax = b$ $x, y \geq 0$ $x \in \mathbb{Z}^n, y \in \mathbb{R}^l, c \in \mathbb{R}^n, d \in \mathbb{R}^l$ $A \in \mathbb{R}^{m \times n}, Q \in \mathbb{R}^{m \times l}, b \in \mathbb{R}^m$
MINLP (Programación No Lineal Entera Mixta)	$\min f(x, y)$ $s.a. h(x, y) = 0$ $g(x, y) \geq 0$ $x^L \leq x \leq x^U$ $x \in \mathbb{R}^n, y \in \mathbb{Z}^m$ $f : \mathbb{R}^n \rightarrow \mathbb{R}$ $g, h : \mathbb{R}^n \rightarrow \mathbb{R}^m$

Tal y como se explica en la sección 3.2.1, el problema matemático correspondiente al presente proyecto es de tipo MINLP.

Además de esta clasificación, existen otros criterios para clasificar los problemas. A continuación se presentan algunos de los más comunes:

- Programación convexa: son los problemas en los que el conjunto de restricciones es convexo.
- Programación estocástica: en estos problemas al menos una de las variables incluidas en la función objetivo o las restricciones son aleatorias.
- Programación heurística: se llega a una solución mediante la realización de suposiciones, por lo que estas son aproximadas. Suele aplicarse en problemas demasiado extensos o complejos y los métodos de resolución suelen ser iterativos.
- Programación dinámica: consiste en dividir el problema principal en subproblemas con el fin de acelerar la resolución.

2.2.1 Algoritmos de resolución de problemas MINLP

Existen diversos algoritmos para resolver problemas mixtos y no lineales, tanto exactos como heurísticos. La naturaleza del problema y sus dimensiones determinarán qué método es más útil en cada caso. Los principales algoritmos están basados en la relajación de las restricciones para hallar la solución del problema con variables enteras relajadas o restricciones lineales.

Uno de los métodos más extendidos es el de Ramificación y Acotamiento (*Branch-and-Bound*). Consiste en crear bifurcaciones de los valores de las variables binarias (algoritmo de árbol) para dividir el problema en subproblemas más abordables y posteriormente ir descartando aquellas ramas que tengan un coste mayor que el de la mejor solución encontrada hasta el momento. En cada nodo de cálculo, el programa tan solo tiene que resolver un problema NLP (o linear si las ecuaciones han sido linealizadas), lo que simplifica el tiempo de computación, y evaluar las ramificaciones que pueden surgir del nodo para descartar las no factibles o que resultan en una solución peor. Conforme se avanza en la resolución del problema el algoritmo reduce el número de ramas activas.

Otro algoritmo es la Aproximación Exterior (*Outer-Approximatio*). Este es un método iterativo en el que inicialmente se establece un valor de las variables binarias, se resuelve el problema continuo NLP y se utilizan los valores obtenidos como límite superior de la solución (si el problema NLP con las variables binarias fijas es factible, su solución corresponde a una solución del MINLP y por lo tanto puede usarse como límite del mismo). Posteriormente se crea un problema maestro de tipo MIP a través de la solución obtenida en el NLP y de la relajación del problema principal, se resuelve y se usa la solución como límite inferior. Se comparan ambos límites con respecto a una tolerancia predefinida y, si no se ha hallado la solución óptima, se repite el proceso: se resuelve el NLP con los valores de las variables binarias obtenidos en la solución del problema maestro para hallar un nuevo límite superior que se incorpora al problema maestro.

El algoritmo LP/NLP basado en Ramificación y Acotamiento aplica el principio del método OA anterior a una búsqueda en árbol. En este caso, el problema maestro se define de forma dinámica para reducir el número de ramas activas progresivamente. Inicialmente se resuelven subproblemas LP en los nodos para obtener límites inferiores y luego, si tienen solución factible, se resuelve el NLP, cuyas soluciones se añaden al problema maestro como límites superiores para acotar las soluciones de los problemas LP en los nodos abiertos [41]. Este método evita la sucesiva resolución del problema maestro MIP, pero puede aumentar el número de NLP a resolver.

2.3 Pyomo

Tanto para la modelización de la red como para la definición y resolución del problema matemático, Pyomo es la herramienta principal que se ha utilizado a lo largo del proyecto [29] [30]. Es una colección de paquetes de Python utilizado para la modelización de problemas de optimización (Python Optimization Modeling Objects) que permite crear grandes problemas con una sintaxis sencilla, incluir todos los paquetes adicionales de Python y también importar y exportar datos en los formatos habituales.

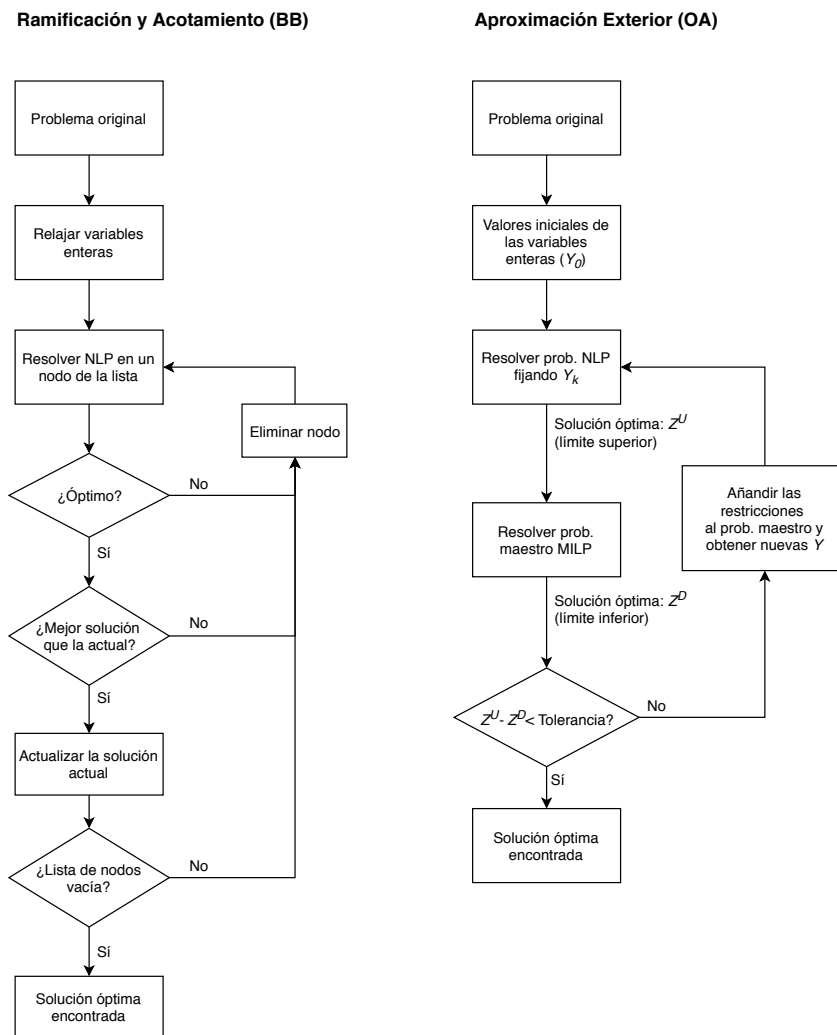


Figura 2.1 Algoritmos para resolver problemas de optimización MINLP (fuente: elaboración propia).

Además de Pyomo, se han utilizado otros programas para diversas acciones, especialmente Microsoft Excel para la introducción de datos a Pyomo y el entorno y lenguaje de programación R [42] para agrupar ciertos datos en escenarios estocásticos. Igualmente se ha utilizado la biblioteca Matplotlib para generar los gráficos directamente con el programa en Python [31]

2.3.1 Elementos de Pyomo

Python es un lenguaje orientado a objetos. Esto quiere decir que usa objetos para manipular datos según su funcionalidad asociada. Los objetos se caracterizan por tres atributos principales:

- Estado: conjunto de datos asignados al objeto.
- Comportamiento: métodos y funcionalidades con las que puede operar el objeto.
- Identidad: identificador del objeto que lo diferencia de otros.

En el caso de Pyomo, se utilizan diversos objetos que son los índices (sets), los parámetros, las variables, las restricciones (constraints) y los objetivos [10].

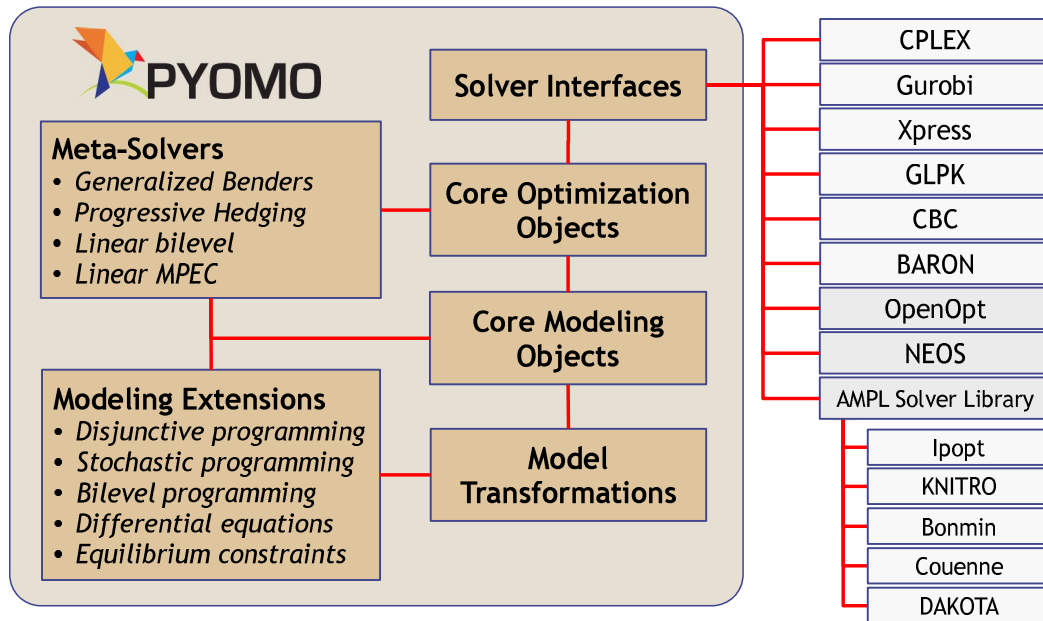


Figura 2.2 Resumen de Pyomo.

- Parámetros: representan los datos que tienen que ser proporcionados al modelo.
- Variables: son los objetos cuyo valor es desconocido inicialmente y que finalmente representan la solución del problema.
- Sets: índices con los que se indexan las variables y parámetros.
- Constraints: restricciones que tienen que cumplir las variables. Se definen mediante ecuaciones formadas por los parámetros y las variables.
- Objetivos: expresiones o ecuaciones definidas mediante los parámetros y las variables y que tienen que ser minimizadas o maximizadas en la medida de lo posible.

Por lo tanto, los parámetros y las variables son indexados por los sets y sirven para representar los elementos de la red que se quiere modelizar. Las variables serán optimizadas mediante un solver para que cumplan todas las restricciones especificadas mientras que se intenta minimizar o maximizar el valor final de los objetivos en la medida de lo posible.

$$\begin{aligned}
 & \text{Minimizar } \sum_{j=1}^n C_{ij} X_j \\
 & \text{s.a. } \sum_{j=1}^n A_{ij} X_j \geq B_i \quad \forall i = 1 \dots m \\
 & X_j \geq 0 \quad \forall j = 1 \dots n
 \end{aligned}$$

En este problema matemático de programación lineal, los parámetros A , B y C están indexados por los sets i y j . La variable X tiene que cumplir las restricciones de forma que se minimice la función objetivo.

Pyomo tiene la capacidad de definir dos tipos de modelo diferentes: abstractos y concretos. Los modelos abstractos son aquellos en los que se utilizan símbolos para representar los parámetros, cuyos valores no se especifican hasta el momento de calcular la solución. De esta forma se permite inicializar los valores mediante diversos métodos y reutilizar los modelos con mayor facilidad. En los

modelos concretos los valores de los parámetros son definidos directamente durante la construcción del modelo. Los modelos abstractos pueden asociarse con una plantilla vacía que se rellena con valores posteriormente mientras que los concretos sería una plantilla con los datos ya incluidos.

Para este proyecto se ha optado por un modelo abstracto, de forma que los datos serán importados desde un archivo Excel, facilitando así la modificación de los mismos.

2.3.2 Solvers

Para resolver los problemas de optimización, Pyomo utiliza softwares matemáticos ajenos. A estos programas se les denomina solvers y su función es recibir la descripción del problema (el modelo) y realizar los cálculos necesarios para obtener la solución óptima. El tiempo de cálculo dependerá del tamaño, de la complejidad del problema y de la capacidad de cada solver, por lo que una evaluación del rendimiento de Pyomo en comparación con otros programas no es factible, las diferencias radicarán principalmente en el solver utilizado por cada uno.

Pyomo soporta una gran variedad de solvers y cuenta con interfaces específicas para algunos de ellos (ver figura 2.2) [10]. Por otro lado, también permite trabajar con cualquier solver que tenga una interfaz AMPL, así como resolver problemas matemáticos a través del servidor NEOS (*Network-Enabled Optimization System*), una aplicación basada en internet, de libre acceso y que cuenta con una gran variedad de solvers matemáticos, tanto comerciales como de código abierto [25]. A través de estos programas, o mediante la combinación de ellos, pueden resolverse prácticamente todo tipo de problemas de optimización.

Sin embargo, la oferta de solvers de código abierto capaces de resolver problemas de optimización MINLP es limitada [34] [53]. Linealizar las ecuaciones es una opción plausible, pero a fin de conseguir la mayor precisión en los resultados se evitará esta opción a pesar de aumentar el tiempo de cálculo. Para aquellos problemas con restricciones no convexas, pueden aplicarse algoritmos heurísticos para encontrar soluciones aproximadas. Algunos de los principales solvers que pueden ser utilizados para resolver problemas MINLP son los siguientes:

- **BARON** (*Branch-And-Reduced Optimization Navigator*): es un programa orientado a resolver problemas con restricciones no convexas de una forma exacta. Es uno de los solvers más avanzados del mercado pero no está disponible en código abierto. Utiliza un método de Ramificación y Acotación optimizado [49].
- **BONMIN** (*Basic Open-Source Nonlinear Mixed INteger programmin*): como su nombre indica, este solver es de código abierto desarrollado dentro del proyecto COIN-OR [1] y permite resolver el tipo de problemas tratados en este proyecto. Permite encontrar soluciones exactas para problemas convexas mediante diferentes algoritmos (BB, OA, LP/NLP o Híbridos) y aplica un método heurístico (BB con posibilidad de generar diferentes puntos iniciales de cálculo aleatoriamente) para encontrar óptimos locales si el problema es no convexo [24].
- **COUENNE** (*Convex Over and Under ENvelopes for Nonlinear Estimation*): este programa también es de código abierto y ha sido igualmente desarrollado dentro de COIN-OR. Es utilizado para calcular soluciones exactas de problemas no convexas mediante linealización y algoritmos de árbol (BB) [14].
- **DICOPT** (*DIcrete and Continuous OPTimizer*): en este caso, el solver no es de código abierto y solo puede aplicarse a problemas en los que los términos no lineales sean estrictamente continuos. Utiliza algoritmos de aproximación exterior [4].

El solver escogido para este proyecto es BONMIN debido a su carácter de código abierto y a su capacidad para resolver problemas MINLP tanto convexas como no convexas. Este será utilizado a través del servidor NEOS para facilitar su uso y evitar los problemas asociados a la instalación, para la cual se requieren programas adicionales y la compilación mediante Cygwin para sistemas operativos Windows.

2.3.3 Extensiones: PySP para modelización estocástica

Pyomo incluye una serie de extensiones para resolver problemas de mayor complejidad o de naturaleza distinta. Entre estas se incluye una extensión dedicada a la optimización dinámica, una para crear redes de problemas y otra para resolver problemas mediante optimización multinivel. La extensión que resulta más relevante para el proyecto es PySP, que está orientada a la modelización y resolución de problemas estocásticos [53].

Para utilizar este programa, se crea un modelo determinista que va a servir como referencia para el estocástico. Posteriormente, se generan los escenarios, usualmente en forma de árbol de nodos en el que se combinan todas las variables estocásticas; cada nodo tendrá una probabilidad asignada respecto al nivel superior. En el nivel más bajo del árbol aparecerán, por lo tanto, todos los escenarios posibles. Por último, se introducen los datos mediante un fichero de texto *.dat*. En el árbol de escenarios cada nivel representa una etapa de decisión. En ella se calculará un valor para las variables asignadas a la misma a partir de los valores de las que ya han sido calculadas en los niveles inferiores. En la sección 3.5 se hace una explicación más detallada de la estructura del problema estocástico.

Esta extensión incluye dos métodos para resolver el problema: resolver directamente la conocida como "forma extensa" del problema, indexando las variables a los escenarios estocásticos e incluyendo probabilidades en la función objetivo (ver 3.2.2), o aplicar el algoritmo *Progressive Hedging*, que lo resuelve mediante una estrategia de descomposición. Según Watson y Woodruff en [52] y [53], este último método es más útil para problemas extensos y puede ser utilizado con efectividad como método heurístico si el problema incluye variables binarias, como es el caso del presente proyecto.

El algoritmo *Progressive Hedging* es un método iterativo en el que se calculan valores medios de las variables para cada nivel según los valores que tomen las variables que se encuentran en los niveles inferiores. Para cada iteración, se compara la solución calculada con la anterior y se detiene el algoritmo cuando la diferencia sea menor que una tolerancia previamente definida [53].

Al contrario que en con el modelo determinista, esta extensión de Pyomo no permite utilizar la interfaz remota con el servidor NEOS, por lo que en este proyecto no se ha tenido acceso a un solver de código abierto capaz de resolver el problema MINLP en el caso estocástico. Para este caso, se ha optado por relajar las variables binarias y resolver el problema NLP resultante con el solver Ipopt y aproximar posteriormente el valor de las variables binarias a 0 ó 1.

3 Caso de Estudio

En este apartado va a presentarse en detalle el objeto de estudio del proyecto. Se explicarán las características de la red escogida, se presentará el problema matemático y su formulación completa y se describirá el funcionamiento y la estructura del programa desarrollado en Python.

3.1 Red de Distribución de Referencia

La red objeto de estudio de este proyecto es una red de referencia desarrollada por un grupo de trabajo del CIGRE (Consejo Internacional de Grandes Redes Eléctricas) [40]. Está enmarcada dentro de un proyecto cuyo objetivo era crear un sistema de referencia internacional para facilitar la realización de tests, análisis y validación de los nuevos métodos y técnicas que surgen para la integración de la generación distribuida en las redes eléctricas, así como para el estudio de las tecnologías de *smart grids*. La necesidad de crear dicho sistema de referencia surge de la proliferación de las unidades de distribución descentralizadas, las cuales suelen ser conectadas a las redes de media y baja tensión, causando significativos cambios estructurales en los actuales sistemas de potencia.

En el marco de dicho proyecto, se desarrolló una metodología para facilitar la modelización de redes y se crearon redes de referencia para alta, media y baja tensión (tanto para América del Norte donde las redes funcionan a 60 Hz como para Europa y redes de 50 Hz). Estas redes pueden ser utilizadas para diversos fines: estudio del impacto de la generación distribuida en el flujo de potencia de líneas de distribución o en las curvas de voltaje, estudio de la estabilidad de la tensión, estudio de protecciones de las redes o estudio de sistemas de gestión de energía (como es el caso del presente estudio) entre otros. En este proyecto se modelizará la red de media tensión de referencia para Europa. Esta red ya ha sido utilizada en estudios como [23].

La red de referencia modelizada tiene una potencia nominal de 20 kV y funciona a una frecuencia de 50 Hz. Es una representación de una red rural de media tensión alemana que alimenta una pequeña ciudad y sus alrededores. Cuenta con dos subredes alimentadas por sendos transformadores de 20 MVA y 110/20 kV de una subestación cercana [45]. La red modelizada consta de catorce nodos divididos entre las dos subredes. Ambas ramas se conectan por un interruptor que puede utilizarse para simular las posibles configuración de la red o estudiar el desempeño de diferentes elementos de conexión, como es el caso de [23], donde se evaluaron los beneficios de implementar convertidores AC-DC para regular la transmisión de potencia. En este proyecto se considerará abierto. Hay un único nodo de transición en el que ninguna carga o generador está conectado, los otros trece tienen algún elemento asociado. Puede verse una representación de la red en la figura 3.1.

3.1.1 Líneas de la red

Los conductores que conectan los nodos están caracterizados por los parámetros presentados en la tabla 3.1. Las líneas que conectan los nodos de la subred 1 (nodos 1 a 11) son subterráneos y tienen

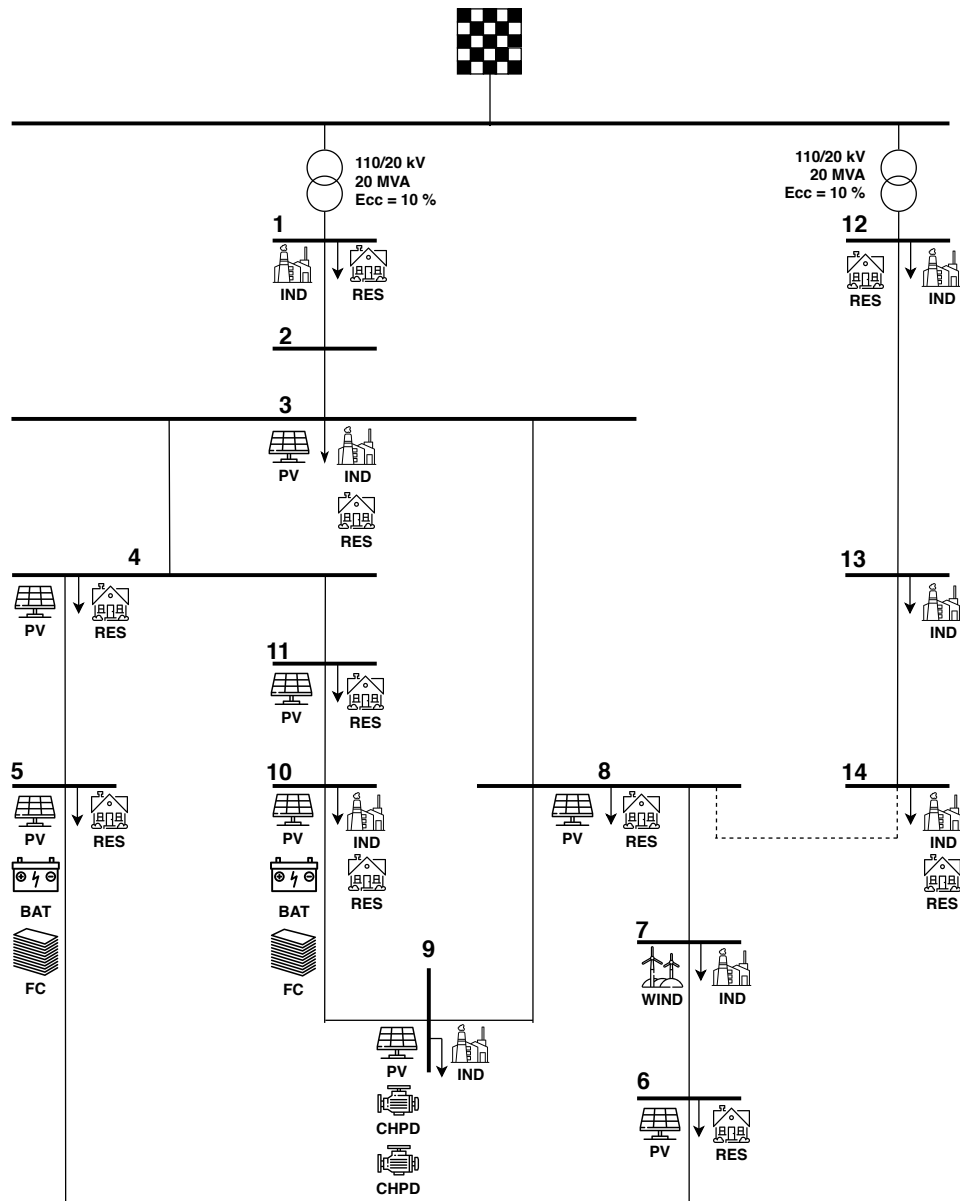


Figura 3.1 Red de referencia [40].

una sección de 120mm^2 mientras que los de la subred 2 son aéreos y tienen una sección de 63mm^2 .

3.1.2 Cargas de la red

Las cargas presentes en la red son de dos tipos: residenciales e industriales. La carga en cada nodo es calculada en [40] a partir del número de consumidores y su potencia máxima de consumo. Para ello, se multiplica la suma de las cargas aparentes individuales por un coeficiente de coincidencia $CF = 0,6(1 + 1/N_c)$, donde N_c es el número de consumidores en ese nodo. La tabla 3.2 muestra los valores picos de potencia aparente (S_{max}), el factor de potencia ($\cos \phi$) para cada nodo y la potencia activa máxima resultante (P_{max}). Puede apreciarse que los nodos 1 y 12 muestran unas cargas mucho mayores que los dados a otros nodos. Esto se debe a que representan las cargas de los otros feeders alimentados por los transformadores.

La diferencia principal entre ellas radica en que las industrias van a ser consideradas en este proyecto como cargas no esenciales y que, por lo tanto, pueden estar sujetas a desconexión para

Tabla 3.1 Parámetros de las líneas de la red [45].

Nodo i	Nodo j	L [km]	R [Ω]	X [Ω]	C [nF]	X _{sh} [Ω]
1	2	2,82	1,63	1,03	448,04	7104,47
2	3	4,42	0,72	0,50	29207,36	108,98
3	4	0,61	0,16	0,07	3952,80	805,28
3	8	1,3	0,22	0,15	8548,80	372,34
4	5	0,56	0,20	0,07	2553,60	1246,51
4	11	0,49	0,21	0,07	2430,40	1309,70
5	6	1,54	0,52	0,19	8451,52	376,63
6	7	0,24	0,06	0,03	902,40	3527,37
7	8	1,67	0,49	0,21	9352,00	340,37
8	9	0,32	0,11	0,04	1397,76	2277,29
9	10	0,77	0,31	0,10	3720,64	855,52
10	11	0,33	0,12	0,04	1504,80	2115,30
12	13	4,89	1,65	1,75	796,48	3996,44
13	14	2,99	0,60	0,36	14304,16	222,53

conseguir mantener las condiciones de optimalidad en el funcionamiento de la red.

En el ámbito de este proyecto, estas cargas van a ser consideradas deterministas, siguiendo siempre el mismo perfil. Estos pueden encontrarse en la figura 3.2 según los valores típicos presentados en [45].

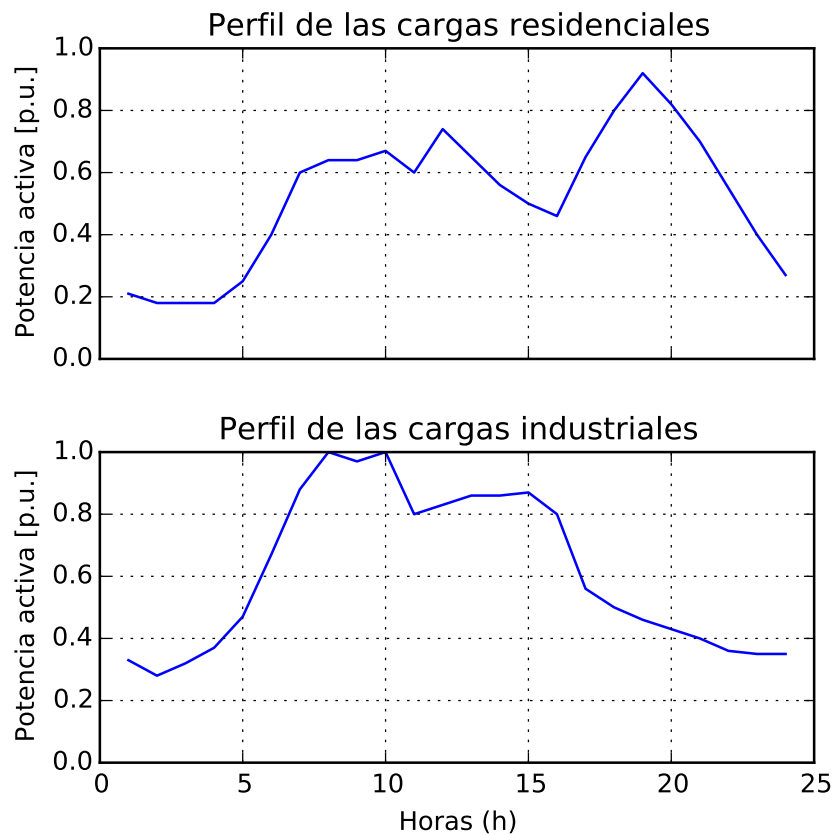
3.1.3 Generadores de la red

En la red de referencia se pueden encontrar diversos sistemas de generación distribuida y sistemas de almacenamiento [40]:

- Aerogenerador: la energía cinética del viento acciona una hélice y esta energía se transforma en corriente alterna mediante un alternador
- Paneles fotovoltaicos (PV): las células fotovoltaicas producen electricidad en corriente continua a partir de la luz solar mediante el efecto fotoeléctrico y un inversor la transforma en corriente alterna para ser inyectada a la red.
- Sistema de cogeneración alimentado con diesel (CHPD): se acciona un turbina mediante un motor de combustión interna para producir electricidad y se aprovecha el calor disipado.
- Sistema de cogeneración basado en una pila de combustible (CHPFC): estos sistemas son alimentados por un combustible (hidrógeno, por ejemplo) que en reacción con un oxidante producen una corriente eléctrica continua, precisan por lo tanto de un inversor para inyectar la corriente a la red tras transformarla en alterna. El calor disipado por la pila de combustible es aprovechado

Tabla 3.2 Parámetros de las cargas de la red [45].

(a) Cargas residenciales.				(b) Cargas industriales.			
Nodo	S_{\max} [MVA]	$\cos \varphi$	P_{\max} [MW]	Nodo	S_{\max} [MVA]	$\cos \varphi$	P_{\max} [MW]
1	15,3	0,98	14,994	1	5,1	0,95	4,845
3	0,285	0,97	0,276	3	0,265	0,85	0,225
4	0,445	0,97	0,431	7	0,09	0,85	0,077
5	0,75	0,97	0,728	9	0,675	0,85	0,574
6	0,565	0,97	0,548	10	0,08	0,85	0,068
8	0,605	0,97	0,587	12	5,28	0,95	5,016
10	0,49	0,97	0,475	13	0,04	0,85	0,034
11	0,34	0,97	0,33	14	0,39	0,85	0,332
12	15,3	0,98	14,994	12	15,3	0,98	14,994
14	0,215	0,97	0,209	14	0,215	0,97	0,209

**Figura 3.2** Perfiles de las cargas de la red [45].

- Pilas de combustible (FC): siguen el mismo funcionamiento que el sistema anterior pero, en este caso, son de una dimensión más reducida para adaptarlas a un uso residencial y no se aprovecha su calor residual. Precisan igualmente de un inversor para transformar la corriente en alterna.
- Sistemas de baterías (BS): son baterías de carácter residencial que almacenan y producen energía según lo más conveniente para la red. Es necesario un inversor para inyectar la corriente a la red.

Pueden encontrarse los parámetros de estos sistemas en la tabla 3.3.

Tabla 3.3 Parámetros de los generadores de la red [45].

Nodo	Generador	P_{\max} [kW]
1	Aerogenerador	1500
2	Fotovoltaico	20
4	Fotovoltaico	20
5	Fotovoltaico	30
6	Fotovoltaico	30
8	Fotovoltaico	30
9	Fotovoltaico	30
10	Fotovoltaico	40
11	Fotovoltaico	10
9	CHP Diesel	310
9	CHP Pila de combustible	212
5	Pila de combustible	33
10	Pila de combustible	14
5	Batería	600
10	Batería	200

3.2 Problema de Optimización

Una vez la red de distribución anteriormente presentada haya sido modelizada, se procederá a definir un problema de optimización matemático con el objetivo de controlar el comportamiento de la red por parte del operador de la misma (DSO). Este modelo se correspondería con la capa de control presentada en la figura 1.8.

Pueden distinguirse generalmente tres niveles jerárquicos para el control de las redes de distribución inteligentes [50]:

- Un primer nivel a escala local en la que diferentes controladores permiten a los diversos sistemas de baterías y generación distribuida actuar de forma autónoma para mantener la estabilidad en la red cuando sea necesario. Estos controladores deben tener una capacidad de reacción muy elevada debido a las rápidas dinámicas que tienen lugar en las pequeñas redes de distribución y a que, generalmente, estas carecen de la inercia de las grandes redes. Para ello, usualmente se eliminan los sistemas de comunicación y los controladores actúan en repuesta a las medidas que obtienen directamente de los contadores inteligentes presentes en la red.
- En un segundo nivel se coordinan los diferentes controladores del primero. Este nivel tiene, por lo tanto, un carácter centralizado. Los controladores no precisan tener una respuesta tan rápida; reciben información clave de la red y transmiten órdenes a los controladores del primer nivel para gestionar los sistemas de generación.
- El tercer nivel, también centralizado, es el que se encarga de integrar a los dos anteriores y servir de interfaz para el control de la red por parte del DSO. Este nivel tiene en cuenta cuestiones económicas para optimizar el comportamiento de la red, tales como los precios de la electricidad. Realiza los cálculos necesarios para hallar los puntos de funcionamiento óptimos y envía esta información a los niveles inferiores. Aquí se incluye el control de los flujos de potencia, del voltaje en los nodos y del mantenimiento de las condiciones óptimas en el Punto de Acoplamiento Común (PCC).

El modelo de optimización incluido en este proyecto se corresponde con este último nivel. Representa un sistema de gestión energética (*energy management system, EMS*) que va a calcular las referencias de potencia activa y voltaje que deberían mantenerse en el PCC para asegurar el funcionamiento óptimo de la red en base a una serie de requisitos, así como determinar el comportamiento de los sistemas de generación para mantener dichas condiciones.

3.2.1 Asignación de unidades y Despacho económico

Los dos problemas de optimización que van a tratarse en este proyecto comparten la misma estructura: el problema de Asignación de Unidades (UC) y el de Despacho Económico (ED). Consisten en planificar el estado de la red para un día. El objetivo es definir el orden en el que los sistemas de generación van a ser conectados a la red y la potencia que proporcionará cada uno durante un periodo de 24 horas para reducir costes y mantener las condiciones deseadas y requisitos de la red (estos requisitos están detallados en la sección 3.2.2). La planificación de los generadores se hace en dos etapas [35].

En una primera etapa, se resuelve el problema de Asignación de Unidades. Este consiste en determinar qué generadores son los que van a estar en funcionamiento ("encenderse") el día en cuestión de forma que sus capacidades puedan suplir la demanda esperada en la red. Este problema se utiliza como herramienta de previsión y se resuelve tradicionalmente con antelación (al menos 24 horas) debido a los importantes costes de mantener generadores encendidos sin producir energía y a la lentitud de algunos generadores para ponerse en funcionamiento (centrales nucleares, por ejemplo). Por ello, se hace una previsión de los precios del mercado eléctrico y de la generación de energía renovable para resolver el problema de forma estocástica. Matemáticamente, este es un problema MINLP estocástico, donde el estado de los generadores está representado por variables binarias (encendido o apagado).

Con un plazo de antelación mucho más reducido (pocas horas o incluso minutos), se crea el problema de Despacho Económico. En este caso, el problema es determinista: ya se conocen las condiciones climatológicas, la demanda de la red y los precios de los mercados de electricidad. En esta etapa lo que se pretende es establecer qué potencia deben producir los generadores anteriormente asignados para suplir la demanda y mantener las condiciones necesarias en la red (problema

determinista NLP, ya que las variables binarias han sido fijadas anteriormente según la solución del UC)

La planificación del funcionamiento de los generadores se hace de esta forma para poder gestionar en mayor medida las incertidumbres inherentes a los sistemas eléctricos (precios de la energía, cargas, generación renovable, etc.) y facilitar la resolución de los problemas matemáticos, puesto que los problemas deterministas NLP pueden resolverse más rápido que los MINLP.

3.2.2 Formulación del problema

En este apartado va a presentarse la formulación matemática del problema. Se explicarán los índices, parámetros y variables del problema, así como la función objetivo y todas las restricciones. Se presentará de forma genérica el problema en su variante determinista para evitar la acumulación de índices en las variables, en el apartado 3.2.2 puede encontrarse una descripción de como cambiaría la formulación del problema para el caso estocástico.

Sets

Los sets, tal y como presentado en el apartado 2.3.1, son los índices utilizados para indexar las variables y los parámetros del problema. Son los siguientes:

- Periodos temporales en horas, $t \in T$
- Índices de los nodos, $i \in N$
- Índices de los generadores no renovables, $g \in G$
- Índices de los generadores fotovoltaicos, $pv \in S$
- Índices de los aerogeneradores, $w \in W$
- Índices de las baterías, $bs \in B$
- Índices de las cargas residenciales, $ni \in R$
- Índices de las cargas industriales, $ind \in I$

Parámetros

Los parámetros son los elementos invariables cuyo valor son proporcionados al problema. Son los siguientes:

- Los generadores no renovables tienen un coste fijo (C_g^A) y un coste proporcional a la potencia producida (C_g^B). Igualmente, tienen límites de potencia activa, reactiva y aparente: P_g^{max} , P_g^{min} , Q_g^{max} , Q_g^{min} y S_g^{max} .
- Los generadores renovables, tanto los paneles solares como los aerogeneradores, tienen una potencia aparente máxima: S_{pv}^{max} y S_w^{max} .
- Las cargas residenciales tienen una demanda de potencia activa ($p_{ni,t}$) y reactiva ($q_{ni,t}$) fijas y determinadas por los datos presentados en el apartado 3.1.2.
- Debido a que las cargas industriales pueden ser desconectadas, el parámetro usado para definir las es la demanda máxima que pueden presentar en cada hora ($pe_{ind,t}$, $qe_{ind,t}$). Estas son calculadas a partir del perfil de la figura 3.2 y de los valores de la tabla 3.2. La desconexión de estas cargas no esenciales también tiene un coste asociado C_{ind} .
- Las baterías precisan de múltiples parámetros para ser definidas. Tienen límites de potencia P_{bs}^{max} y S_{bs}^{max} , rendimientos de carga y descarga $\eta_{c,bs}$ y $\eta_{d,bs}$ y parámetros para definir los límites del estado de carga SOC_{bs}^{max} , SOC_{bs}^{min} y SOC_{bs}^{init} .

- Las líneas de la red precisan de parámetros para definirlos. Estos derivan de los presentados en la tabla 3.1 y son la susceptancia, la conductancia y la potencia aparente máxima en las líneas: G_{ij} , B_{ij} y S_{ij}^{max} .
- Otros parámetros asociados a la red son los límites de voltaje en los nodos V^{min} y V^{max} .
- Para representar algunas restricciones auxiliares se necesitan de parámetros como las reservas para el control primario de frecuencia (de subida ur_t y de bajada dr_t) y el porcentaje máximo de potencia reactiva importada permitido en la red $\phi_t \in [0,1]$.

Por otro lado, existen también parámetros que tienen un carácter estocástico pero que van a presentarse como deterministas en este caso general. Son los relativos a la producción de energías renovables y a los precios del mercado eléctrico:

- Los precios horarios de la electricidad importada por la red son representados con π_t .
- La potencia producida por los generadores fotovoltaicos es $p_{pv,t}$ y la de los aerogeneradores $p_{w,t}$.

Variables

Las variables son los elementos cuyo valor el programa calculará para encontrar la solución óptima. Son los siguientes:

- Las potencias producidas por los generadores son variables $(p_{g,t}, q_{g,t})$, así como su estado (encendido o apagado), que se representa por la variable binaria $u_{g,t}$.
- La potencia reactiva de los generadores renovables también es variable: $q_{pv,t}$ y $q_{w,t}$.
- Como ya se ha mencionado, las demandas de la industria pueden ser reducidas o directamente desconectadas para mantener las condiciones en la red. Esto las convierte en variables $(p_{ind,t}, q_{ind,t})$ y hace necesario crear una variable que represente el porcentaje de conexión de estas cargas: $ls_{ind,t} \in [0,1]$ (siendo 0 totalmente desconectada).
- El uso de las baterías está definido por la potencias de carga y descarga, potencia aparente y el estado de carga en cada hora: $pc_{bs,t}$, $pd_{bs,t}$, $q_{bs,t}$ y $soc_{bs,t}$.
- Las variables de estado de la red, el voltaje en los nodos, se representa por el fasor $V_{i,t} \angle \theta_{i,t}$ y la potencia aparente en las líneas es $s_{ij,t}$. El ángulo de los nodos conectados a los transformadores se establece como referencia a 0° .

Función objetivo

La función objetivo representa los costes de la red y la finalidad del programa es minimizarla. Incluye los costes de importar energía a la red, los costes de generación de los generadores no renovables y los costes de desconexión de las cargas industriales para un periodo de 24 horas:

$$\sum_{t \in T} \left(\pi_t \cdot mp_t + \sum_{g \in G} u_{g,t} (C_g^A + C_g^B \cdot p_{g,t}) + \sum_{ind \in I} C_{ind} ([1 - ls_{ind,t}] \cdot pe_{ind,t}) \right) \quad (3.1)$$

En esta ecuación, mp_t representa la energía importada por la red. Al estar dividido el problema en periodos de una hora, todas las potencias son equivalentes a la energía consumida durante esa hora.

Restricciones

Las restricciones del problema constituyen los límites operacionales y de transmisión de los elementos de la red, así como algunos requisitos adicionales que pueden ser utilizados por el DSO para asegurar el correcto funcionamiento o anticiparse a posibles problemas [47] [35].

- Balances de potencia en los nodos: la potencia producida en los generadores e importada por la red debe ser igual a la potencia consumida por las cargas y las pérdidas en las líneas (representadas por los componentes de la matriz de admitancia G_{ij} y B_{ij}). Este balance debe cumplirse tanto para la potencia activa como para la reactiva:

$$\begin{aligned} \sum_{g \in Ci} P_{g,t} + \sum_{pv \in Ci} P_{pv,t} + \sum_{w \in Ci} P_{w,t} + \sum_{bs \in Ci} (pd_{bs,t} - pc_{bs,t}) - \sum_{ni \in Ci} P_{ni,t} - \sum_{ind \in Ci} P_{ind,t} \\ = \sum_{j \in N} V_{i,t} \cdot V_{j,t} (G_{ij} \cdot \cos \phi_{ij,t} + B_{ij} \cdot \sin \phi_{ij,t}) \quad \forall i, \forall t \end{aligned} \quad (3.2)$$

$$\begin{aligned} \sum_{g \in Ci} q_{g,t} + \sum_{pv \in Ci} q_{pv,t} + \sum_{w \in Ci} q_{w,t} + \sum_{bs \in Ci} q_{bs,t} - \sum_{ni \in Ci} q_{ni,t} - \sum_{ind \in Ci} q_{ind,t} \\ = \sum_{j \in N} V_{i,t} \cdot V_{j,t} (G_{ij} \cdot \sin \phi_{ij,t} - B_{ij} \cdot \cos \phi_{ij,t}) \quad \forall i, \forall t \end{aligned} \quad (3.3)$$

Los ángulos $\phi_{ij,t}$ son calculados como la diferencia entre el desfase de los nodos i y j : $\phi_{ij,t} = \theta_{i,t} - \theta_{j,t}$.

Las ecuaciones del balance de potencia se obtienen a partir del cálculo de la potencia compleja (S_B) y la corriente (I_B) en una barra en función de la matriz de admitancia (Y_B) y del vector de voltajes de barra (V_B) [33]:

$$S_B = B_B \cdot I_B^* \quad (3.4)$$

$$S_i = P_i + jQ_i \quad (3.5)$$

$$I_B = Y_B \cdot V_B \quad (3.6)$$

- Límites operacionales de la red: el voltaje en los nodos y la corriente que circula por las líneas tienen límites operacionales.

$$V^{min} \leq V_{i,t} \leq V^{max} \quad \forall i, \forall t \quad (3.7)$$

$$s_{ij,t} \leq S_{ij}^{max} \quad \forall i, \forall t \quad (3.8)$$

El voltaje en los nodos de acoplamiento común (1 y 12) será establecido por el transformador de la subestación que los alimenta y se le permitirá un margen de variación más limitado que en el resto de los nodos, un 1 %.

La potencia aparente en las líneas se calcula a partir de la intensidad que circula por las líneas:

$$|I_{ij,t}|^2 = [(V_{i,t} \cdot \cos \theta_{i,t} - V_{j,t} \cdot \cos \theta_{j,t}) + (V_{i,t} \cdot \sin \theta_{i,t} - V_{j,t} \cdot \sin \theta_{j,t})] (G_{ij}^2 + B_{ij}^2) \quad (3.9)$$

sabiendo que $I_{ij} = \frac{S_{ij}\sqrt{3}}{V}$

- Límites de los generadores: los generadores no renovables tienen límites de potencia activa y reactiva. Aquellos que alimentan la red a través de un inversor (todos excepto el generador diésel), pueden producir o consumir potencia reactiva dentro de unos límites establecidos por la potencia aparente máxima de los inversores. Es necesario incluir las variables binarias de los generadores (encendido/apagado) en estas restricciones para que la potencia máxima sea cero cuando su estado es apagado.

$$u_{g,t} \cdot P_g^{min} \leq p_{g,t} \leq u_{g,t} \cdot P_g^{max} \quad \forall g, \forall t \quad (3.10)$$

$$u_{g,t} \cdot Q_g^{min} \leq q_{g,t} \leq u_{g,t} \cdot Q_g^{max} \quad \forall g, \forall t \quad (3.11)$$

$$(q_{g,t})^2 \leq (S_g^{max})^2 - (p_{g,t})^2 \quad \forall g \neq CHPD, \forall t \quad (3.12)$$

- Potencia reactiva en los generadores renovables: estos generadores deben cumplir la misma restricción que los anteriores en cuanto a potencia reactiva se refiere.

$$(q_{pv,t})^2 \leq (S_{pv}^{max})^2 - (p_{pv,t})^2 \quad \forall pv, \forall t \quad (3.13)$$

$$(q_{w,t})^2 \leq (S_w^{max})^2 - (p_{w,t})^2 \quad \forall w, \forall t \quad (3.14)$$

- Límites y estado de carga de las baterías: las baterías deben respetar los límites de potencia máxima (tanto producida como consumida), los límites del inversor y los límites de carga. Se añade igualmente una ecuación para hacer el balance de la energía almacenada (estado de carga).

$$pd_{bs,t} \leq P_{bs}^{max} \quad \forall bs, \forall t \quad (3.15)$$

$$pc_{bs,t} \leq P_{bs}^{max} \quad \forall bs, \forall t \quad (3.16)$$

$$(pd_{bs,t} - pc_{bs,t})^2 + (q_{bs,t})^2 \leq (S_{bs}^{max})^2 \quad \forall bs, \forall t \quad (3.17)$$

$$SOC_{bs}^{min} \leq soc_{bs,t} \leq SOC_{bs}^{max} \quad \forall bs, \forall t \quad (3.18)$$

$$soc_{bs,t} = soc_{bs,t-1} + \eta_c \cdot pc_{bs,t} - pd_{bs,t} / \eta_d \quad \forall bs, \forall t \quad (3.19)$$

No es necesario incluir una restricción para especificar que una batería no puede cargarse y descargarse para una misma hora t ya que el programa evitará estas situaciones de manera natural al optimizar el problema gracias a la presencia de rendimientos de carga y descarga.

- Las cargas industriales pueden ser desconectadas durante algunas horas para disminuir la demanda de la red cuando los precios del mercado eléctrico son demasiado elevados o los generadores no tienen capacidad suficiente para alimentar a todos los consumidores. La posibilidad de desconectarlas hace necesario definir restricciones para calcular la demanda en función del porcentaje de desconexión (variable $ls_{ind,t}$) y de la potencia máxima de las cargas para la hora t ($pe_{ind,t}$).

$$p_{ind,t} = ls_{ind,t} \cdot pe_{ind,t} \quad \forall ind, \forall t \quad (3.20)$$

$$q_{ind,t} = ls_{ind,t} \cdot qe_{ind,t} \quad \forall ind, \forall t \quad (3.21)$$

Restricciones auxiliares

Además de las condiciones anteriores, que deben cumplirse para asegurar el correcto funcionamiento de la red, pueden identificarse algunas restricciones adicionales que puede ser exigidas por el DSO [35]. Estas restricciones auxiliares pueden ser contratadas por el operador o exigidas en tiempo real para mantener el balance en la red.

- Potencia reactiva máxima en la red: puede exigirse mantener un porcentaje máximo de potencia reactiva respecto a la activa en la potencia importada por la red.

$$mq_t \leq \phi_t \cdot mp_t \quad \forall t \quad (3.22)$$

donde mq_t y mp_t son respectivamente las potencias reactivas y activas importadas por la red en la hora t .

- Reservas para el control primario de frecuencia: esta restricción está relacionada con el primer nivel de control de la *smart grid*. Son las llamadas reservas rotativas. Cuando sea necesario, los controladores del primer nivel gestionarán los generadores a nivel local aumentando o reduciendo su potencia con el fin de mantener la frecuencia de la red en 50 Hz. Sin embargo, es el control centralizado el que debe garantizar que dichos generadores tienen capacidad suficiente para adaptar su producción. Esto se consigue estableciendo niveles de reserva de potencia activa, tanto de subida (ur_t) como de bajada (dr_t). Una reserva de potencia de subida implica que los generadores o baterías tienen que ser capaces de inyectar dicha potencia a la red. Una reserva de potencia de bajada significa que los generadores deben poder reducir su potencia o las baterías aumentar su demanda en la medida establecida por la reserva:

$$\sum_{g \in G} u_{g,t} (P_g^{max} - p_{g,t}) + \sum_{bs \in B} ur_{bs,t} \geq ur_t \quad \forall t \quad (3.23)$$

$$\sum_{g \in G} u_{g,t} (p_{g,t} - P_g^{min}) + \sum_{bs \in B} dr_{bs,t} \geq dr_t \quad \forall t \quad (3.24)$$

donde $ur_{bs,t}$ es el incremento de potencia que puede proporcionar la batería bs en la hora t y $dr_{bs,t}$ el margen de reducción de potencia o aumento de demanda que tiene.

En estas ecuaciones se considera que únicamente los generadores y los sistemas de baterías pueden ser utilizados para control de frecuencia, aunque también podría considerarse añadir la reducción de consumo de las cargas no esenciales. Esto último se corresponde con las reservas rotativas mencionadas en la sección 1.4.

El aumento/reducción de potencia que una batería podría proporcionar es calculado según su potencia máxima y la potencia que estén produciendo o consumiendo durante la hora t .

$$ur_{bs,t} = P_{bs}^{max} - pd_{bs,t} + pc_{bs,t} \quad (3.25)$$

$$dr_{bs,t} = P_{bs}^{max} - pc_{bs,t} + pd_{bs,t} \quad (3.26)$$

Si una batería que está descargándose necesita aumentar la potencia inyectada en la red, solo podrá hacerlo con el margen disponible hasta la potencia máxima. Por otro lado, si la batería está cargándose, la batería tiene un margen mayor de incremento de la potencia inyectada, ya que podría dejar de consumir la potencia de carga y pasar a descargarse con la potencia máxima. La figura 3.3 ilustra cómo se calcula la capacidad que tienen un generador produciendo una potencia p_t y una batería consumiendo p_c para aumentar su producción y asegurar una reserva de subida.

Evidentemente, las baterías deben tener suficiente energía almacenada (o margen de carga) para hacer frente a estas variaciones. Sin embargo, hay que tener en cuenta que el control de frecuencia se realizaría únicamente durante escasos minutos en una hora t (se van a considerar 10 minutos). Por lo tanto, la energía almacenada/capacidad libre debe ser mayor que la potencia a producir/consumir dividida entre 6 (equivalente a carga/descargar durante 10 minutos).

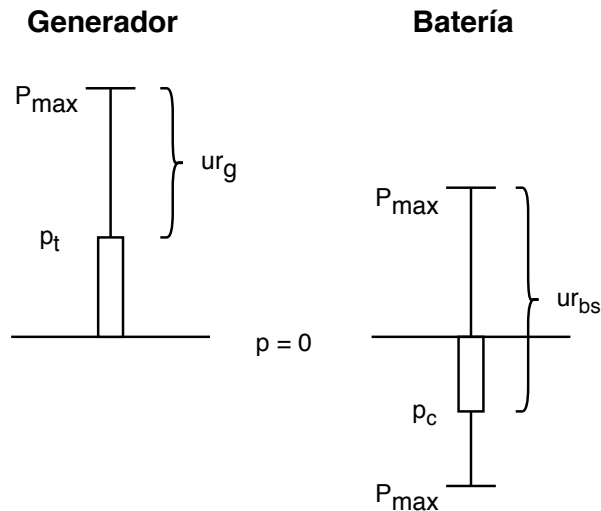


Figura 3.3 Incremento de potencia disponible para control primario de frecuencia (reserva de subida).

Por ejemplo, para las reservas de subida el razonamiento es el siguiente: si la capacidad de aumento de la potencia inyectada, calculada con la ecuación 3.25, dividida entre 6 es mayor que la energía disponible ($\frac{ur_{bs,t}}{6} > soc_{bs,t} - SOC_{bs}^{min}$), significa que la batería no tiene suficiente energía para proporcionar la potencia necesaria durante diez minutos, por lo que la capacidad de las baterías de cara al control de frecuencia pasa a ser la energía almacenada $ur_{bs,t} = (soc_{bs,t} - SOC_g^{min})$.

Para implementar esta comprobación en el código de Pyomo es necesario iterar en las soluciones. Primero ha de resolverse el problema de optimización sin tener en cuenta las limitaciones de energía en las baterías. En esta solución se comprueba si alguna de las baterías no tendría suficiente energía o margen de carga para asegurar la reserva durante 10 minutos. Si es el caso, se incluyen restricciones adicionales para tener en cuenta los niveles de carga en las baterías y horas específicas que incumplan la condición. Posteriormente, se comprueba si han surgido nuevos casos problemáticos y se continúa iterando si fuese necesario.

Formulación completa

En la tabla 3.4 puede encontrarse la formulación del problema completo para el problema determinista.

Problema estocástico

El problema estocástico tiene la peculiaridad de contar con parámetros de naturaleza aleatoria. Para abordar esta cuestión, se crean diferentes escenarios con los valores más característicos que pueden adoptar los parámetros y se le atribuye una probabilidad de ocurrencia a cada uno. Los valores de las variables en la solución de cada escenario son, por lo tanto, diferentes.

Esto implica que para la formulación del problema estocástico es necesario crear un set adicional por cada parámetro estocástico presente en el problema:

- Escenarios de precio: $p \in \Omega_p$ con una probabilidad asociada a cada uno μ_p que resultan en los precios horarios del mercado eléctrico π_t^p .
- Escenarios de generación fotovoltaica: $s \in \Omega_s$ con una probabilidad asociada a cada uno μ_s que resultan en las potencias generadas $p_{pv,t}^s$.
- Escenarios de generación eólica: $v \in \Omega_v$ con una probabilidad asociada a cada uno μ_v que resultan en las potencias generadas $p_{w,t}^v$.

Tabla 3.4 Formulación del problema optimización determinista.

Minimize	$\sum_{t \in T} \left(\pi_t \cdot mp_t + \sum_{g \in G} u_{g,t} (C_g^A + C_g^B \cdot p_{g,t}) + \sum_{ind \in I} C_{ind} ([1 - ls_{ind,t}] \cdot pe_{ind,t}) \right)$
Sujeto a:	$\sum_{g \in Ci} p_{g,t} + \sum_{pv \in Ci} p_{pv,t} + \sum_{w \in Ci} p_{w,t} + \sum_{bs \in Ci} (pd_{bs,t} - pc_{bs,t}) - \sum_{ni \in Ci} p_{ni,t} - \sum_{ind \in Ci} p_{ind,t} = \sum_{j \in N} V_{i,t} \cdot V_{j,t} (G_{ij} \cdot \cos \phi_{ij,t} + B_{ij} \cdot \sin \phi_{ij,t}) \quad \forall i, \forall t$ $\sum_{g \in Ci} q_{g,t} + \sum_{pv \in Ci} q_{pv,t} + \sum_{w \in Ci} q_{w,t} + \sum_{bs \in Ci} q_{bs,t} - \sum_{ni \in Ci} q_{ni,t} - \sum_{ind \in Ci} q_{ind,t} = \sum_{j \in N} V_{i,t} \cdot V_{j,t} (G_{ij} \cdot \sin \phi_{ij,t} - B_{ij} \cdot \cos \phi_{ij,t}) \quad \forall i, \forall t$ $V^{min} \leq V_{i,t} \leq V^{max} \quad \forall i, \forall t$ $s_{ij,t} \leq S_{ij}^{max} \quad \forall i, \forall t$ $u_{g,t} \cdot P_g^{min} \leq p_{g,t} \leq u_{g,t} \cdot P_g^{max} \quad \forall g, \forall t$ $u_{g,t} \cdot Q_g^{min} \leq q_{g,t} \leq u_{g,t} \cdot Q_g^{max} \quad \forall g, \forall t$ $(q_{g,t})^2 \leq (S_g^{max})^2 - (p_{g,t})^2 \quad \forall g \neq CHPD, \forall t$ $(q_{pv,t})^2 \leq (S_{pv}^{max})^2 - (p_{pv,t})^2 \quad \forall pv, \forall t$ $(q_{w,t})^2 \leq (S_w^{max})^2 - (p_{w,t})^2 \quad \forall w, \forall t$ $pd_{bs,t} \leq P_{bs}^{max} \quad \forall bs, \forall t$ $pc_{bs,t} \leq P_{bs}^{max} \quad \forall bs, \forall t$ $(pd_{bs,t} - pc_{bs,t})^2 + (q_{bs,t})^2 \leq (S_{bs}^{max})^2 \quad \forall bs, \forall t$ $SOC_{bs}^{min} \leq soc_{bs,t} \leq SOC_{bs}^{max} \quad \forall bs, \forall t$ $soc_{bs,t} = soc_{bs,t-1} + \eta_c \cdot pc_{bs,t} - pd_{bs,t} / \eta_d \quad \forall bs, \forall t$ $p_{ind,t} = ls_{ind,t} \cdot pe_{ind,t} \quad \forall ind, \forall t$ $q_{ind,t} = ls_{ind,t} \cdot qe_{ind,t} \quad \forall ind, \forall t$ $mq_t \leq \phi_t \cdot mp_t \quad \forall t$ $\sum_{g \in G} u_{g,t} (P_g^{max} - p_{g,t}) + \sum_{bs \in B} ur_{bs,t} \geq ur_t \quad \forall t$ $\sum_{g \in G} u_{g,t} (p_{g,t} - P_g^{min}) + \sum_{bs \in B} dr_{bs,t} \geq dr_t \quad \forall t$

La principal diferencia con la formulación presentada anteriormente radica en el hecho de que tanto los parámetros estocásticos como las variables, estarán indexados por estos sets. La función objetivo también se ve alterada y se formula de la siguiente forma:

$$\sum_{p \in \Omega_p} \mu_p \cdot \sum_{s \in \Omega_s} \mu_s \cdot \sum_{v \in \Omega_v} \mu_v \cdot \sum_{t \in T} \left(\pi_t^p \cdot mp_t^{p,s,v} + \sum_{g \in G} u_{g,t} (C_g^A + C_g^B \cdot p_{g,t}^{p,s,v}) + \sum_{ind \in I} C_{ind} ([1 - ls_{ind,t}^{p,s,v}] \cdot pe_{ind,t}) \right) \quad (3.27)$$

3.2.3 Recapitulación

En este apartado se ha presentado el problema de optimización que será modelizado con Pyomo. Se ha formulado el problema determinista completo, presentado los índices, los parámetros, las variables, la función objetivo y las restricciones operacionales y auxiliares, y se ha introducido la formulación del problema estocástico.

Este problema representa un sistema gestión de energía de *smart grid* cuya finalidad es determinar cuáles son los valores óptimos de generación y consumo de las cargas no esenciales para mantener las condiciones necesarias en la red y se minimizar los costes asociados.

Con el fin de resolver el problema de Asignación de Unidades y tratar la naturaleza aleatoria de algunos parámetros, el modelo estocástico puede ser utilizado para determinar qué generadores se pondrán en funcionamiento en cada hora. Una vez hayan desaparecido las incertidumbres, se resolvería el problema de Despacho Económico con el modelo determinista fijando el estado de cada generador según la solución del problema anterior. De esta forma se ajustaría con exactitud la generación y la demanda en la red.

3.3 Estructura del Programa

En esta sección se presentará brevemente la estructura del programa creado y se darán ejemplos representativos en los que se aprecie la sintáxis propia de Pyomo. El código completo puede encontrarse en el Apéndice A.

En primer lugar hay un bloque dedicado a la declaración de sets, parámetros y variables. Este bloque corresponde a la modelización de la red de distribución en sí. Siempre debe crearse primero un modelo (en este caso abstracto, ya que los valores se inicializarán más adelante) que incluirá a los demás elementos. En el siguiente código se muestra a título informativo la declaración del modelo, de un set T que va desde 1 hasta el número de horas definido, de un set I que sirve para indexar respecto a los nodos y de una variable (el voltaje en los nodos V_i) que puede tomar valores reales no negativos, cuyo valor se inicializa a 20 kV y que está indexada mediante los sets anteriores:

Código 3.1 Declaración del modelo, sets y variables en Pyomo.

```
# Creación del modelo
m = AbstractModel()

# Indices i y t de los nodos
m.T = RangeSet(1, m.h)
m.I = RangeSet(1,m.n)

# Voltaje en los nodos
m.V_i = Var(m.I, m.T, within=NonNegativeReals, initialize=20)
```

Algunos parámetros o variables son declarados mediante funciones, como es el caso de la potencia producida por los paneles fotovoltaicos (p_{pv}) que se forma para cada hora t a partir de un factor de producción f_{pv} multiplicado por la potencia máxima del sistema fotovoltaico $P_{max_{pv}}$. La siguiente función actúa como sendos bucles *for*, iterando sucesivamente respecto al número de horas definido y a los generadores fotovoltaicos:

Código 3.2 Declaración de parámetros mediante función en Pyomo.

```
# Declaración de la potencia PV
```

```
def p_pv_init(m, i, pv, t):
    return m.f_pv[t]*m.Pmax_pv[i,pv]

m.p_pv = Param(m.PV, m.T, within=NonNegativeReals, initialize=p_pv_init)
```

Un segundo bloque está formado por las restricciones y la función objetivo. Representan el problema de optimización propiamente hablando, es decir, el programa de gestión de la energía eléctrica en la red. La declaración de la función objetivo y las restricciones sigue una lógica parecida a la anterior, realizándose a través de funciones. Es importante aclarar que la importación de energía desde los transformadores ha sido modelizada como generadores sin límite y que no están sujetos a las mismas restricciones que estos.

Código 3.3 Declaración de función objetivo y restricciones en Pyomo.

```
# Declaración función objetivo
def obj_expression(m):
    return (sum(m.price[t]*(m.p_g[1,"Red",t] + m.p_g[12,"Red",t])
            + sum(m.u_g[i,g,t]*(m.CosteB_g[i,g]*m.p_g[i,g,t]+m.CosteA_g[i,g])
                for (i,g) in m.G if g != "Red")
            + sum(m.C_ind[i,ind]*(m.pe_ind[i,ind,t]*(1 - m.ls[i,ind,t]))) for
                (i,ind) in m.IND) for t in m.T))

m.OBJ = Objective(rule=obj_expression)

# Limites de voltaje en nodos
def Vbounds_const_rule(m, i, t):
    return m.Vmin <= m.V_i[i,t] <= m.Vmax

m.Vbounds_const = Constraint(m.I, m.T, rule=Vbounds_const_rule)
```

El siguiente bloque corresponde a la introducción de los datos. Como se detallará en el apartado 3.4, los parámetros son importados a partir de un archivo Excel para facilitar su manipulación. Esto se consigue a través de un elemento *DataPortal*. El programa busca en el archivo indicado el valor asignado a un rango de Excel y lo asocia al parámetro o set especificado. En lugar de un único valor, puede ser que el rango incluya múltiples valores, como es el siguiente caso en el cual se inicializa el set *ARCS* mediante una matriz que incluye información sobre qué nodos están directamente conectados por líneas eléctricas. El hecho de inicializar los valores supone crear un modelo concreto de Pyomo a partir del abstracto. Para ello se crea un instancia (*instance*) con la que se trabajará a partir de ese momento, puesto que contiene los elementos con sus valores ya asignados:

Código 3.4 Asignación de valores a los parámetros de Pyomo a partir de Excel.

```
data = DataPortal()

# Importa los parámetros de entrada
data.load(filename='Datos.xlsx', range='horas', param='h')
data.load(filename='Datos.xlsx', range='nodos', param='n')

# Carga los índices de la red
data.load(filename='No_escen.xlsx', range='arcos', format='set_array',
          set=m.ARCS)
```

```
instance = m.create_instance(data)
```

Por último, se incluyen las líneas de código dedicadas a la resolución del problema matemático. Como ya se ha indicado, en este caso se resolverá a través de la interfaz NEOS y mediante el solver BONMIN. También se crea un archivo *.pickle* que guarda la solución encontrada para su posterior uso. Pueden incluirse igualmente líneas de código dedicadas a la creación de un archivo PDF con gráficos y su presentación por pantalla:

Código 3.5 Resolución del problema matemático y creación de gráficas en Pyomo.

```
# Para resolver remotamente a través de la interfaz NEOS
solver_manager = SolverManagerFactory('neos')
results = solver_manager.solve(instance, opt='baron')

# Guarda la solución en results
instance.solutions.store_to(results)

# Archiva results: solución & metadatos de ejecución
file = open("results.pickle", "wb")
pickle.dump(results, file)
file.close()

# Gráfica del perfil de las cargas residenciales
fig4 = plt.figure()
fig4, f4_ax = plt.subplots()

carga = []
for t in horas:
    carga.append(value(instance.f_house[t]))

f4_ax.plot(horas, carga)
f4_ax.set_ylim([0,1])
f4_ax.set(xlabel = 'Horas', ylabel='Potencia activa [p.u.]',
          title='Perfil de las cargas residenciales')
f4_ax.grid()
fig4.savefig("cargas.pdf", bbox_inches='tight')
plt.show()
```

3.4 Importación de Datos

En esta sección se detallarán datos más específicos que son necesarios para el funcionamiento de la red. Algunos surgen de investigaciones o valores reales habitualmente utilizados y otros son suposiciones. Para facilitar la manipulación de los datos, estos son importados por el programa desde un archivo Excel.

3.4.1 Datos deterministas

Los datos deterministas son aquellos que se utilizan como base de la simulación y muchos de ellos también estarán presentes en la simulación del caso estocástico.

Datos de la red

Además de los datos ya indicados en la sección 3.1, a continuación se muestran ciertos parámetros necesarios para definir el problema determinista:

Tabla 3.5 Parámetros de la red [45].

Parámetro	Valor
Horas	24
Nodos	14
V_{base}	20 kV
Desviación de voltaje máxima permitida	5 %
S_{max}	10 MVA
ϕ	30 %

donde ϕ es el porcentaje máximo de potencia reactiva importada con respecto a la potencia activa y que equivale a un factor de potencia ($\cos \phi$) aproximado de 0,95.

Costes

Los costes de los generadores y de desconexión de la industria han sido supuestos y tienen un carácter instrumental, ya que serán modificados para comprobar el correcto funcionamiento del programa en diferentes casos. A los generadores se les ha asignado un coste fijo instrumental para evitar que sean encendidos cuando no sean necesarios. En este caso, la industria tiene un coste de desconexión igualmente instrumental para evitar que sean desconectadas en condiciones normales de precio del mercado eléctrico.

Tabla 3.6 Costes de la red.

Elemento	Coste fijo (C_A) [€/MWh]	Coste variable (C_B) [€/MWh]
CHPD	20	150
FC	10	100
CHPFC	10	100
Industria	200	-

El coste de importación de la electricidad ha sido calculado a partir de los datos del mercado eléctrico español para el año 2018 en periodos de 24 horas [9]. Se han tratado los datos para eliminar los fines de semana y se ha hecho una media de los días restantes. Estos precios han sido incrementados en un 30 % para determinar cuál será el coste de importar energía a la red en el modelo determinista (ver figura 3.4).

Generación renovable

Para hallar el perfil de generación de los aerogeneradores, se ha seguido un procedimiento similar al anterior a partir de los datos presentados en [6] para el año 2018 en Oporto. Los datos de

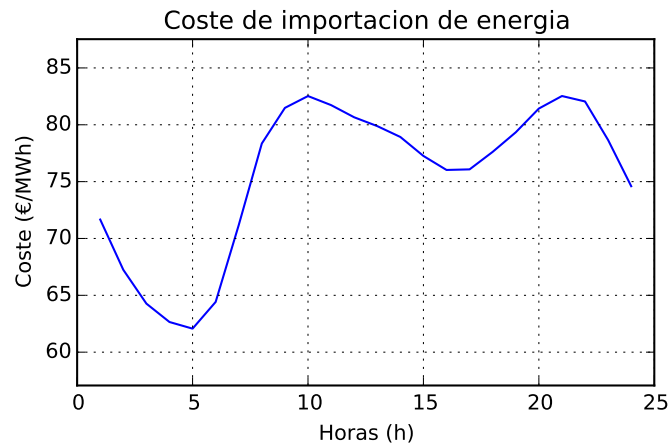


Figura 3.4 Coste medio de importación de energía a la red.

generación fotovoltaica han sido calculados a partir de 26 días. Ambos perfiles son utilizados en el caso determinista y pueden ser encontrados en la figura 3.5. La potencia reactiva producida por estos generadores es una variables del problema, y debe cumplir las ecuaciones 3.13 y 3.14. Para ello, se ha establecido una potencia aparente máxima (S_{pv}^{max} y S_w^{max}) un 50% mayor que la potencia activa máxima de cada generador renovable.

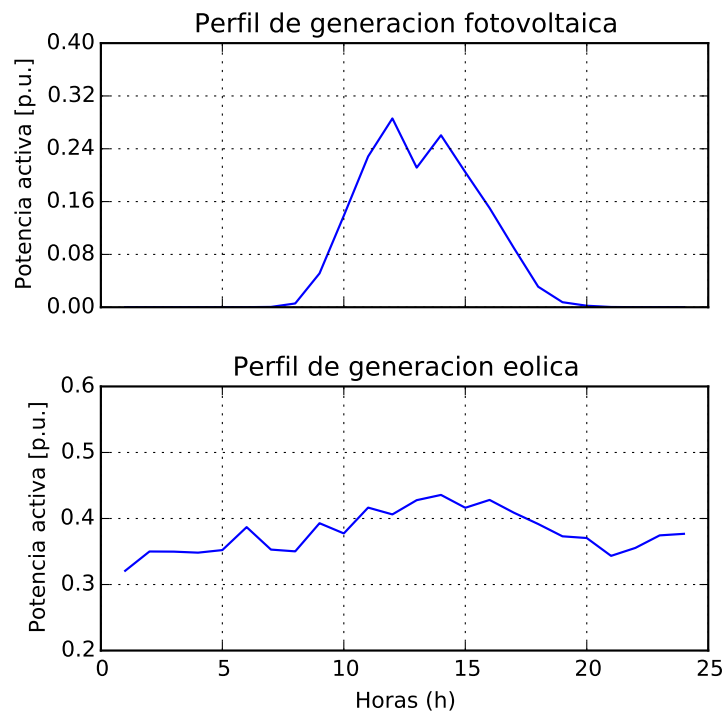


Figura 3.5 Perfiles medios de generación renovable.

Baterías

Además de los datos de potencia máxima indicados en la tabla 3.3, es necesario indicar los valores de los otros parámetros de las baterías. La capacidad de almacenamiento de las baterías se ha establecido en 3 veces el valor de potencia máxima. El estado de carga mínimo que debe mantenerse será del 5% de la capacidad, y el estado inicial se establece para ambas baterías en este valor. Los

rendimientos de carga y descarga serán del 95 % en ambos casos y la potencia aparente máxima será de 5 MVA.

Tabla 3.7 Parámetros de las baterías.

(a) Batería nodo 5.		(b) Batería nodo 200.	
Parámetro	Valor	Parámetro	Valor
SOC^{max}	1,8 MWh	SOC^{max}	0,6 MWh
SOC^{min}	0,9 MWh	SOC^{min}	0,03 MWh
SOC^{init}	0,9 MWh	SOC^{init}	0,03 MWh
η_{carga}	0,95	η_{carga}	0,95
$\eta_{descarga}$	0,95	$\eta_{descarga}$	0,95
S^{max}	5 MVA	S^{max}	5 MVA

3.5 Estructura y Datos del Problema Estocástico

Con la intención de mitigar el efecto de la incertidumbre que por naturaleza tienen los precios del mercado eléctrico y la generación renovable, se crean diferentes escenarios con una probabilidad de ocurrencia asociada a cada uno. Para generar estos escenarios a partir de los datos de un año, se ha recurrido a un modelo de código abierto desarrollado en R que agrupa los datos longitudinales (24 valores para cada día) según sus similitudes, tanto de valor como de forma de la curva, y asocia una probabilidad a cada agrupación [44]. Se han encontrado 4 escenarios para cada uno de los parámetros estocásticos.

3.5.1 Escenarios de precios

Las probabilidades asociadas a los escenarios de precio son 0,38, 0,02, 0,36 y 0,25 para los escenarios 1, 2, 3 y 4 respectivamente. En la figura 3.6 puede encontrarse el resultado del modelo en R. En las imágenes superiores aparecen los datos iniciales agrupados y en las inferiores los valores medios de cada escenario. Los ejes de abscisas representan las 24 horas del día y los de ordenadas el precio del mercado eléctrico en €/MWh.

Como se ha explicado anteriormente, se considera que este precio se encarecerá en un 30% para la red, por lo que los escenarios finales pueden verse en la figura 3.7. El coste máximo de importación es de 94,22 €/MWh, el mínimo 28,33 €/MWh y el medio es 67,28 €/MWh.

3.5.2 Escenarios de generación eólica

Las probabilidades asociadas a los escenarios de generación eólica son respectivamente 0,33, 0,29, 0,21 y 0,17. En la figura 3.8 puede encontrarse el resultado del modelo en R usado para identificarlos. En las imágenes superiores aparecen los datos iniciales agrupados y en las inferiores los valores medios de cada escenario. Los ejes de abscisas representan las 24 horas del día y los de ordenadas la potencia generada por el aerogenerador en porcentaje de potencia activa máxima.

Los escenarios finales pueden encontrarse en la figura 3.9. La generación máxima será el 68,66 % de la capacidad máxima, la mínima el 6,86 % y la media el 34,73 %.

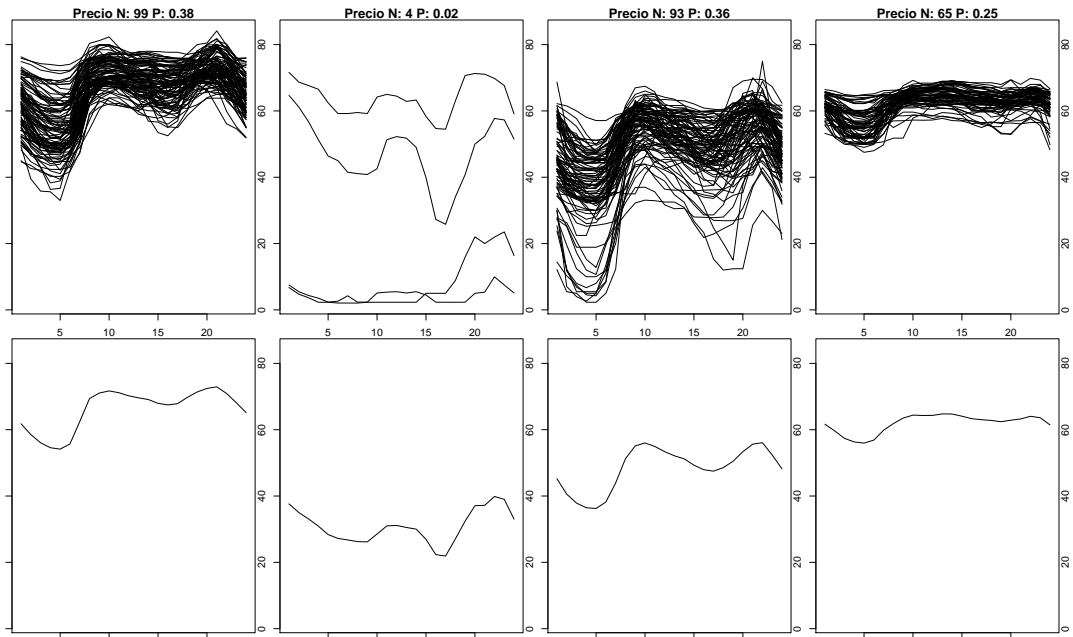


Figura 3.6 Creación de escenarios de precios del mercado eléctrico.

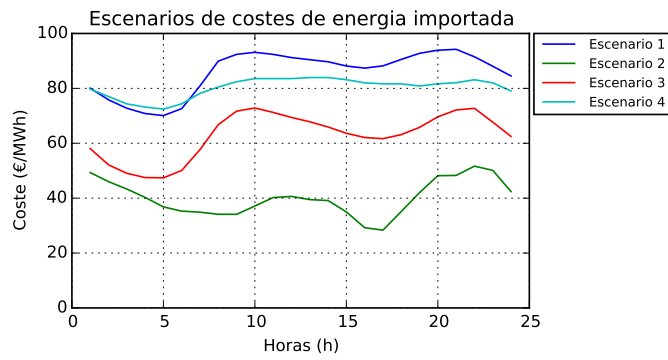


Figura 3.7 Escenarios de costes de la energía importada por la red.

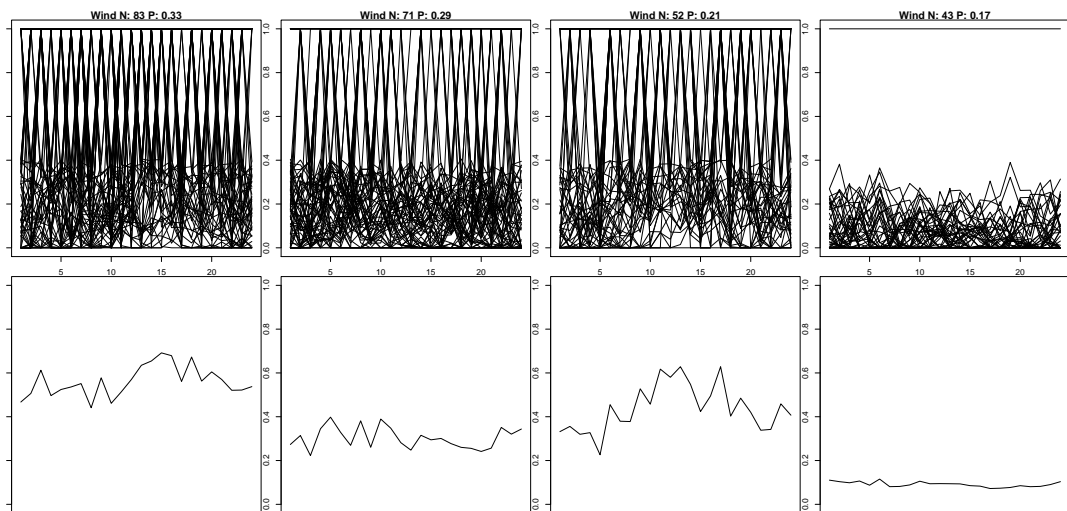


Figura 3.8 Creación de escenarios de generación de energía eólica.

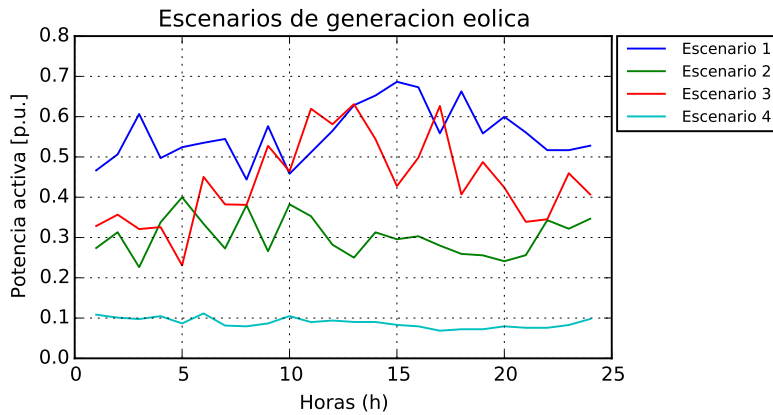


Figura 3.9 Escenarios de generación eólica.

3.5.3 Escenarios de generación fotovoltaica

Las probabilidades asociadas a los escenarios de generación de los paneles solares son respectivamente 0,2, 0,08, 0,29 y 0,43.

Los escenarios finales pueden encontrarse en la figura 3.10. La generación máxima será el 49,58 % de la potencia máxima y la media el 6,3 %. La potencia mínima en este caso es del 0% ya que durante la noche los paneles fotovoltaicos no tienen producción.

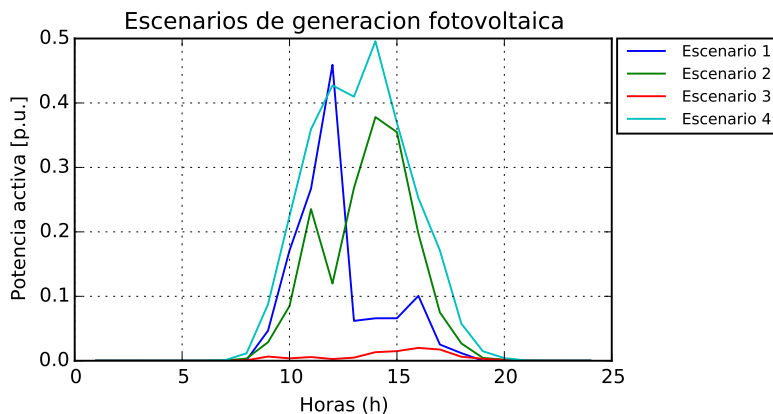


Figura 3.10 Escenarios de generación fotovoltaica.

3.5.4 Estructura del problema estocástico

La extensión PySP incluida en Pyomo será utilizada para modelizar el problema estocástico. Para ello se requiere definir un árbol de escenarios con una estructura específica en un archivo *.dat* llamado *ScenarioStructure.dat* [53]. El archivo en concreto puede encontrarse en el Apéndice B. Tan solo existen dos niveles de decisión, uno inicial de un único nodo que recogerá la solución global y otro inferior con 64 nodos (las combinaciones de los 3 parámetros estocásticos y los 4 escenarios de cada uno). En cada escenario del nivel inferior se resolverá el problema de optimización determinista completo con los valores de los parámetros estocásticos que correspondan. En el primer nivel de decisión se encuentran las variables binarias de los generadores, de forma que el programa pueda determinar de forma global qué generadores deben encenderse en función de los resultados calculados en los escenarios inferiores y de las probabilidades de cada uno. Es necesario crear también un archivo *.dat* por cada nodo creado en el archivo anterior. Los nombres de los archivos

deben corresponder con los nombres de los nodos. Igualmente, el modelo determinista que servirá como base debe tener el nombre *ReferenceModel.py*.

Como se ha comentado en la sección 2.3.3, no ha sido posible utilizar un solver de código abierto compatible con problemas MINLP. Puesto que el resultado que se pretende conseguir con este modelo es calcular la configuración óptima de los generadores con antelación al día estudiado, es necesario mantener las variables binarias, las cuales precisamente representan el estado de los generadores. Las posibles soluciones a este problema pasan por linealizar las ecuaciones del problema y convertirlo en un MIP o por relajar las variables binarias para resolver el resultante NLP y posteriormente aproximar sus valores a 0 ó 1. Se ha optado por esta segunda opción, ya que aunque PySP cuenta con una función para linealizar funciones, esta tan solo se aplica a la función objetivo, por lo que un solver MIP no podría resolver el problema resultante al existir restricciones no lineales. Por lo tanto, se utilizará el solver Ipopt [51] para resolver el NLP a través del algoritmo *Progressive Hedging* (ver sección 2.3.3). Sin embargo, tal y como se explica en [52], este método de aproximación probablemente resulte en una solución poco exacta, por lo que se revela necesario conseguir un solver MINLP adecuado para resolver el problema estocástico con mayor exactitud en el futuro.

Este programa puede ejecutarse desde el intérprete de comandos del símbolo del sistema de Windows mediante el siguiente comando:

Código 3.6 Ejecución del problema estocástico a través del terminal de comandos de Windows.

```
runph -m models -i nodedata --solver=ipopt --solution-writer=pyomo.pysp.  
plugins.csvsolutionwriter --default-rho=1
```

donde la carpeta *models* incluye el modelo *ReferenceModel.py* usado como base y *nodedata* incluye todos los archivos de texto *.dat* usados para definir el árbol de escenarios. El comando *--solution-writer=pyomo.pysp.plugins.csvsolutionwriter* crea un archivo *ph.csv* con la solución encontrada. El parámetro ρ es un coeficiente de penalidad utilizado para la convergencia de las soluciones y se establece en su valor por defecto.

4 Resultados del Modelo

En esta sección se presentarán los resultados del proyecto. Se analizarán varios casos para estudiar el correcto funcionamiento del modelo determinista y se presentará la solución del modelo estocástico. El modelo calcula para un periodo de 24 horas el coste total, el voltaje en los nodos, la energía importada por la red, el estado y la potencia producida por los generadores, la carga y descarga de las baterías y el porcentaje de desconexión de la industria.

La potencia producida por los generadores renovables será constante en todos los casos (excepto cuando se estudia el aumento de generación renovable) y se puede encontrar en la siguiente figura:

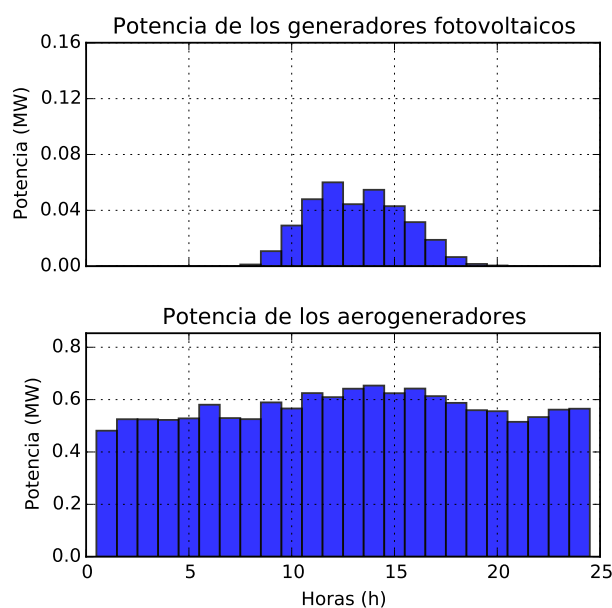


Figura 4.1 Producción de los generadores renovables.

4.1 Caso Base

Este primer caso será utilizado como referencia para comparar con los siguientes. En él las reservas de subida y bajada de potencia activa se mantienen a cero y los costes de los generadores y de desconexión de la industria son los indicados en el apartado 3.4.1.

El coste total de la red es de **44351,31 €** con una energía total importada por la red de **573,815 MWh**.

En la figura 4.3a puede apreciarse que toda la energía consumida por la red es importada a través de la subestación o producida por los generadores renovables para evitar poner en marcha los generadores convencionales. La energía importada por la red sigue el perfil del consumo total (residencial e industrial): existe un pico prolongado durante la mañana y un pico mayor por la noche debido al aumento de las cargas residenciales (que son las que tienen mayor peso en esta red). La figura 4.2 sirve de comprobación de que las industrias no se han desconectado al tener estas un coste mayor al de importación de energía.

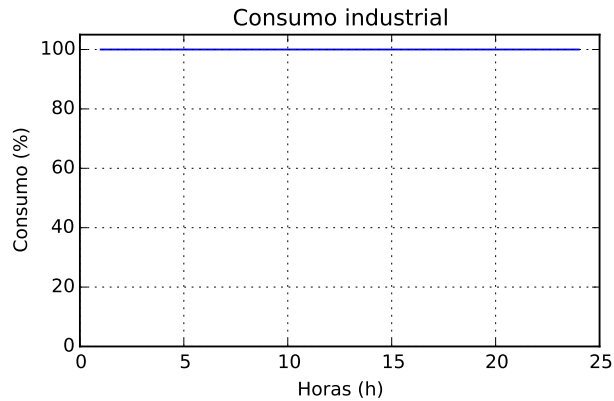


Figura 4.2 Consumo industrial en el caso base.

En la subred 1, para disminuir las pérdidas de carga en las líneas, el voltaje en el nodo de entrada (PCC) se mantiene por encima del voltaje de los nodos intermedios para disminuir las pérdidas de carga y varía entre sus límites superior e inferior ($20 \pm 1\%kV$). El voltaje en los nodos varía según la carga y la generación en los nodos; el aumento de potencia en un nodo ocasiona un incremento del voltaje y el aumento de la carga resulta en una caída. Este comportamiento va a mantenerse para todos los casos estudiados excepto cuando existe una importante generación en los nodos intermedios.

Por otro lado, la subred 2 muestra una lógica diferente en el voltaje de sus nodos. Al ser una red muy capacitiva y que está formada únicamente por cargas, las tensiones en los nodos intermedios son más elevadas que en el nodo de PCC. Esta inversión se debe a que el carácter capacitivo tiene un efecto similar al de un generador en los nodos intermedios si la red tiene cargas reducidas (esto puede apreciarse también en la subred 1 para la hora 1, ya que las cargas en la red en ese momento son mínimas). El hecho de que el voltaje del nodo 12 se mantenga en su valor mínimo es también consecuencia de las capacidades presentes en la red. Normalmente, la tensión se mantendría en su valor máximo para minimizar las pérdidas de carga pero las elevadas capacidades ocasionan fugas de corriente a tierra, por lo que un voltaje mayor ocasionaría una pérdida mayor de la corriente capacitiva y un mayor coste.

En la figura 4.3b se muestra el resultado que se obtendría si se eliminase el carácter capacitivo de las líneas. Las tensiones en los nodos PCC podrían mantenerse en su nivel máximo y ser mayores que las de los nodos intermedios para minimizar pérdidas. Esto ocurre para ambas redes, lo que implica que las horas en que la tensión del nodo 1 está en su valor mínimo es consecuencia del carácter capacitivo de las líneas.

Las baterías funcionan siguiendo ambas un mismo patrón, por lo que se representa únicamente la suma de sus potencias y estados de carga. Almacenan energía importada la red cuando el precio es menor, notablemente durante la noche, para descargarse cuando repunta, que suelen corresponderse con los momentos de mayor demanda eléctrica, y reducir el coste de funcionamiento.

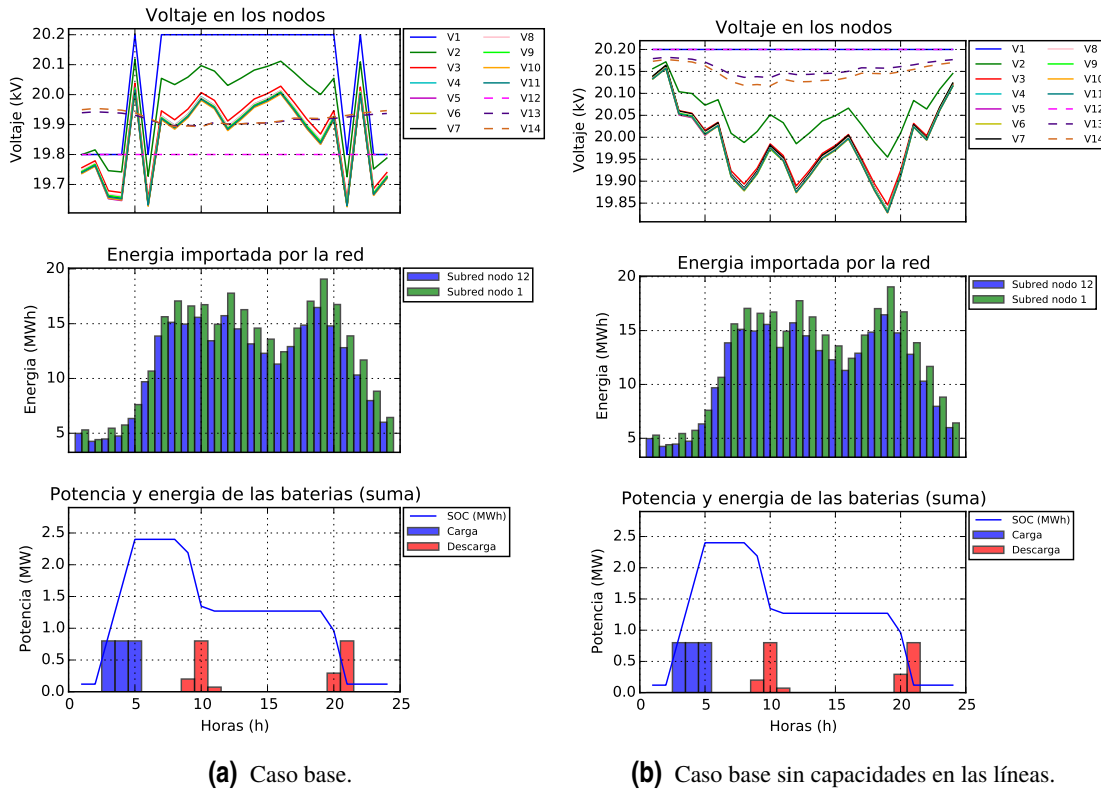


Figura 4.3 Voltajes, potencias y energía en el caso base.

4.2 Limitación de Potencia Importada

En este caso de estudio se limita la potencia que puede importar la subred 1 a 16,5 MW para estudiar el comportamiento de los generadores. Los valores del resto de parámetros se mantienen iguales a los del caso base.

El coste total de la red es de **44620,22 €** con una energía total importada por la red de **571,609 MWh**. En comparación con el caso base, el coste ha aumentado mientras que la energía importada se ha reducido debido a que parte de la demanda se ha suministrado a través de los generadores o desconectando las industrias, lo que tiene un coste mayor que importar energía.

Durante los periodos de mayor consumo, la potencia importada alcanza su nivel máximo permitido y es necesario encender los generadores para responder a la demanda. El generador que se utiliza es el de mayor capacidad dentro de los de menor coste: el sistema de cogeneración basado en pila de combustible (CHPFC).

Igualmente, es necesario que las baterías descarguen potencia a la red durante esos momentos. Para ello, las baterías se cargan durante el día, al contrario de lo que ocurría en el caso base que únicamente lo hacían cuando el precio de electricidad era menor. Esto ocurre en las horas 11, 15, 16 y 17 con el fin de que las baterías tengan suficiente energía en los momentos críticos.

Las industrias se desconectan en algunos momentos críticos, especialmente en la hora 19, ya que los generadores por sí solos no serían capaces de responder a la demanda. Para entender por qué no se utilizan todos los generadores a su máxima potencia en lugar de desconectar las industrias en menor medida hay que considerar sus costes fijos: a pesar de que el coste por MWh de desconectar la industria es mayor que el de los generadores, producir energía con ellos tendría un coste mayor para la red. Esto se demuestra con el cálculo realizado a continuación. En la hora 12, las industrias producen al 83% de su capacidad máxima. Dado que en esta hora se desconecta el 99% de la

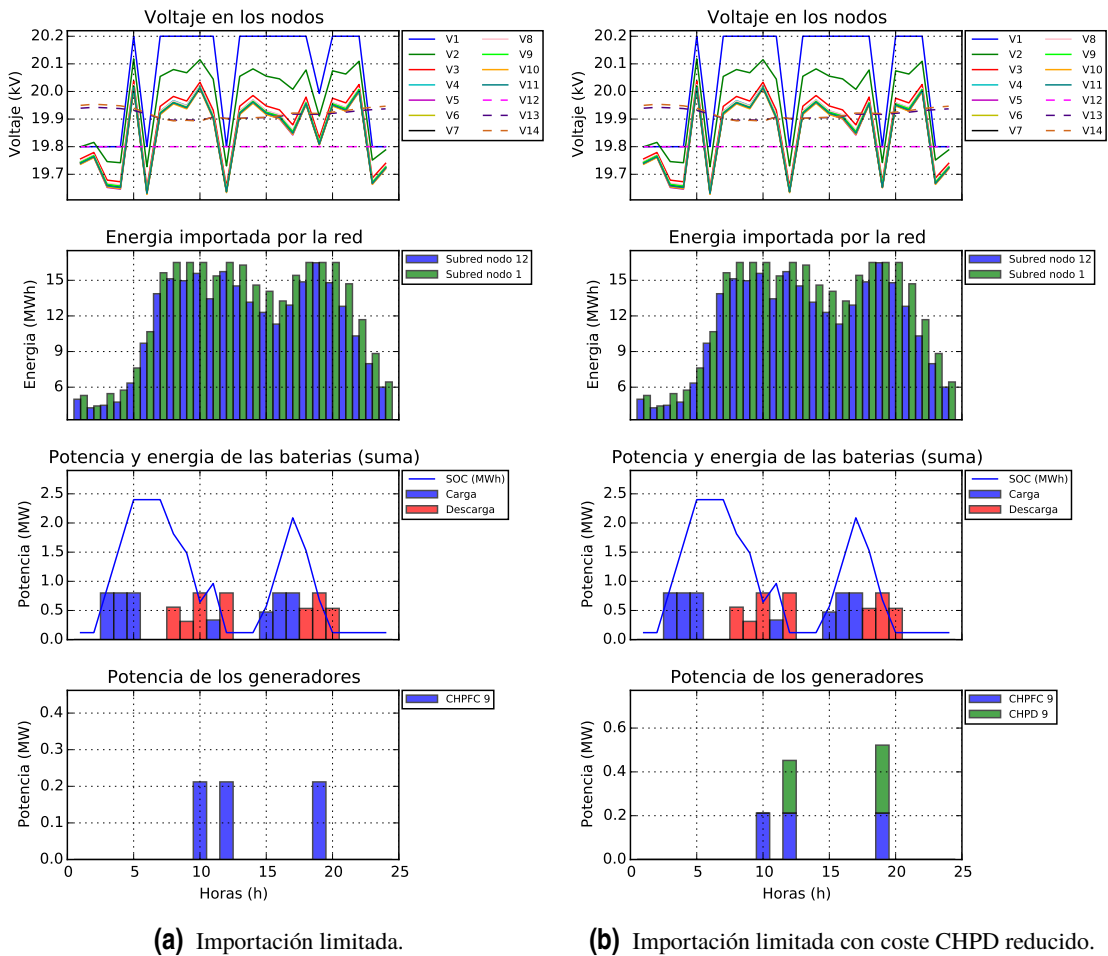


Figura 4.4 Voltajes, potencias y energía con importación limitada.

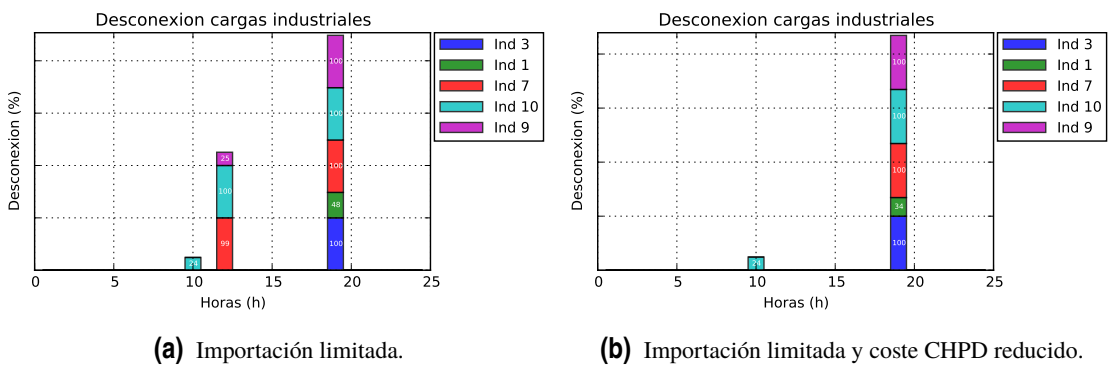


Figura 4.5 Desconexión de la industria con importación limitada.

industria en el nodo 10 y 35% de la del 9, el descenso de consumo es de 0,223 MWh.

$$\Delta E_{ind} = 0,83 \cdot (0,57375 \cdot 0,35 + 0,068 \cdot 0,99) = 0,223 MWh$$

Esto supone un coste de 44,51 € desconectando las industrias. Si esa energía se produjese con el generador diésel, el coste sería de 53,45 €:

$$C_{CHPD} = 150 \cdot 0,223 + 20 = 53,45 MWh$$

En la figura 4.4b y 4.5b se muestran los resultados obtenidos si el generador diésel tuviese un coste fijo de 11 €/h en lugar de 20 €/h. La totalidad de la potencia sería producida por el generador CHPD en la hora 11 y se reduciría la desconexión de la industria en la hora 19.

4.3 Limitación de Potencia Importada y Coste de Desconexión Reducido

Con el fin de estudiar el correcto funcionamiento de la desconexión de la industria, se han mantenido las restricciones y los valores de los parámetros del caso anterior (red limitada a 16,5 MW, costes de los generadores del caso base y reservas a cero). Sin embargo, en este caso se ha reducido el coste de desconexión de la industria a 100 €/MWh, menos que el coste de los generadores.

La energía total importada por la red es exactamente la misma a la del caso anterior, aunque los costes ahora son más reducidos gracias al menor coste de desconexión **44413,39 €**

Puede apreciarse en la figura 4.7 que vuelven a desconectarse las industrias durante los periodos de máximo consumo de energía, en este caso en mayor medida que en el anterior (existe desconexión también en la hora 10). La potencia que antes era suministrada por los generadores se consigue ahora mediante la desconexión de las industrias.

El comportamiento de las baterías sigue la misma lógica que en el caso anterior, ya que, desde su punto de vista, el funcionamiento de la red es exactamente el mismo (en lugar de producir más con generadores, se reduce la demanda con la industria).

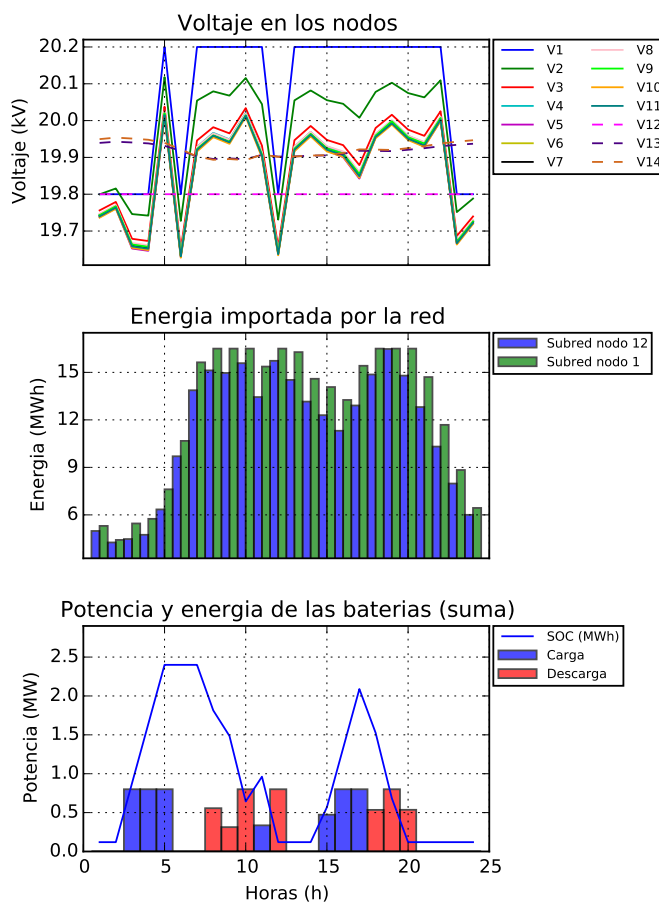


Figura 4.6 Voltajes, potencias y energía con coste de desconexión reducido.

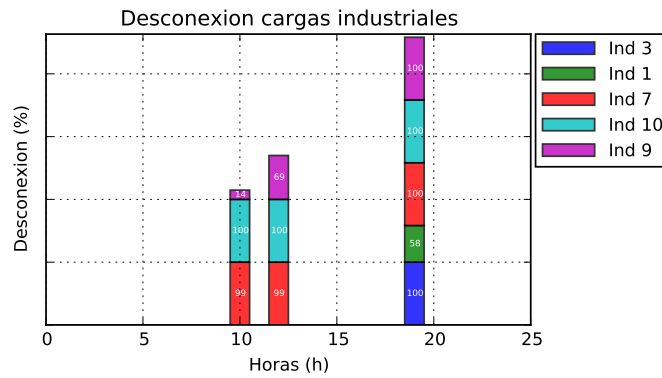


Figura 4.7 Desconexión de las industrias con coste de desconexión reducido.

4.4 Alta Generación Renovable

Para este caso se ha aumentado en gran medida la capacidad máxima de los generadores renovables, de forma que en algunos momentos del día la red produzca más de lo que consume, por lo que pasa a exportar energía. Se han multiplicado por 10 las capacidades de los sistemas fotovoltaicos y por 15 la del aerogenerador. Las potencias producida por los generadores renovables en este caso son las siguientes:

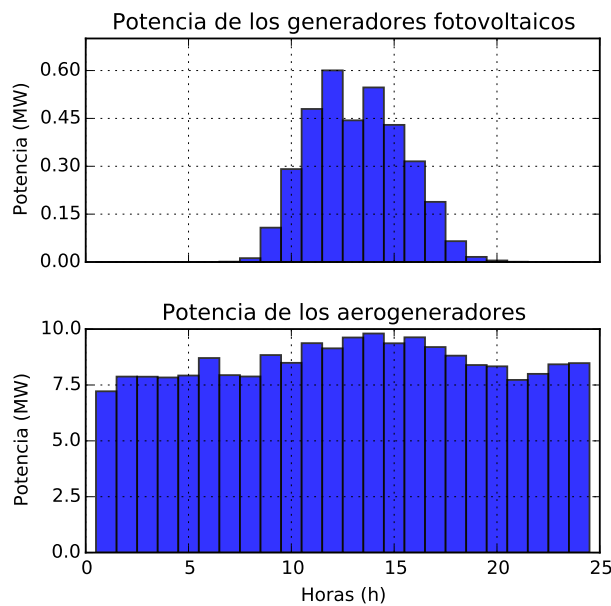


Figura 4.8 Producción de los generadores renovables aumentada.

El coste total de la red es de **30062,33 €** (-32,2% respecto al caso base) con una energía total importada por la red de **385,35 MWh** (reducción del 32,8%).

Puede verse en la figura 4.9 que la curva de energía importada por la subred 1 se ha desplazado verticalmente hacia abajo con respecto a la del caso base, ya que el perfil de generación de los generadores renovables y de las cargas se ha mantenido. De hecho, durante los periodos de menor consumo (horas 1 a 5 y hora 24) la subred 1 pasa de importar a exportar energía, lo que permite vender la energía durante esas horas. Es precisamente en las horas de menor coste que parte de esa energía es utilizada para cargar las baterías. El perfil de carga de estas es muy similar al caso base,

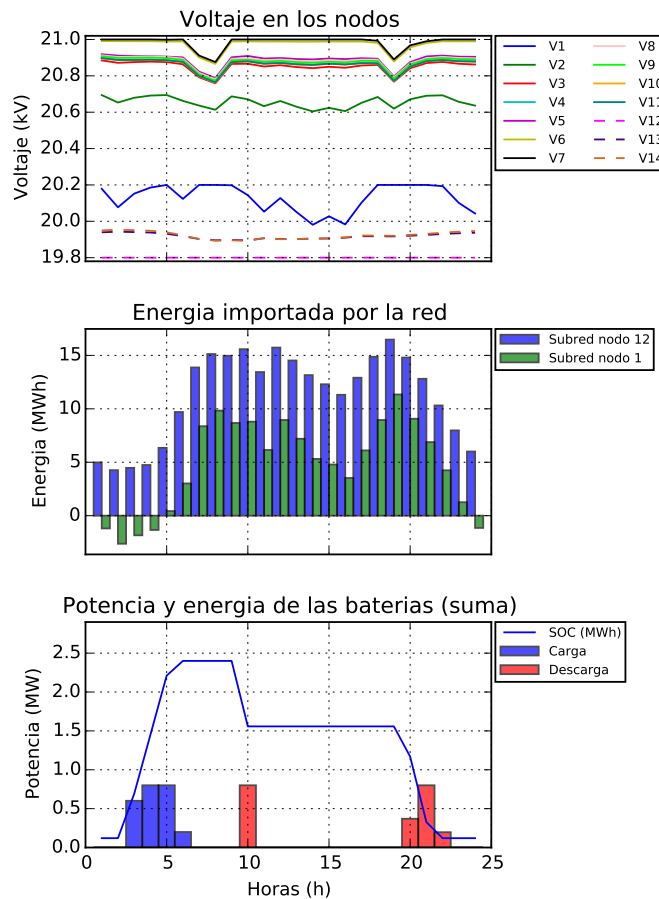


Figura 4.9 Voltajes, potencias y energía con alta generación de renovables.

descargándose en las horas de mayor coste. El consumo industrial se mantiene en todo momento a su nivel máximo.

Al contrario que los otros casos, el voltaje de los primeros nodos de la subred 1 siempre es menor que en los nodos intermedios ya que la producción renovable tiene lugar en estos últimos. Disminuye aún en mayor medida cuando existe una mayor generación en los nodos intermedios, es decir, cuando la producción de energía fotovoltaica se suma a la eólica en las horas centrales del día. La tensión en los nodos de la subred 2 no varía ya que esta red no cuenta con generadores renovables.

4.5 Restricción de Reservas

Este último caso es el más complicado pues supone añadir las restricciones de reserva de potencia para control de frecuencia primario y realizar sucesivas iteraciones resolviendo el problema matemático para poder considerar la energía disponible en las baterías (ver apartado 3.2.2). Se van a mantener los valores de los parámetros del caso base y se va a establecer una reserva de 620 kW tanto para subida como para bajada de potencia.

El coste total de la red es de **44442,98 €** con una energía total importada por la red de **574,05 MWh**. Estos valores no divergen prácticamente nada de los del caso base (+0,2% para el coste) porque los costes derivados de mantener algunos generadores encendidos es muy limitado.

Tal y como se aprecia en las figuras 4.10 y ??, las baterías son las que proveen prácticamente en su totalidad las reservas de subida y bajada, debido a que tienen un menor coste que los generadores. Se limita la carga a alrededor de 200 kW de forma que siempre exista un margen de aumento de

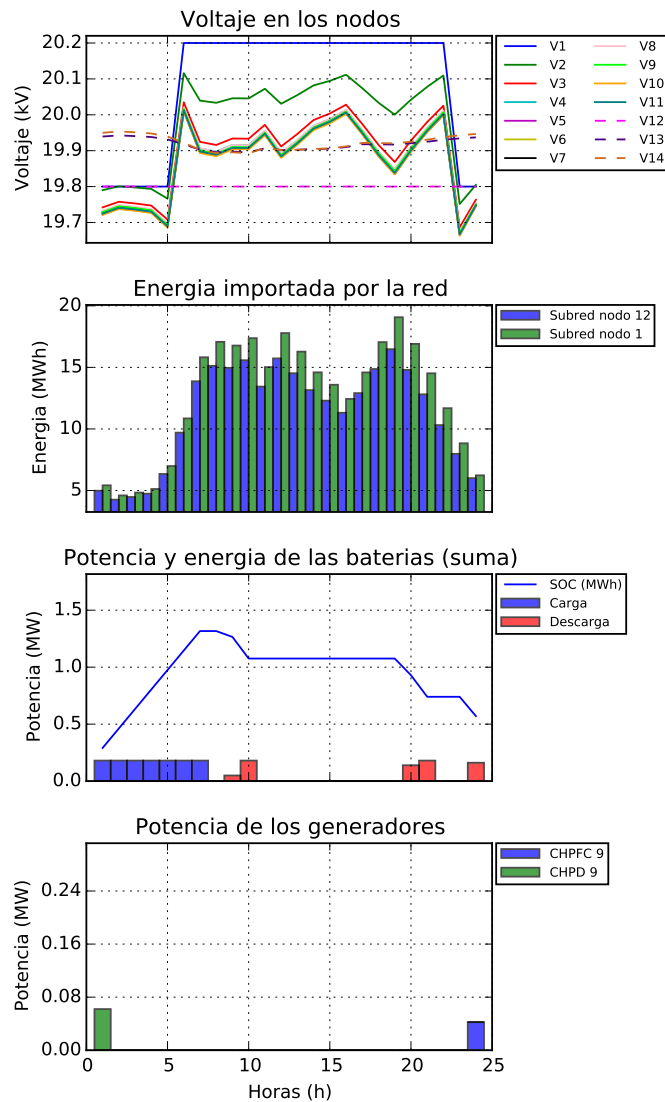


Figura 4.10 Voltajes, potencias y energía con restricciones de reserva.

620 kW hasta la potencia máxima de carga conjunta (800 kW) para asegurar la reserva de bajada (si fuese necesaria, las baterías pueden aumentar el consumo de la red en ese margen). Ocurre lo mismo para la reserva de subida, en los momentos en los que las baterías descargan potencia a la red esta es limitada a un valor cercano a 200 kW.

En la primera hora es necesario encender el generador diésel para hacer frente a la reserva de subida: pese a que la batería del nodo 5 tiene capacidad para aumentar su potencia de descarga en la red en 696 kW (600 kW de potencia máxima y 96 kW que consume de la red para cargarse), tan solo cuenta con 180 kWh almacenados, por lo que nunca sería capaz de producir la potencia necesaria durante 10 minutos (solo podría descargar 90 kWh debido a que debe mantener un mínimo estado de carga de 90 kWh). Lo mismo ocurre en la última hora para la batería del nodo 10: su estado de carga está al mínimo, por lo que es necesario encender el generador CHPFC para asegurar la reserva de subida. En este caso se necesita menos margen (la batería del nodo 10 tiene menor potencia que la del 5) y este se puede proveer con un menor coste gracias al CHPFC. En ambos casos los generadores están produciendo a su mínima potencia.

Con respecto al voltaje en los nodos, se mantiene la lógica explicada anteriormente. Al comparar estos valores con los del caso base, se aprecia que la curva de voltaje en los nodos intermedios sigue

de una forma más exacta la curva de consumo (o de importación de energía), ya que en este caso la carga o descarga de las baterías es mucho más limitada y no altera mucho el perfil de voltaje.

4.6 Resultados Estocásticos

El objetivo del modelo estocástico es resolver el problema de Asignación de Unidades. Se pretende determinar con antelación qué generadores será necesario poner en marcha teniendo en cuenta la incertidumbre asociada a los precios de los mercados eléctricos y a las fuentes de energías renovables.

Se ha establecido un límite a la potencia importada por la red de 16,5 MW, igual que en el caso 4.2, para poder estudiar la puesta en marcha de los generadores y se ha mantenido el resto de parámetros con los valores del caso base. Los parámetros estocásticos están representados por los escenarios especificados en la sección 3.5. Como se ha explicado en el apartado 3.5.4, no se ha tenido acceso a ningún solver con capacidad para resolver el problema MINLP, ya que el entorno PySP no permite acceder remotamente a la interfaz NEOS. Por lo tanto, se ha resuelto el problema relajando las variables binarias de estado de los generadores y posteriormente se ha aproximado el valor de estas variables a 0 ó 1 (apagado/encendido). Los valores de estas variables son calculados a partir de los resultados de cada escenario haciendo una media ponderada con las probabilidades asociadas a cada uno. El resultado se muestra en la figura 4.11.

El coste máximo para el caso estocástico se ha dado en el escenario 134 (escenario 1 de precios, 3 de generación fotovoltaica y 4 de generación eólica). Este escenario tiene el mayor coste promedio de electricidad (85,9 €/MWh) y las menores producciones medias renovables (0,042 y 0,088 para PV y eólica respectivamente). Esto supone un coste total de **51941,5 €**. El coste mínimo de **22499,44 €** se da en el escenario 241, que representa las condiciones opuestas con el menor precio medio y la mayor generación renovable. El coste medio ponderado con la probabilidad de ocurrencia de cada escenario es de **44626,49 €**.

Hay que considerar, tal y como se explica en [52], que la aproximación realizada puede implicar importantes desviaciones en cuanto a la optimalidad de la solución. La potencia que se estaba considerando para los generadores con un valor de $u_{g,t}$ que haya sido aproximado a 0 deberá ser producida por otros medios. En este caso, al estar la red limitada, tendría que aumentarse la producción de los generadores encendidos, pero si estos ya hubieran alcanzado su límite, sería necesario encender otros, por lo que cambiaría la solución a la que se ha llegado. Por lo tanto, se plantea necesario utilizar un solver capaz de resolver problemas MINLP para poder conseguir una solución más exacta en el problema estocástico.

A pesar de ello, la solución se asemeja en gran medida a la presentada en el caso determinista equivalente: el coste difiere en menos de 6 € y el estado de los generadores es muy similar (ver figura 4.4a). Por lo tanto, este modelo podría ser utilizado para anticipar los generadores que deberán ser puestos en marcha a pesar de que los costes finales difieran en mayor o menor medida.

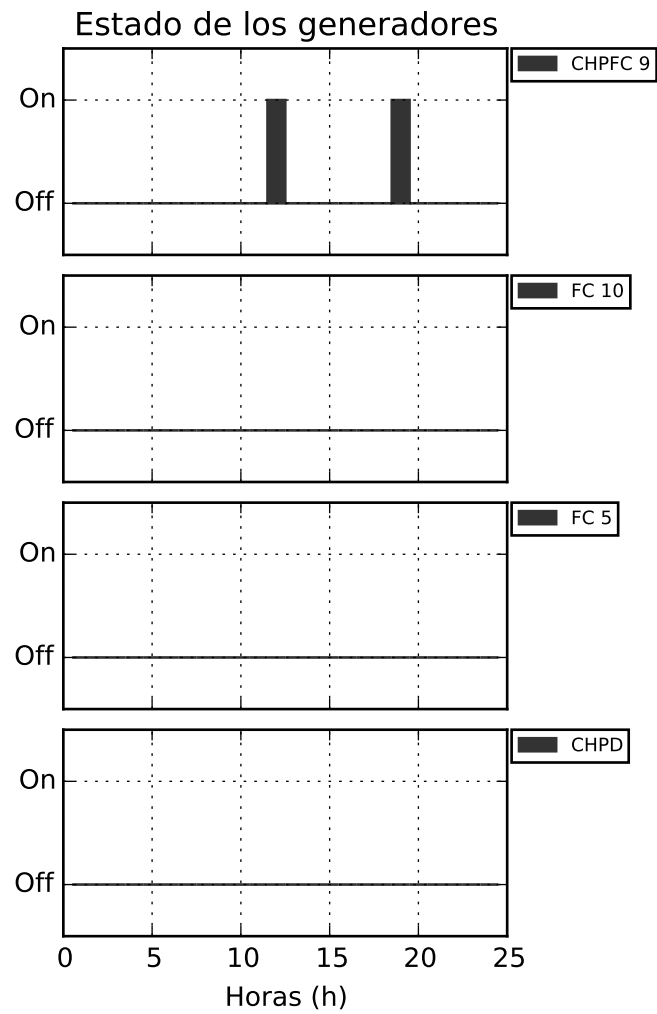


Figura 4.11 Estado de los generadores en el caso estocástico.

5 Conclusiones y Líneas Futuras

Este proyecto ha tenido como resultado final la modelización de una red de media tensión de referencia en Europa y la creación de un programa de optimización matemática en Pyomo. Se entregan junto a esta memoria los archivos necesarios para la utilización del modelo determinista (archivo *.py* con el modelo, archivo Excel para introducción de datos y un archivo *.py* utilizado para crear las gráficas a partir de los resultados del modelo) y los archivos del modelo estocástico (archivo *.py* con el modelo de referencia, archivos *.dat* que definen el árbol de escenarios y los archivos Excel con los valores de los parámetros estocásticos en cada escenario). Todos los archivos cuentan con comentarios para facilitar su comprensión y se incluye también un archivo de texto con instrucciones para el uso y modificación de los modelos.

El modelo creado sirve para determinar los valores óptimos de generación e importación de energía en la red para un periodo de 24 horas de forma que se mantengan las condiciones operacionales deseadas y se reduzcan los costes. Se han presentado distintos casos para estudiar las funcionalidades del modelo con resultados positivos. Es posible establecer restricciones de reserva de potencia para el control de frecuencia primario y se ha estudiado el comportamiento de los generadores y de desconexión de la industria mediante casos en los que la red tiene una capacidad de importación limitada y diversos costes. Se ha analizado igualmente un caso con alta generación de energías renovables en el que la red pasa a exportar energía por algunas horas.

En todos estos casos, el modelo ha encontrado una solución óptima del problema y los valores hallados coinciden con el comportamiento esperado.

En el caso estocástico, pese a que se ha tenido que aproximar la solución del problema relajado, los resultados son muy similares a los del caso determinista equivalente, por lo que podría considerarse este modelo para el problema de Asignación de Unidades.

A continuación se especifican algunos factores o funcionalidades adicionales que podrían incluirse en el modelo como pasos a seguir del proyecto:

- Conectar las subredes a través de un enlace entre los nodos 8 y 14 para estudiar configuraciones de la red o el comportamiento de diferentes enlaces.
- Incluir la presencia de vehículos eléctricos que representen el comportamiento de cargas adicionales pero también tengan funcionalidades para actuar como baterías (*vehicle-to-grid*).
- Aumentar la versatilidad de desconexión de las cargas para estudiar en mayor medida la gestión de la demanda (*demand-side management*) y sus efectos en la red.
- Incluir la desconexión de las cargas reservas de potencia para el control primario de frecuencia.
- Considerar la alimentación de los sistemas basados en pilas de combustible para modelizar su comportamiento de una manera más fidedigna.
- Incluir costes de puesta en marcha y aumento o bajada de potencia en los generadores.

- Utilizar un solver adecuado para la resolución del problema MINLP estocástico.

Apéndice A

Modelo Determinista

El modelo creado en Python para el modelo determinista está incluido en un archivo denominado *Determinista.py*:

Código A.1 Modelo determinista en Pyomo.

```
from __future__ import division
from pyomo.environ import *
from pyomo.opt import SolverFactory, IOptSolver
from math import pi, sqrt
import csv
import matplotlib
import matplotlib.pyplot as plt
import pickle
import numpy as np

# MDOELO ABSTRACTO
m = AbstractModel()

#=====
#                                     DATOS DE ENTRADA
#=====

# Numero de horas
m.h = Param()

# Numero de nodos
m.n = Param()

#=====
#                                     INDICES
#=====
```

```

# Indice t de horas
m.T = RangeSet(1, m.h)

# Indice i de los nodos
m.I = RangeSet(1, m.n)

# Set (i,j) que contiene los arcos
m.ARCS = Set(within=m.I*m.I)

# Set (i,-) que contiene todas las parejas nodo-generador, nodo-pv, etc
m.G = Set(dimen=2)
m.NI = Set(dimen=2)
m.PV = Set(dimen=2)
m.WIND = Set(dimen=2)
m.IND = Set(dimen=2)
m.BS = Set(dimen=2)

#=====
#                                     PARAMETROS
#=====

# ESTOCASTICOS

# Precios estocasticos
m.price = Param(m.T, within=NonNegativeReals)

# Factor de producción solar (igual para todos los PV)
m.f_pv = Param(m.T, within=PercentFraction)

# Factor de producción de aerogeneradores
m.f_w = Param(m.T, within=PercentFraction)

# DETERMINISTAS

# Costes (generador y load-shedding)
m.CosteA_g = Param(m.G, within=NonNegativeReals)
m.CosteB_g = Param(m.G, within=NonNegativeReals)
m.C_ind = Param(m.IND, within=NonNegativeReals)

# Limites de generadores
m.Pmin_g = Param(m.G)
m.Pmax_g = Param(m.G, within=NonNegativeReals)
m.Qmin_g = Param(m.G)
m.Qmax_g = Param(m.G, within=NonNegativeReals)
m.Smax_g = Param(m.G, within=NonNegativeReals)

# Renovables

```

```

m.Pmax_pv = Param(m.PV, within=NonNegativeReals)
m.Smax_pv = Param(m.PV, within=NonNegativeReals)
m.Pmax_w = Param(m.WIND, within=NonNegativeReals)
m.Smax_w = Param(m.WIND, within=NonNegativeReals)

def p_pv_init(m, i, pv, t):
    return m.f_pv[t]*m.Pmax_pv[i,pv]
m.p_pv = Param(m.PV, m.T, within=NonNegativeReals, initialize=p_pv_init)

def p_wind_init(m, i, w, t):
    return m.f_w[t]*m.Pmax_w[i,w]
m.p_wind = Param(m.WIND, m.T, within=NonNegativeReals, initialize=
    p_wind_init)

# Cargas esenciales
m.f_house = Param(m.T, within=NonNegativeReals)
m.Pmax_ni = Param(m.NI, within=NonNegativeReals)
m.phi_ni = Param(m.NI, within=NonNegativeReals)

def p_house_init(m, i, ni, t):
    return m.f_house[t]*m.Pmax_ni[i,ni]
m.p_house = Param(m.NI, m.T, within=NonNegativeReals, initialize=
    p_house_init)

def q_house_init(m, i, ni, t):
    return m.p_house[i,ni,t]*tan(m.phi_ni[i,ni])
m.q_house = Param(m.NI, m.T, within=NonNegativeReals, initialize=
    q_house_init)

# Cargas no esenciales
m.f_ind = Param(m.T, within=NonNegativeReals)
m.Pmax_ind = Param(m.IND, within=NonNegativeReals)
m.phi_ind = Param(m.IND, within=NonNegativeReals)

def pe_ind_init(m, i, ind, t):
    return m.f_ind[t]*m.Pmax_ind[i,ind]
m.pe_ind = Param(m.IND, m.T, within=NonNegativeReals, initialize=
    pe_ind_init)

# Parametros de baterias
m.Smax_bs = Param(m.BS, within=NonNegativeReals)
m.Pmax_bs = Param(m.BS, within=NonNegativeReals)
m.rend_c = Param(m.BS, within=PercentFraction)
m.rend_d = Param(m.BS, within=PercentFraction)
m.soc_min = Param(m.BS, within=NonNegativeReals)
m.soc_max = Param(m.BS, within=NonNegativeReals)
m.soc_init = Param(m.BS, within=NonNegativeReals)

```

```

# Parametros de red
m.G_ij = Param(m.I*m.I)
m.B_ij = Param(m.I*m.I)
m.Smax_ij = Param(m.T, within=NonNegativeReals)

# Voltajes minimos y maximos en la red
m.V = Param()
m.Vmin = Param(initialize=m.V*0.95)
m.Vmax = Param(initialize=m.V*1.05)
m.Vmin_pcc = Param(initialize=m.V*0.99)
m.Vmax_pcc = Param(initialize=m.V*1.01)

# Parametros de las restricciones auxiliares
m.f_Q = Param(m.T, within=PercentFraction)
m.ur = Param(m.T)
m.dr = Param(m.T)

#=====
#                               VARIABLES
#=====

# Generadores
m.p_g = Var(m.G, m.T, initialize=5)
m.q_g = Var(m.G, m.T, initialize=5)
m.u_g = Var(m.G, m.T, within=Binary, initialize=1)

# Reactiva en renovables
m.q_pv = Var(m.PV, m.T, initialize=0)
m.q_wind = Var(m.WIND, m.T, initialize=0)

# Load shedding
m.p_ind = Var(m.IND, m.T, within=NonNegativeReals, initialize=1)
m.q_ind = Var(m.IND, m.T, within=NonNegativeReals, initialize=1)
m.ls = Var(m.IND, m.T, within=PercentFraction, initialize=1)

# Baterias
m.pd_bs = Var(m.BS, m.T, within=NonNegativeReals, initialize=0.5)
m.pc_bs = Var(m.BS, m.T, within=NonNegativeReals, initialize=0.5)
m.q_bs = Var(m.BS, m.T, initialize=0.5)
m.soc = Var(m.BS, m.T, within=NonNegativeReals, initialize=0.1)

# Red
m.V_i = Var(m.I, m.T, within=NonNegativeReals, initialize=20)
m.theta_i = Var(m.I, m.T, bounds=(-pi,pi), initialize=2)
m.s_ij = Var(m.ARCS, m.T, initialize=0)

```

```

#=====
#                               FUNCION OBJETIVO
#=====

def obj_expression(m):
    return (sum(m.price[t]
        *(m.p_g[1,"Red",t] + m.p_g[12,"Red",t])
        + sum(m.u_g[i,g,t]*(m.CosteB_g[i,g]*m.p_g[i,g,t]+m.CosteA_g[
            ,g]) for (i,g) in m.G if g != "Red")
        + sum(m.C_ind[i,ind]
            *(m.pe_ind[i,ind,t]*(1 - m.ls[i,ind,t]))) for (i,ind) in m.IND
        )
        for t in m.T))

m.OBJ = Objective(rule=obj_expression)

#=====
#                               RESTRICCIONES
#=====

# FLUJOS DE POTENCIA

# Balance de flujos de activa
def actflux_const_rule(m, i, t):
    return (sum(m.p_g[i,g,t] for (j,g) in m.G if j==i)
        + sum(m.p_pv[i,pv,t] for (j,pv) in m.PV if j==i)
        + sum(m.p_wind[i,wind,t] for (j,wind) in m.WIND if j==i)
        + sum((m.pd_bs[i,bs,t] - m.pc_bs[i,bs,t]) for (j,bs) in m.BS
            if j==i)
        - sum(m.p_house[i,ni,t] for (j,ni) in m.NI if j==i)
        - sum(m.p_ind[i,ind,t] for (j,ind) in m.IND if j==i)
        == m.V_i[i,t]*m.V_i[i,t]*(m.G_ij[i,i]*cos(0) + m.B_ij[i,i]*
            sin(0))
        + sum(m.V_i[i,t]*m.V_i[j,t]
            *(m.G_ij[i,j]*cos(m.theta_i[i,t]-m.theta_i[j,t])
            + m.B_ij[i,j]*sin(m.theta_i[i,t]-m.theta_i[j,t])) for j in m.
            I if (i,j) in m.ARCS))

m.actflux_const = Constraint(m.I, m.T, rule=actflux_const_rule)

# Balance de flujos de reactiva
def reactflux_const_rule(m, i, t):
    return (sum(m.q_g[i,g,t] for (j,g) in m.G if j==i)
        + sum(m.q_pv[i,pv,t] for (j,pv) in m.PV if j==i)
        + sum(m.q_wind[i,w,t] for (j,w) in m.WIND if j==i)
        + sum(m.q_bs[i,bs,t] for (j,bs) in m.BS if j==i)

```

```

- sum(m.q_house[i,ni,t] for (j,ni) in m.NI if j==i)
- sum(m.q_ind[i,ind,t] for (j,ind) in m.IND if j==i)
== m.V_i[i,t]*m.V_i[i,t]*(m.G_ij[i,i]*sin(0) - m.B_ij[i,i]*
  cos(0))
+ sum(m.V_i[i,t]*m.V_i[j,t]
*(m.G_ij[i,j]*sin(m.theta_i[i,t]-m.theta_i[j,t])
- m.B_ij[i,j]*cos(m.theta_i[i,t]-m.theta_i[j,t])) for j in m.
  I if (i,j) in m.ARCS))

m.reactflux_const = Constraint(m.I, m.T, rule=reactflux_const_rule)

## Calculo de potencia aparente en las lineas (aumenta el tiempo de
  resolucion del problema)
#def Sij_const_rule(m, i, j, t):
#   A = (sqrt(3)/m.V)**2
#
#   return ((m.s_ij[i,j,t]**2)*A == (m.G_ij[i,j]**2 + m.B_ij[i,j]**2)
#     *((m.V_i[i,t]*cos(m.theta_i[i,t]) - m.V_i[j,t]*cos(m.theta_i[
#       j,t]))**2
#     +(m.V_i[i,t]*sin(m.theta_i[i,t]) - m.V_i[j,t]*sin(m.theta_i[
#       j,t]))**2))
#
#m.Sij_const = Constraint(m.ARCS, m.T, rule=Sij_const_rule)

# LIMITES EN LA RED

# Limites de voltaje en nodos
def Vbounds_const_rule(m, i, t):
  return m.Vmin <= m.V_i[i,t] <= m.Vmax

m.Vbounds_const = Constraint(m.I, m.T, rule=Vbounds_const_rule)

## Limite de aparente en arcos
#def Sijbounds_const_rule(m, i, j, t):
#   return m.s_ij[i,j,t] <= m.Smax_ij[t]
#
#m.Sijbounds_const = Constraint(m.ARCS, m.T, rule=Sijbounds_const_rule)

# Mantener voltaje en nodos PCC
def Vpcc1_const_rule(m, t):
  return m.Vmin_pcc <= m.V_i[1,t] <= m.Vmax_pcc
m.Vpcc1_const = Constraint(m.T, rule=Vpcc1_const_rule)

def Vpcc12_const_rule(m, t):
  return m.Vmin_pcc <= m.V_i[12,t] <= m.Vmax_pcc
m.Vpcc12_const = Constraint(m.T, rule=Vpcc12_const_rule)

```



```

# Fijar el angulo en nodos PCC a cero
def Theta1_const_rule(m, t):
    return m.theta_i[1,t] == 0
m.Theta1_const = Constraint(m.T, rule=Theta1_const_rule)

def Theta12_const_rule(m, t):
    return m.theta_i[12,t] == 0
m.Theta12_const = Constraint(m.T, rule=Theta12_const_rule)

# DESCONEXION DE LA INDUSTRIA

# Desconexion de potencia activa
def actls_const_rule(m, i, ind, t):
    return m.p_ind[i,ind,t] == m.ls[i,ind,t]*m.pe_ind[i,ind,t]
m.actls_const = Constraint(m.IND, m.T, rule=actls_const_rule)

# Calculo de potencia reactiva
def reactls_const_rule(m, i, ind, t):
    return m.q_ind[i,ind,t] == m.p_ind[i,ind,t]*tan(m.phi_ind[i,ind])
m.reactls_const = Constraint(m.IND, m.T, rule=reactls_const_rule)

# GENERADORES

# Limite min de activa en generadores
def Pgenlowbound_const_rule(m, i, g, t):
    return m.u_g[i,g,t]*m.Pmin_g[i,g] <= m.p_g[i,g,t]
m.Pgenlowbound_const = Constraint(m.G, m.T, rule=Pgenlowbound_const_rule
)

#Limite max de activa en generadores
def Pgenupbound_const_rule(m, i, g, t):
    return m.p_g[i,g,t] <= m.u_g[i,g,t]*m.Pmax_g[i,g]
m.Pgenupbound_const = Constraint(m.G, m.T, rule=Pgenupbound_const_rule)

# Limite min de reactiva en generadores
def Qgenlowbound_const_rule(m, i, g, t):
    return m.u_g[i,g,t]*m.Qmin_g[i,g] <= m.q_g[i,g,t]

```

```

m.Qgenlowbound_const = Constraint(m.G, m.T, rule=Qgenlowbound_const_rule
)

# Limite max de reactiva en generadores
def Qgenupbound_const_rule(m, i, g, t):
    if (g != "FC" and g != "CHPFC"):
        return m.q_g[i,g,t] <= m.u_g[i,g,t]*m.Qmax_g[i,g]
    else:
        return Constraint.Skip

m.Qgenupbound_const = Constraint(m.G, m.T, rule=Qgenupbound_const_rule)

# Restricción de reactiva segun potencia aparente para generadores con
inversor
def FCreateact_const_rule(m, i, g, t):
    if (g == "FC" or g == "CHPFC"):
        return m.q_g[i,g,t]**2 <= (m.Smax_g[i,g]**2 - m.p_g[i,g,t]**2)*m.
            u_g[i,g,t]
    else:
        return Constraint.Skip

m.FCreateact_const = Constraint(m.G, m.T, rule=FCreateact_const_rule)

# RENOVABLES

# Reactiva en PV
def PVreact_const_rule(m, i, pv, t):
    return m.q_pv[i,pv,t]**2 <= m.Smax_pv[i,pv]**2 - m.p_pv[i,pv,t]**2

m.PVreact_const = Constraint(m.PV, m.T, rule=PVreact_const_rule)

# Reactiva en eolica
def Wreact_const_rule(m, i, w, t):
    return m.q_wind[i,w,t]**2 <= m.Smax_w[i,w]**2 - m.p_wind[i,w,t]**2

m.Wreact_const = Constraint(m.WIND, m.T, rule=Wreact_const_rule)

# BATERIAS

# Balance en las baterias
def SOC_const_rule(m, i, bs, t):
    if (t == 1):
        soc_0 = m.soc_init[i,bs]
    else:
        soc_0 = m.soc[i,bs,t-1]

```

```

    return (m.soc[i,bs,t] == soc_0
            + m.pc_bs[i,bs,t]*m.rend_c[i,bs]
            - m.pd_bs[i,bs,t]/m.rend_d[i,bs])

m.SOC_const = Constraint(m.BS, m.T, rule=SOC_const_rule)

# Limites de aparente en baterias
def Sbsbounds_const_rule(m, i, bs, t):
    return ((m.pd_bs[i,bs,t] - m.pc_bs[i,bs,t])**2
            + m.q_bs[i,bs,t]**2
            <= m.Smax_bs[i,bs]**2)

m.Sbsbounds_const = Constraint(m.BS, m.T, rule=Sbsbounds_const_rule)

# Limites de las baterias
def SOCbounds_const_rule(m, i, bs, t):
    return (m.soc_min[i,bs] <= m.soc[i,bs,t] <= m.soc_max[i,bs])

m.SOCbounds_const_rule = Constraint(m.BS, m.T, rule=SOCbounds_const_rule
)

def PmaxDbs_const_rule(m, i, bs, t):
    return (m.pd_bs[i,bs,t] <= m.Pmax_bs[i,bs])

m.PmaxDbs_const = Constraint(m.BS, m.T, rule=PmaxDbs_const_rule)

def PmaxCbs_const_rule(m, i, bs, t):
    return (m.pc_bs[i,bs,t] <= m.Pmax_bs[i,bs])

m.PmaxCbs_const = Constraint(m.BS, m.T, rule=PmaxCbs_const_rule)

#=====
#                               RESTRICCIONES AUXILIARES
#=====

# Porcentaje máximo de reactiva
def Qin_const_rule(m, t):
    return ((m.q_g[1,'Red',t] + m.q_g[12,'Red',t])
            <= m.f_Q[t]*(m.p_g[1,'Red',t] + m.p_g[12,'Red',t]))

m.Qin_const = Constraint(m.T, rule=Qin_const_rule)

# Reservas control de frecuencia (inicialmente calculadas sin tener en
  cuenta los niveles de carga de las baterías)
def UpReserve_const_rule(m, t):

```

```

suma_bs = 0
for (i,bs) in m.BS:
    Ur_bs = (m.Pmax_bs[i,bs] - m.pd_bs[i,bs,t] + m.pc_bs[i,bs,t])
    suma_bs += Ur_bs

return (sum(m.u_g[i,g,t]*(m.Pmax_g[i,g] - m.p_g[i,g,t]) for (i,g) in
        m.G if g!='Red')
        + suma_bs >= m.ur[t])

m.UpReserve_const = Constraint(m.T, rule=UpReserve_const_rule)

def DownReserve_const_rule(m, t):
    suma_bs = 0
    for (i,bs) in m.BS:
        Dr_bs = (m.Pmax_bs[i,bs] + m.pd_bs[i,bs,t] - m.pc_bs[i,bs,t])
        suma_bs += Dr_bs

    return (sum(m.u_g[i,g,t]*(m.p_g[i,g,t] - m.Pmin_g[i,g]) for (i,g) in
            m.G if g!='Red')
            + suma_bs >= m.dr[t])

m.DownReserve_const = Constraint(m.T, rule=DownReserve_const_rule)

#=====
#                               INTRODUCCION DE LOS DATOS
#=====

data = DataPortal()

# Carga los parametros de entrada
data.load(filename='Datos.xlsx', range='horas', param='h')
data.load(filename='Datos.xlsx', range='nodos', param='n')

# Carga los indices de la red
data.load(filename='Datos.xlsx', range='arcos', format='set_array', set=
m.ARCS)
data.load(filename='Datos.xlsx', range='G_i', set=m.G)
data.load(filename='Datos.xlsx', range='PV_i', set=m.PV)
data.load(filename='Datos.xlsx', range='W_i', set=m.WIND)
data.load(filename='Datos.xlsx', range='NI_i', set=m.NI)
data.load(filename='Datos.xlsx', range='IND_i', set=m.IND)
data.load(filename='Datos.xlsx', range='BS_i', set=m.BS)

# Carga los parametros estocasticos
data.load(filename='Datos.xlsx', range='estoc', select=('Horas',
Precio_red'), param=m.price)

```

```
data.load(filename='Datos.xlsx', range='estoc', select=('Horas','f_pv'),
          param=m.f_pv)
data.load(filename='Datos.xlsx', range='estoc', select=('Horas','f_w'),
          param=m.f_w)

# Carga los parametros de los generadores
data.load(filename='Datos.xlsx', range='generadores', select=('Nodo','
  Generador','CosteA'), param=m.CosteA_g)
data.load(filename='Datos.xlsx', range='generadores', select=('Nodo','
  Generador','CosteB'), param=m.CosteB_g)
data.load(filename='Datos.xlsx', range='generadores', select=('Nodo','
  Generador','Pmin'), param=m.Pmin_g)
data.load(filename='Datos.xlsx', range='generadores', select=('Nodo','
  Generador','Pmax'), param=m.Pmax_g)
data.load(filename='Datos.xlsx', range='generadores', select=('Nodo','
  Generador','Qmin'), param=m.Qmin_g)
data.load(filename='Datos.xlsx', range='generadores', select=('Nodo','
  Generador','Qmax'), param=m.Qmax_g)
data.load(filename='Datos.xlsx', range='generadores', select=('Nodo','
  Generador','Smax'), param=m.Smax_g)

# Carga los parametros de las renovables
data.load(filename='Datos.xlsx', range='PV', select=('Nodo','PV','Pmax')
          , param=m.Pmax_pv)
data.load(filename='Datos.xlsx', range='PV', select=('Nodo','PV','Smax')
          , param=m.Smax_pv)
data.load(filename='Datos.xlsx', range='wind',select=('Nodo','Aerogen','
  Pmax'), param=m.Pmax_w)
data.load(filename='Datos.xlsx', range='wind',select=('Nodo','Aerogen','
  Smax'), param=m.Smax_w)

# Carga los parametros de las cargas esenciales
data.load(filename='Datos.xlsx', range='estoc', select=('Horas','f_house
  '), param=m.f_house)
data.load(filename='Datos.xlsx', range='cargas_ni', select=('Nodo','
  Carga','Pmax'), param=m.Pmax_ni)
data.load(filename='Datos.xlsx', range='cargas_ni', select=('Nodo','
  Carga','Phi'), param=m.phi_ni)

# Carga los parametros de las cargas no esencial
data.load(filename='Datos.xlsx', range='estoc', select=('Horas','f_ind')
          , param=m.f_ind)
data.load(filename='Datos.xlsx', range='cargas_ind', select=('Nodo','
  Carga','Pmax'), param=m.Pmax_ind)
data.load(filename='Datos.xlsx', range='cargas_ind', select=('Nodo','
  Carga','Phi'), param=m.phi_ind)
data.load(filename='Datos.xlsx', range='cargas_ind', select=('Nodo','
  Carga','Coste'), param=m.C_ind)

# Carga los parametros de las baterias
```

```

data.load(filename='Datos.xlsx', range='baterias', select=('Nodo',
    Bateria', 'SOCinit'), param=m.soc_init)
data.load(filename='Datos.xlsx', range='baterias', select=('Nodo',
    Bateria', 'SOCmin'), param=m.soc_min)
data.load(filename='Datos.xlsx', range='baterias', select=('Nodo',
    Bateria', 'SOCmax'), param=m.soc_max)
data.load(filename='Datos.xlsx', range='baterias', select=('Nodo',
    Bateria', 'Rcarga'), param=m.rend_c)
data.load(filename='Datos.xlsx', range='baterias', select=('Nodo',
    Bateria', 'Rdescarga'), param=m.rend_d)
data.load(filename='Datos.xlsx', range='baterias', select=('Nodo',
    Bateria', 'Smax'), param=m.Smax_bs)
data.load(filename='Datos.xlsx', range='baterias', select=('Nodo',
    Bateria', 'Pmax'), param=m.Pmax_bs)

# Carga los parametros de la red
data.load(filename='Datos.xlsx', range='Gij', format='array', param=m.
    G_ij)
data.load(filename='Datos.xlsx', range='Bij', format='array', param=m.
    B_ij)
data.load(filename='Datos.xlsx', range='Auxiliar', select=('Horas', 'Smax
    '), param=m.Smax_ij)
data.load(filename='Datos.xlsx', range='Vbase', param='V')

# Carga los parametros de los constraints auxiliares
data.load(filename='Datos.xlsx', range='Auxiliar', select=('Horas', 'f_Q
    '), param=m.f_Q)
data.load(filename='Datos.xlsx', range='Auxiliar', select=('Horas', 'Ur')
    , param=m.ur)
data.load(filename='Datos.xlsx', range='Auxiliar', select=('Horas', 'Dr')
    , param=m.dr)

# Se crea el modelo concreto con los datos introducidos
instance = m.create_instance(data)

## Linea de codigo opcional para crear un archivo GAMS con el modelo
concreto
#instance.write('determ.gms')

#=====
#
#                               RESOLUCION
#=====

# Para resolver remotamente a través de la interfaz NEOS
solver_manager = SolverManagerFactory('neos')
results = solver_manager.solve(instance, opt='bonmin')

```

```

results.write()

#=====

# Iteraciones para considerar el nivel de carga de las baterias en las
  reservas
#=====

# Comprobamos si hay restricciones de reserva y si es el caso se
  comprueba si hay casos conflictivos
iterar = False
comprob = False

for t in instance.T:
    if value(instance.ur[t]) > 0 or value(instance.dr[t]) > 0:
        comprob = True
        break

if comprob == True:
    for t in instance.T:
        for (i,bs) in instance.BS:
            Ur_bs = (instance.Pmax_bs[i,bs] - instance.pd_bs[i,bs,t] +
                instance.pc_bs[i,bs,t])
            if value(Ur_bs)/6 >= value(instance.soc[i,bs,t] - instance.
                soc_min[i,bs]):
                iterar = True
                break
            Dr_bs = (instance.Pmax_bs[i,bs] + instance.pd_bs[i,bs,t] -
                instance.pc_bs[i,bs,t])
            if value(Dr_bs)/6 >= value(instance.soc_max[i,bs] - instance.
                soc[i,bs,t]):
                iterar = True
                break
        if iterar == True:
            break

# Se itera añadiendo restricciones en las baterias y horas que no tengan
  el nivel
# de carga adecuado en la solucion anterior
while (iterar == True):

    # Listas para la comprobacion posterior, guardan la batería y la
      hora para las que se añade restriccion
    instance.modif_up = []
    instance.modif_down = []

```

```

def UpReserveAdd_const_rule(instance, t):
    suma_bs = 0
    changed = False
    for (i,bs) in instance.BS:
        Ur_bs = (instance.Pmax_bs[i,bs] - instance.pd_bs[i,bs,t] +
                 instance.pc_bs[i,bs,t])
        if value(Ur_bs)/6 >= value(instance.soc[i,bs,t] - instance.
            soc_min[i,bs]):
            Ur_bs = instance.soc[i,bs,t] - instance.soc_min[i,bs]
            instance.modif_up.append(str(i)+' '+str(t))
            changed = True
        suma_bs += Ur_bs

    # Si no tiene nivel de carga suficiente, se añade la restricción
    # para tenerlo en cuenta
    if changed == True:
        return (sum(instance.u_g[i,g,t]*(instance.Pmax_g[i,g] -
            instance.p_g[i,g,t]) for (i,g) in instance.G if g!='Red')
            + suma_bs >= instance.ur[t])
    else:
        return Constraint.Skip

instance.UpReserveAdd_const = Constraint(instance.T, rule=
    UpReserveAdd_const_rule)

def DownReserveAdd_const_rule(instance, t):
    suma_bs = 0
    changed = False
    for (i,bs) in instance.BS:
        Dr_bs = (instance.Pmax_bs[i,bs] + instance.pd_bs[i,bs,t] -
                 instance.pc_bs[i,bs,t])
        if value(Dr_bs)/6 >= value(instance.soc_max[i,bs] - instance.
            soc[i,bs,t]):
            Dr_bs = instance.soc_max[i,bs] - instance.soc[i,bs,t]
            instance.modif_down.append(str(i)+' '+str(t))
            changed = True
        suma_bs += Dr_bs

    # Si no tiene margen de carga suficiente, se añade la restricción
    # para tenerlo en cuenta
    if changed == True:
        return (sum(instance.u_g[i,g,t]*(instance.p_g[i,g,t] -
            instance.Pmin_g[i,g]) for (i,g) in instance.G if g!='Red')
            + suma_bs >= instance.dr[t])
    else:
        return Constraint.Skip

```



```

instance.DownReserveAdd_const = Constraint(instance.T, rule=
    DownReserveAdd_const_rule)

# Vuelve a resolverse el problema de optimizacion
solver_manager = SolverManagerFactory('neos')
results = solver_manager.solve(instance, opt='bonmin')
results.write()

# Comprobar si en la nueva solución hay algun caso conflictivo que
    no haya sido ya resuelto
iterar = False
comp = ''
for t in instance.T:
    for (i,bs) in instance.BS:
        Ur_bs = (instance.Pmax_bs[i,bs] - instance.pd_bs[i,bs,t] +
            instance.pc_bs[i,bs,t])
        if value(Ur_bs)/6 >= value(instance.soc[i,bs,t] - instance.
            soc_min[i,bs]):
            comp = str(i) + ' ' + str(t)
            if comp not in instance.modif_up:
                iterar = True
                break
        Dr_bs = (instance.Pmax_bs[i,bs] + instance.pd_bs[i,bs,t] -
            instance.pc_bs[i,bs,t])
        if value(Dr_bs)/6 >= value(instance.soc_max[i,bs] - instance.
            soc[i,bs,t]):
            comp = str(i) + ' ' + str(t)
            if comp not in instance.modif_down:
                iterar = True
                break
    if iterar == True:
        break

#=====
#                               GRAFICA 1: Voltajes y Potencias
#=====

# Vector con las horas y los colores
horas = np.arange(1,int(value(instance.h))+1)
colores = ['b','g','r','c','m','y','k','#FFCOCB','#00FF00','#FFA500',
    '#008080','#FF00FF','#4B0082','#D2691E','b','g','r','c','m','y','k',
    '#FFCOCB','#00FF00','#FFA500','#008000','#FF00FF','#4B0082','#D2691E']

fig1 = plt.figure()

```

```

# Comprobar si algún generador está produciendo y guardar nodo y nombre
del generador
prod_i = []
prod_g = []
for (i,g) in instance.G:
    if g != 'Red':
        for t in horas:
            if value(instance.p_g[i,g,t]) > 0.01:
                prod_i.append(i)
                prod_g.append(g)
                break

# Si ninguno produce, no se incluye la gráfica de los generadores
if not prod_i:
    params = {'legend.fontsize': 7,
              'legend.handlelength': 3,
              'figure.figsize': (4, 8)}
    plt.rcParams.update(params)

    fig1, (f1_ax1, f1_ax2, f1_ax3) = plt.subplots(3, sharex=True)

else:
    params = {'legend.fontsize': 7,
              'legend.handlelength': 3,
              'figure.figsize': (4, 10)}
    plt.rcParams.update(params)

    fig1, (f1_ax1, f1_ax2, f1_ax3, f1_ax4) = plt.subplots(4, sharex=True
    )

# Voltaje en los nodos
Vmin = 40
Vmax = 0
count = 0

for i in instance.I:
    # Tan solo hay 7 colores
    if i <= 11:
        volt = []
        #Crea el vector con los valores
        for t in horas:
            volt.append(value(instance.V_i[i,t]))

        # Guarda el Vmax y Vmin para los ejes
        if volt[-1] > Vmax:
            Vmax = volt[-1]
        if volt[-1] < Vmin:
            Vmin = volt[-1]

    f1_ax1.plot(horas, volt, color=colores[count], label='V%i'%i)

```

```

count += 1

if i > 11:
    volt = []
    for t in horas:
        volt.append(value(instance.V_i[i,t]))

        if volt[-1] > Vmax:
            Vmax = volt[-1]
        if volt[-1] < Vmin:
            Vmin = volt[-1]

    f1_ax1.plot(horas, volt, color=colores[count], label='V%i'%i, ls
                ='--')
    count += 1

f1_ax1.set_ylim([Vmin-0.02,Vmax+0.02])
f1_ax1.ticklabel_format(useOffset=False)
f1_ax1.set(ylabel='Voltaje (kV)', title='Voltaje en los nodos')
f1_ax1.grid()
f1_lgd1 = f1_ax1.legend(bbox_to_anchor=(1.01, 1), loc='upper left',
                       borderaxespad=0., ncol=2)

# Potencia de la red
opacity = 0.7
bar_width = 0.5
Pmin = 40
Pmax = 0
count = 0

for (i,g) in instance.G:
    if g == 'Red':
        pg = []
        for t in horas:
            pg.append(value(instance.p_g[i,g,t]))

            if pg[-1] > Pmax:
                Pmax = pg[-1]
            if pg[-1] < Pmin:
                Pmin = pg[-1]

        f1_ax2.bar(horas - bar_width*(1-count)/2 + bar_width*count/2, pg,

                  bar_width, align='center', alpha=opacity, color=colores
                  [count],
                  edgecolor=None, label='Subred nodo %i'%i)

# Se incrementa el contador para posicionar las barras y asignar
colores

```

```

        count += 1

f1_ax2.yaxis.set_major_locator(plt.MaxNLocator(5))
f1_ax2.set_ylim([Pmin-1, Pmax+1])
f1_ax2.set(ylabel='Energia (MWh)', title='Energia importada por la red')
f1_ax2.grid()
f1_lgd2 = f1_ax2.legend(bbox_to_anchor=(1.01, 1), loc='upper left',
                        borderaxespad=0.)

# Potencia y SOC de las baterias
bar_width = 1
Pmin = 40
Pmax = 0
pc = []
pd = []
soc = []

for t in horas:
    pc.append(sum(value(instance.pc_bs[i,bs,t]) for (i,bs) in instance.
                  BS))
    pd.append(sum(value(instance.pd_bs[i,bs,t]) for (i,bs) in instance.
                  BS))
    soc.append(sum(value(instance.soc[i,bs,t]) for (i,bs) in instance.BS
                  ))

    if soc[-1] > Pmax:
        Pmax = soc[-1]

f1_ax3.bar(horas, pc, bar_width, align='center',
           alpha=opacity, color='b', edgecolor=None, label='Carga')
f1_ax3.bar(horas, pd, bar_width, align='center',
           alpha=opacity, color='r', edgecolor=None, label='Descarga')
f1_ax3.plot(horas, soc, label='SOC (MWh)')

f1_ax3.set_ylim([0, Pmax+0.5])
f1_ax3.grid()
f1_lgd3 = f1_ax3.legend(bbox_to_anchor=(1.01, 1), loc='upper left',
                        borderaxespad=0.)

if not prod_i:
    f1_ax3.set(xlabel='Horas (h)', ylabel='Potencia (MW)', title='
        Potencia y energia de las baterias (suma)')
else:
    f1_ax3.set(ylabel='Potencia (MW)', title='Potencia y energia de las
        baterias (suma)')

# Potencia de los generadores
bar_width = 1
Pmax = 0

```

```

# Si alguno produce se dibuja un gráfico de barras solo con los que
    producen
if prod_i:
    pg = [None]*len(prod_i)
    h = [0]*len(horas)

# Se itera en los generadores que producen
for i in range(len(prod_i)):
    pg[i] = []
    for t in horas:
        pg[i].append(value(instance.p_g[prod_i[i],prod_g[i],t]))

# Se guarda la altura de las barras anteriores para usarla de
    base
if i > 0:
    h[t-1] += pg[i-1][t-1]

# Se guarda la altura máxima para los límites de los ejes
if (h[t-1] + pg[i][t-1]) > Pmax:
    Pmax = h[t-1] + pg[i][t-1]

f1_ax4.bar(horas, pg[i], bottom=h, width=bar_width, align='center
',
           color=colores[i], alpha=opacity, edgecolor=None, label
           ='%s %i'%(prod_g[i],prod_i[i]))

f1_ax4.yaxis.set_major_locator(plt.MaxNLocator(5))
f1_ax4.set_ylim([0, Pmax+0.25])
f1_ax4.set(xlabel='Horas (h)', ylabel='Potencia (MW)', title='
Potencia de los generadores')
f1_ax4.grid()
f1_lgd4 = f1_ax4.legend(bbox_to_anchor=(1.01, 1), loc='upper left',
borderaxespad=0.)

fig1.subplots_adjust(hspace=0.3)
plt.setp([a.get_xticklabels() for a in fig1.axes[:-1]], visible=False)

fig1.savefig("potencias.pdf", bbox_inches='tight')
plt.show()

#=====
#
#                               GRAFICA 2: Estado de los generadores
#=====

fig2 = plt.figure()
params = {'legend.fontsize': 7,
          'legend.handlelength': 2,

```

```

        'figure.figsize': (3, 6)}
plt.rcParams.update(params)

fig2, (f2_ax1, f2_ax2, f2_ax3, f2_ax4) = plt.subplots(4, sharex=True)
axes = (f2_ax1, f2_ax2, f2_ax3, f2_ax4)
ax = [f2_ax1, f2_ax2, f2_ax3, f2_ax4]

opacity = 0.8
bar_width = 1
count = 0

# Se itera en los generadores distintos de la red, creando una gráfica
# para cada uno
for (i,g) in instance.G:
    if g != 'Red':
        ug = []
        for t in horas:
            ug.append(value(instance.u_g[i,g,t]))

        ax[count].bar(horas, ug, bar_width, align='center',
            alpha=opacity, color='black', label='%s %i'%(g,i))
        ax[count].yaxis.set_major_locator(plt.MaxNLocator(integer=True))
        ax[count].set_ylim([-0.5, 1.5])
        ax[count].grid()
        ax[count].legend(bbox_to_anchor=(1.01, 1), loc='upper left',
            borderaxespas=0., ncol=2)

        count = count + 1

f2_ax1.set_title('Estado de los generadores')
f2_ax4.set_xlabel('Horas (h)')
fig2.subplots_adjust(hspace=0.1)
plt.setp([a.get_xticklabels() for a in fig2.axes[:-1]], visible=False)
plt.setp(axes, yticks=[0,1], yticklabels=['Off','On'])

fig2.savefig("generadores.pdf", bbox_inches='tight')
plt.show()

#=====
#                               GRAFICA 3: Desconexion de la industria
#=====

fig3 = plt.figure()

params = {'legend.fontsize': 10,
          'legend.handlelength': 3,
          'figure.figsize': (5,3)}
plt.rcParams.update(params)

```

```

fig3, f3_ax = plt.subplots()

# Calcular cuantas se desconectan y guardar los nodos correspondientes
descon = []
for (i,ind) in instance.IND:
    for t in horas:
        if value(instance.ls[i,ind,t]) < 1:
            descon.append(i)
            break

# Si no hay desconexión se crea una gráfica con el consumo industrial al
100 %
if not descon:
    ls = []
    for t in horas:
        ls.append(value(instance.ls[1,'Ind',t])*100)

    f3_ax.plot(horas, ls)

    f3_ax.set_ylim([0, 105])
    f3_ax.set(xlabel='Horas (h)', ylabel='Consumo (%)', title='Consumo
    industrial')
    f3_ax.grid()

# Si hay desconexión se crea un gráfico de barras únicamente con las que
se desconectan
else:
    ls = [None]*len(descon)
    h = [0]*len(horas)
    barwidth = 1
    Dmax = 0

    # Iterar entre las que se desconectan
    for i in range(len(descon)):
        ls[i] = []
        for t in horas:
            ls[i].append((1-value(instance.ls[descon[i],'Ind',t]))*100)

            # Se guarda la altura de las barras anteriores para usarla de
            base
            if i > 0:
                h[t-1] += ls[i-1][t-1]

            # Si hay desconexión se incluye etiqueta con el valor de
            desconexión
            if ls[i][t-1] > 1:
                plt.text(horas[t-1], h[t-1] + ls[i][t-1]/2, '%i'%ls[i][t
                -1],
                    ha="center", va="center", color="white", size =
                    5)

```

```

        # Se guarda la altura máxima para los ejes
        if (h[t-1] + ls[i][t-1]) > Dmax:
            Dmax = h[t-1] + ls[i][t-1]

        f3_ax.bar(horas, ls[i], bottom=h, width=bar_width, align='center
        ',
                alpha=opacity, color=colores[i], edgecolor=None, label='
                Ind %i'%descon[i])

        f3_ax.yaxis.set_major_locator(plt.MaxNLocator(5))
        f3_ax.set_ylim([0, Dmax + 5])
        f3_ax.set_yticklabels([])
        f3_ax.set(xlabel='Horas (h)', ylabel='Desconexion (%)', title='
        Desconexion cargas industriales')
        f3_ax.grid()
        f3_lgd = f3_ax.legend(bbox_to_anchor=(1.01, 1), loc='upper left',
        borderaxespad=0.)

    fig3.savefig("industria.pdf", bbox_inches='tight')
    plt.show()

#=====
#                               GRAFICA 4: Perfiles de las cargas
#=====

fig4 = plt.figure()
params = {'figure.figsize': (5, 5)}
plt.rcParams.update(params)

fig4, (f4_ax1, f4_ax2) = plt.subplots(2, sharex=True)

# Perfil residencial
res = []

for t in horas:
    res.append(value(instance.f_house[t]))

f4_ax1.plot(horas, res)

f4_ax1.set_ylim([0,1])
f4_ax1.ticklabel_format(useOffset=False)
f4_ax1.set(ylabel='Potencia activa [p.u.]', title='Perfil de las cargas
residenciales')
f4_ax1.grid()

```



```

# Perfil industrial
ind = []

for t in horas:
    ind.append(value(instance.f_ind[t]))

f4_ax2.plot(horas, ind)

f4_ax2.set_ylim([0,1])
f4_ax2.set_xlabel='Horas (h)', ylabel='Potencia activa [p.u.]', title='
    Perfil de las cargas industriales')
f4_ax2.grid()

fig4.subplots_adjust(hspace=0.3)
plt.setp([a.get_xticklabels() for a in fig1.axes[:-1]], visible=False)

fig4.savefig("cargas.pdf", bbox_inches='tight')
plt.show()

#=====
#                               GRAFICA 5: Precios de importación
#=====

fig5 = plt.figure()
params = {'figure.figsize': (5, 3)}
plt.rcParams.update(params)

fig5, f5_ax = plt.subplots()

p = []
Pmax = 0
Pmin = 1000

#Crea el vector con los valores
for t in horas:
    p.append(value(instance.price[t]))

    if p[-1] > Pmax:
        Pmax = p[-1]
    if p[-1] < Pmin:
        Pmin = p[-1]

f5_ax.plot(horas, p)

f5_ax.set_ylim([Pmin - 5, Pmax + 5])

```

```

f5_ax.set(xlabel = 'Horas (h)', ylabel='Coste (%s/MWh)%(u"\N{euro sign
    }")', title='Coste de importacion de energia')
f5_ax.grid()

fig5.savefig("precios.pdf", bbox_inches='tight')
plt.show()

#=====
#                               GRAFICA 6: Renovables
#=====

fig6 = plt.figure()
params = {'figure.figsize': (5, 5)}
plt.rcParams.update(params)

fig6, (f6_ax1, f6_ax2) = plt.subplots(2, sharex=True)

# Perfil PV
pv = []

#Crea el vector con los valores
for t in horas:
    pv.append(value(instance.f_pv[t]))

f6_ax1.plot(horas, pv)

f6_ax1.yaxis.set_major_locator(plt.MaxNLocator(5))
f6_ax1.set_ylim([0,0.4])
f6_ax1.ticklabel_format(useOffset=False)
f6_ax1.set(ylabel='Potencia activa [p.u.]', title='Perfil de generacion
    fotovoltaica')
f6_ax1.grid()

w = []
for t in horas:
    w.append(value(instance.f_w[t]))

f6_ax2.plot(horas, w)

f6_ax2.yaxis.set_major_locator(plt.MaxNLocator(5))
f6_ax2.set_ylim([0.2,0.6])
f6_ax2.set(xlabel='Horas (h)', ylabel='Potencia activa [p.u.]', title='
    Perfil de generacion eolica')
f6_ax2.grid()

```

```
fig6.subplots_adjust(hspace=0.3)
plt.setp([a.get_xticklabels() for a in fig6.axes[:-1]], visible=False)

fig6.savefig("renovables.pdf", bbox_inches='tight')
plt.show()

#=====
#           GRAFICA 7: Potencia generadores renovables
#=====

fig7 = plt.figure()

params = {'legend.fontsize': 10,
          'legend.handlelength': 3,
          'figure.figsize': (5,5)}
plt.rcParams.update(params)

fig7, (f7_ax1, f7_ax2) = plt.subplots(2, sharex=True)

aux_pv = []
aux_wind = []
Pmax_pv = 0
Pmax_wind = 0
bar_width = 1

for t in horas:
    suma_pv = 0
    suma_wind = 0

    for (i,pv) in instance.PV:
        suma_pv += value(instance.p_pv[i,pv,t])

    for (i,w) in instance.WIND:
        suma_wind += value(instance.p_wind[i,w,t])

    aux_pv.append(suma_pv)
    aux_wind.append(suma_wind)

    if suma_pv > Pmax_pv:
        Pmax_pv = suma_pv
    if suma_wind > Pmax_wind:
        Pmax_wind = suma_wind

f7_ax1.bar(horas, aux_pv, width=bar_width, align='center', alpha=opacity
, edgecolor=None)

f7_ax1.set_ylim([0, Pmax_pv+0.1])
```

```

f7_ax1.set(ylabel='Potencia (MW)', title='Potencia de los generadores
    fotovoltaicos')
f7_ax1.grid()
f7_ax1.ticklabel_format(useOffset=False)
f7_ax1.yaxis.set_major_locator(plt.MaxNLocator(5))

f7_ax2.bar(horas, aux_wind, width=bar_width, align='center', alpha=
    opacity, edgecolor=None)

f7_ax2.set_ylim([0, Pmax_wind+0.2])
f7_ax2.set_xlabel='Horas (h)', ylabel='Potencia (MW)', title='Potencia
    de los aerogeneradores')
f7_ax2.grid()
f7_ax2.yaxis.set_major_locator(plt.MaxNLocator(5))

fig7.subplots_adjust(hspace=0.3)
plt.setp([a.get_xticklabels() for a in fig7.axes[:-1]], visible=False)

fig7.savefig("renovables_potencias.pdf", bbox_inches='tight')
plt.show()

#=====
#                               MOSTRAR RESULTADOS POR PANTALLA
#=====

results.write()
instance.OBJ.display()

pg = 0
for (i,g) in instance.G:
    if g == 'Red':
        for t in horas:
            pg += (value(instance.p_g[i,g,t]))
print('La energía total importada es %f MWh'%pg)

#=====
#                               GUARDAR LOS RESULTADOS
#=====

# Guarda la solución en results
instance.solutions.store_to(results)

# Archiva los resultados: el archivo es usado por Solucion.py para crear
    las graficas
# sin necesidad de volver a resolver el problema de optimización
file = open("results.pickle", "wb")

```

```

pickle.dump(results, file)
file.close()

# Escribe un archivo CSV con los resultados
with open('results.csv', mode='w') as Result:
    results_writer = csv.writer(Result, delimiter=';', quotechar='"',
        quoting=csv.QUOTE_NONNUMERIC)

    results_writer.writerow(['OBJ', value(instance.OBJ)])

# Variables
for v in instance.component_objects(Var, active=True):
    if v.name == 'p_g' or v.name == 'u_g':
        for (nodo,x) in instance.G:
            for t in instance.T:
                results_writer.writerow([v[nodo,x,t], value(v[nodo,x,t]
                    )])

    elif v.name == 'pd_bs' or v.name == 'pc_bs' or v.name == 'soc':
        for (nodo,x) in instance.BS:
            for t in instance.T:
                results_writer.writerow([v[nodo,x,t], value(v[nodo,x,t]
                    )])

    elif v.name == 'p_ind' or v.name == 'ls':
        for (nodo,x) in instance.IND:
            for t in instance.T:
                results_writer.writerow([v[nodo,x,t], value(v[nodo,x,t]
                    )])

    elif v.name == 's_ij':
        for (i,j) in instance.ARCS:
            for t in instance.T:
                results_writer.writerow([v[i,j,t], value(v[i,j,t])])

    elif v.name == 'V_i':
        for nodo in instance.I:
            for t in instance.T:
                results_writer.writerow([v[nodo,t], value(v[nodo,t])])

    elif v.name == 'theta_i':
        for nodo in instance.I:
            for t in instance.T:
                results_writer.writerow([v[nodo,t], value(v[nodo,t])])

# Parametros
for v in instance.component_objects(Param, active=True):
    if v.name == 'p_house':
        for (nodo,x) in instance.NI:

```

```
        for t in instance.T:
            results_writer.writerow([v.name,nodo,x,t,v[nodo,x,t]])

elif v.name == 'p_pv':
    for (nodo,x) in instance.PV:
        for t in instance.T:
            results_writer.writerow([v.name,nodo,x,t,v[nodo,x,t]])

elif v.name == 'p_wind':
    for (nodo,x) in instance.WIND:
        for t in instance.T:
            results_writer.writerow([v.name,nodo,x,t,v[nodo,x,t]])

elif v.name == 'price':
    for t in instance.T:
        results_writer.writerow([v.name,t,v[t]])
```

Apéndice B

Estructura del problema estocástico

La estructura del modelo estocástico se especifica mediante un archivo de text denominado *ScenarioStructurre.dat*:

Código B.1 Árbol de escenarios del problema estocástico.

```
set Stages := FirstStage SecondStage ;

set Nodes := RootNode

    # Precios1 (4 nodos PV y 4 nodos Wind)
    Node111
    Node112
    Node113
    Node114

    Node121
    Node122
    Node123
    Node124

    Node131
    Node132
    Node133
    Node134

    Node141
    Node142
    Node143
    Node144

    # Precios 2
    Node211
    Node212
    Node213
    Node214
```

```
Node221
Node222
Node223
Node224

Node231
Node232
Node233
Node234

Node241
Node242
Node243
Node244

# Precios 3
Node311
Node312
Node313
Node314

Node321
Node322
Node323
Node324

Node331
Node332
Node333
Node334

Node341
Node342
Node343
Node344

# Precios 4
Node411
Node412
Node413
Node414

Node421
Node422
Node423
Node424

Node431
Node432
Node433
```



```
Node434  
  
Node441  
Node442  
Node443  
Node444 ;
```

```
param NodeStage := RootNode FirstStage
```

```
Node111 SecondStage  
Node112 SecondStage  
Node113 SecondStage  
Node114 SecondStage
```

```
Node121 SecondStage  
Node122 SecondStage  
Node123 SecondStage  
Node124 SecondStage
```

```
Node131 SecondStage  
Node132 SecondStage  
Node133 SecondStage  
Node134 SecondStage
```

```
Node141 SecondStage  
Node142 SecondStage  
Node143 SecondStage  
Node144 SecondStage
```

```
# Precios 2
```

```
Node211 SecondStage  
Node212 SecondStage  
Node213 SecondStage  
Node214 SecondStage
```

```
Node221 SecondStage  
Node222 SecondStage  
Node223 SecondStage  
Node224 SecondStage
```

```
Node231 SecondStage  
Node232 SecondStage  
Node233 SecondStage  
Node234 SecondStage
```

```
Node241 SecondStage  
Node242 SecondStage  
Node243 SecondStage  
Node244 SecondStage
```

```
# Precios 3
```

```
Node311 SecondStage
Node312 SecondStage
Node313 SecondStage
Node314 SecondStage

Node321 SecondStage
Node322 SecondStage
Node323 SecondStage
Node324 SecondStage

Node331 SecondStage
Node332 SecondStage
Node333 SecondStage
Node334 SecondStage

Node341 SecondStage
Node342 SecondStage
Node343 SecondStage
Node344 SecondStage

# Precios 4
Node411 SecondStage
Node412 SecondStage
Node413 SecondStage
Node414 SecondStage

Node421 SecondStage
Node422 SecondStage
Node423 SecondStage
Node424 SecondStage

Node431 SecondStage
Node432 SecondStage
Node433 SecondStage
Node434 SecondStage

Node441 SecondStage
Node442 SecondStage
Node443 SecondStage
Node444 SecondStage ;

set Children[RootNode] := Node111
Node112
Node113
Node114

Node121
Node122
Node123
Node124
```

Node131
Node132
Node133
Node134

Node141
Node142
Node143
Node144

Node211
Node212
Node213
Node214

Node221
Node222
Node223
Node224

Node231
Node232
Node233
Node234

Node241
Node242
Node243
Node244

Node311
Node312
Node313
Node314

Node321
Node322
Node323
Node324

Node331
Node332
Node333
Node334

Node341
Node342
Node343
Node344

```
Node411
Node412
Node413
Node414

Node421
Node422
Node423
Node424

Node431
Node432
Node433
Node434

Node441
Node442
Node443
Node444 ;

param ConditionalProbability := RootNode      1.0

Node111 0.0251
Node112 0.022
Node113 0.016
Node114 0.0129

Node121 0.01
Node122 0.0088
Node123 0.0064
Node124 0.0052

Node131 0.0364
Node132 0.032
Node133 0.0231
Node134 0.0187

Node141 0.0539
Node142 0.0474
Node143 0.0343
Node144 0.0278

Node211 0.0013
Node212 0.0012
Node213 0.0008
Node214 0.0007

Node221 0.0005
Node222 0.0005
```

Node223 0.0003
Node224 0.0003

Node231 0.0019
Node232 0.0017
Node233 0.0012
Node234 0.001

Node241 0.0028
Node242 0.0025
Node243 0.0018
Node244 0.0015

Node311 0.0231
Node312 0.0203
Node313 0.0147
Node314 0.0119

Node321 0.0092
Node322 0.0081
Node323 0.0059
Node324 0.0048

Node331 0.0335
Node332 0.0294
Node333 0.0213
Node334 0.0173

Node341 0.0497
Node342 0.0436
Node343 0.0316
Node344 0.0256

Node411 0.0165
Node412 0.0145
Node413 0.0105
Node414 0.0085

Node421 0.0066
Node422 0.0058
Node423 0.0042
Node424 0.0034

Node431 0.0239
Node432 0.021
Node433 0.0152
Node434 0.0123

Node441 0.0355
Node442 0.0312

```
Node443 0.0226
Node444 0.0183 ;

set Scenarios :=
  # Precios1 (4 nodos PV y 3 nodos Wind)
    Scenario111
    Scenario112
    Scenario113
    Scenario114

    Scenario121
    Scenario122
    Scenario123
    Scenario124

    Scenario131
    Scenario132
    Scenario133
    Scenario134

    Scenario141
    Scenario142
    Scenario143
    Scenario144

  # Precios 2
    Scenario211
    Scenario212
    Scenario213
    Scenario214

    Scenario221
    Scenario222
    Scenario223
    Scenario224

    Scenario231
    Scenario232
    Scenario233
    Scenario234

    Scenario241
    Scenario242
    Scenario243
    Scenario244

  # Precios 3
    Scenario311
    Scenario312
    Scenario313
```

```
Scenario314

Scenario321
Scenario322
Scenario323
Scenario324

Scenario331
Scenario332
Scenario333
Scenario334

Scenario341
Scenario342
Scenario343
Scenario344

# Precios 4
Scenario411
Scenario412
Scenario413
Scenario414

Scenario421
Scenario422
Scenario423
Scenario424

Scenario431
Scenario432
Scenario433
Scenario434

Scenario441
Scenario442
Scenario443
Scenario444 ;

param ScenarioLeafNode :=
  # Precios1 (4 nodos PV y 3 nodos Wind)
    Scenario111 Node111
    Scenario112 Node112
    Scenario113 Node113
    Scenario114 Node114

    Scenario121 Node121
    Scenario122 Node122
    Scenario123 Node123
    Scenario124 Node124
```

```
Scenario131 Node131
Scenario132 Node132
Scenario133 Node133
Scenario134 Node134

Scenario141 Node141
Scenario142 Node142
Scenario143 Node143
Scenario144 Node144

# Precios 2
Scenario211 Node211
Scenario212 Node212
Scenario213 Node213
Scenario214 Node214

Scenario221 Node221
Scenario222 Node222
Scenario223 Node223
Scenario224 Node224

Scenario231 Node231
Scenario232 Node232
Scenario233 Node233
Scenario234 Node234

Scenario241 Node241
Scenario242 Node242
Scenario243 Node243
Scenario244 Node244

# Precios 3
Scenario311 Node311
Scenario312 Node312
Scenario313 Node313
Scenario314 Node314

Scenario321 Node321
Scenario322 Node322
Scenario323 Node323
Scenario324 Node324

Scenario331 Node331
Scenario332 Node332
Scenario333 Node333
Scenario334 Node334

Scenario341 Node341
Scenario342 Node342
Scenario343 Node343
```



```
Scenario344 Node344

# Precios 4
  Scenario411 Node411
  Scenario412 Node412
  Scenario413 Node413
  Scenario414 Node414

  Scenario421 Node421
  Scenario422 Node422
  Scenario423 Node423
  Scenario424 Node424

  Scenario431 Node431
  Scenario432 Node432
  Scenario433 Node433
  Scenario434 Node434

  Scenario441 Node441
  Scenario442 Node442
  Scenario443 Node443
  Scenario444 Node444 ;

set StageVariables[FirstStage] := u_g[*,*,*] ;
set StageVariables[SecondStage] := p_g[*,*,*]
  ls[*,*,*]
  soc[*,*,*]
  V_i[*,*]
  theta_i[*,*] ;

param StageCost := FirstStage OBJ
  SecondStage OBJ ;

param ScenarioBasedData := False ;
```


Índice de Figuras

1.1	Predicciones de crecimiento de la población mundial [13]	1
1.2	Predicciones de crecimiento de la demanda de energía primaria [21]	2
1.3	Riesgos e impactos del cambio climático [36]	3
1.4	Evolución necesaria del vehículo eléctrico para alcanzar el objetivo de 2 °C según [17]	4
1.5	Actualización de las previsiones de capacidad instalada de energía solar [21]	5
1.6	Evolución de la potencia renovable instalada a nivel mundial [17]	6
1.7	Esquema general de una red de distribución eléctrica inteligente [15]	7
1.8	Representación de los elementos de una red de distribución eléctrica inteligente (fuente: elaboración propia)	8
1.9	Categorías de DSM [39]	10
1.10	Impacto de diferentes tecnologías en el consumo eléctrico diario [39]	11
2.1	Algoritmos para resolver problemas de optimización MINLP (fuente: elaboración propia)	17
2.2	Resumen de Pyomo	18
3.1	Red de referencia [40]	22
3.2	Perfiles de las cargas de la red [45]	24
3.3	Incremento de potencia disponible para control primario de frecuencia (reserva de subida)	32
3.4	Coste medio de importación de energía a la red	38
3.5	Perfiles medios de generación renovable	38
3.6	Creación de escenarios de precios del mercado eléctrico	40
3.7	Escenarios de costes de la energía importada por la red	40
3.8	Creación de escenarios de generación de energía eólica	40
3.9	Escenarios de generación eólica	41
3.10	Escenarios de generación fotovoltaica	41
4.1	Producción de los generadores renovables	43
4.2	Consumo industrial en el caso base	44
4.3	Voltajes, potencias y energía en el caso base	45
4.4	Voltajes, potencias y energía con importación limitada	46
4.5	Desconexión de la industria con importación limitada	46
4.6	Voltajes, potencias y energía con coste de desconexión reducido	47
4.7	Desconexión de las industrias con coste de desconexión reducido	48
4.8	Producción de los generadores renovables aumentada	48
4.9	Voltajes, potencias y energía con alta generación de renovables	49
4.10	Voltajes, potencias y energía con restricciones de reserva	50
4.11	Estado de los generadores en el caso estocástico	52

Índice de Tablas

2.1	Problemas de optimización	15
3.1	Parámetros de las líneas de la red [45]	23
3.2	Parámetros de las cargas de la red [45]	24
3.3	Parámetros de los generadores de la red [45]	25
3.4	Formulación del problema optimización determinista	33
3.5	Parámetros de la red [45]	37
3.6	Costes de la red	37
3.7	Parámetros de las baterías	39

Índice de Códigos

3.1	Declaración del modelo, sets y variables en Pyomo	34
3.2	Declaración de parámetros mediante función en Pyomo	34
3.3	Declaración de función objetivo y restricciones en Pyomo	35
3.4	Asignación de valores a los parámetros de Pyomo a partir de Excel	35
3.5	Resolución del problema matemático y creación de gráficas en Pyomo	36
3.6	Ejecución del problema estocástico a través del terminal de comandos de Windows	42
A.1	Modelo determinista en Pyomo	55
B.1	Árbol de escenarios del problema estocástico	83

Bibliografía

- [1] *COIN-OR: Computational Infrastructure for Operations Research – Open-source Software for the Operations Research Community.*
- [2] *Conference of the Parties (COP) | UNFCCC.*
- [3] *El grito de los jóvenes contra el cambio climático se convierte en global | Sociedad | EL PAÍS.*
- [4] *GAMS/DICOPT.*
- [5] *Global Warming of 1.5 °C — IPCC Special Report.*
- [6] *IEEE PES Intelligent Systems Subcommittee - Open Data Sets.*
- [7] *La transición energética. Una Nueva Cultura de la Energía.*
- [8] *Lundi Vert.*
- [9] *Operador del Mercado Ibérico de Energía (OMIE).*
- [10] *Pyomo Documentation — Pyomo 5.1 documentation.*
- [11] *Sowee : Contrats énergie, Station connectée, Pilotage chauffage.*
- [12] *Sustainable Development Goals - United Nations Sustainable Development.*
- [13] *World Population Prospects - Population Division - United Nations.*
- [14] *Couenne, an exact solver for nonconvex MINLPs, (2006).*
- [15] *Electricity Grid Modernization: Progress Being Made on Cybersecurity Guidelines, but Key Challenges Remain to be Addressed, Tech. Report GAO-11-117, U.S. Government Accountability Office, 1 2011.*
- [16] *OMS | Los efectos sobre la salud, 2014.*
- [17] *Chapter 2 of Perspectives for the Energy Transition - Investment Needs for a Low-Carbon Energy System, Tech. report, OECD/IEA, 2017.*
- [18] *IEA Global Energy and CO2 Status Report, Tech. report, 2017.*
- [19] *EC Communication - A Clean Planet for all A European strategic long-term vision for a prosperous, modern, competitive and climate neutral economy, Tech. report, European Commission, 2018.*

- [20] *Negative emission technologies: What role in meeting Paris Agreement targets?*, Tech. report, EASAC, 2018.
- [21] *WEO 2018*, 2018.
- [22] *L'appel des 500 pour un « lundi vert » : « Nous nous engageons à remplacer la viande et le poisson chaque lundi »*, 1 2019.
- [23] Manuel Barragan, Juan M. Mauricio, Alejandro Marano, Manuel Nieves, Julio Churio, Jose M. Maza-Ortega, Esther Romero, and Antonio Gomez, *Operational benefits of multiterminal DC-links in active distribution networks*, IEEE Power and Energy Society General Meeting, 2012.
- [24] Pierre Bonami and Jon Lee, *BONMIN Users' Manual*, Tech. report, 2011.
- [25] Joseph Czyzyk, Michael P. Mesnier, and Jorge J. More, *NEOS server*, IEEE computational science & engineering (1998).
- [26] European SmartGrids Technology Platform, *Vision and strategy for Europe's electricity networks of the future*, vol. 19, 2006.
- [27] Flavia Gangale, Julija Vasiljevska, and Catalin Felix Covrig, *Smart grid projects outlook 2017 Facts, figures and trends in Europe*, (2017).
- [28] P. Godinho Matos, P. Lbano Monteiro, M. So Miguel Oliveira, A. Aires Messias, A.M. Veiga, and P.R. Daniel, *InovGrid, a smart vision for a next generation distribution system*, 22nd International Conference and Exhibition on Electricity Distribution (CIRED 2013), 2013.
- [29] William E Hart, Carl D Laird, Jean-Paul Watson, David L Woodruff, Gabriel A Hackebeil, Bethany L Nicholson, and John D Sirola, *Pyomo—optimization modeling in python*, second ed., vol. 67, Springer Science & Business Media, 2017.
- [30] William E. Hart, Jean Paul Watson, and David L. Woodruff, *Pyomo: Modeling and solving mathematical programs in Python*, Mathematical Programming Computation **3(3)** (2011), 219–260.
- [31] John D. Hunter, *Matplotlib: A 2D Graphics Environment*, Computing in Science & Engineering **9** (2007), no. 3, 90–95.
- [32] International renewable energy Agency, *Renewable Power Generation Costs in 2017*, 2018.
- [33] Cabrera G. Játiva J., Constante G., *Flujo de Potencia por Newton-Raphson con el Jacobiano Calculado en las Ecuaciones de Errores de Potencia*, REVISTA POLITÉCNICA **33** (2014), no. 3, 7.
- [34] Jan Kronqvist, David E Bernal, Andreas Lundell, and Ignacio E Grossmann, *A Review and Comparison of Solvers for Convex MINLP*, Tech. report, 2018.
- [35] Jose L. Martinez-Ramos, Alejandro Marano-Marcolini, Francisco P. Garcia-Lopez, Fernando Almagro-Yravedra, Ahmet Onen, Yeliz Yoldas, Mounir Khiat, Leila Ghomri, and Nunziatina Fragale, *Provision of ancillary services by a smart microgrid: An OPF approach*, 2018 International Conference on Smart Energy Systems and Technologies, SEST 2018 - Proceedings (2018).

- [36] Valérie Masson-Delmotte, Panmao Zhai, Hans-Otto Pörtner, Debra Roberts, Jim Skea, Priyadarshi R Shukla, Anna Pirani, Wilfran Moufouma-Okia, Clotilde Péan, Roz Pidcock, Sarah Connors, J B Robin Matthews, Yang Chen, Xiao Zhou, Melissa I Gomis, Elisabeth Lonnoy, Tom Maycock, Melinda Tignor, and Tim Waterfield, *Global warming of 1.5°C - Figure SPM.2*, Tech. report, IPCC.
- [37] Thomas Morstyn, Niall Farrell, Sarah J. Darby, and Malcolm D. McCulloch, *Using peer-to-peer energy-trading platforms to incentivize prosumers to form federated power plants*, 2018.
- [38] O. Edenhofer, R. Pichs-Madruga, Y. Sokona, K. Seyboth, P. Matschoss, S. Kadner, T. Zwickel, P. Eickemeier, G. Hansen, S. Schlömer, and C. von Stechow, *IPCC, 2011: Summary for Policymakers. In: IPCC Special Report on Renewable Energy Sources and Climate Change Mitigation*, Cambridge University Press, 2011.
- [39] Peter Palensky and Dietmar Dietrich, *Demand side management: Demand response, intelligent energy systems, and smart loads*, IEEE Transactions on Industrial Informatics (2011).
- [40] Convener Prof, Kai Strunz, Chad Abbey, Christophe Andrieu, Ryan C Campbell, and Robert Fletcher, *Benchmark Systems for Network Integration of Renewable and Distributed Energy Resources*, no. July, 2009.
- [41] I. Quesada and I. E. Grossmann, *An LP/NLP based branch and bound algorithm for convex MINLP optimization problems*, Computers and Chemical Engineering **16** (1992), no. 10-11, 937–947.
- [42] R Core Team, *R: A Language and Environment for Statistical Computing*, 2014.
- [43] Ramyar Rashed Mohassel, Alan Fung, Farah Mohammadi, and Kaamran Raahemifar, *A survey on Advanced Metering Infrastructure*, International Journal of Electrical Power & Energy Systems **63** (2014), 473–484.
- [44] Anna Dalla Rosa, Michelangelo Vianello, Elisa Maria Galliani, and Pasquale Anselmi, *Longitudinal Cluster Analysis*, (2017).
- [45] K. Rudion, A. Orths, Z.A. Styczynski, and K. Strunz, *Design of benchmark of medium voltage distribution network for investigation of DG integration*, 2006 IEEE Power Engineering Society General Meeting (2006), 6 pp.
- [46] Jeffrey D. Sachs, *From millennium development goals to sustainable development goals*, 2012.
- [47] Alireza Soroudi, *Power system optimization modeling in GAMS*, 2017.
- [48] Brett P. Spigarelli and S. Komar Kawatra, *Opportunities and challenges in carbon dioxide capture*, 2013.
- [49] M Tawarmalani and N. V. Sahinidis, *A polyhedral branch-and-cut approach to global optimization*, Mathematical Programming **103** (2005), no. 2, 225–249.
- [50] Tine L. Vandoorn, Juan C. Vasquez, Jeroen De Kooning, Josep M. Guerrero, and Lieven Vandeveldde, *Microgrids: Hierarchical control and an overview of the control and reserve management strategies*, IEEE Industrial Electronics Magazine **7** (2013), no. 4, 42–55.
- [51] A. Watcher and L. T. Biegler, *On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming*, Mathematical Programming **106(1)** (2006), 25–27.

- [52] Jean Paul Watson and David L. Woodruff, *Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems*, Computational Management Science (2011).
- [53] Jean Paul Watson, David L. Woodruff, and William E. Hart, *PySP: Modeling and solving stochastic programs in Python*, Mathematical Programming Computation (2012).
- [54] Nanpeng Yu, Sunil Shah, Raymond Johnson, Robert Sherick, Mingguo Hong, and Kenneth Loparo, *Big data analytics in power distribution systems*, 2015 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference, ISGT 2015, 2015.