



Universidad
Zaragoza

Trabajo Fin de Máster

Optimización de sistemas de autoconsumo con baterías mediante el algoritmo de enjambre de partículas.

Optimization of self-consumption systems with batteries using the particle swarm optimization algorithm.

Autor/es

Daniel Vila Guinda

Director/es

José Luis Bernal Agustín

Máster Universitario en Energías Renovables y Eficiencia Energética

Escuela de Ingeniería y Arquitectura (Zaragoza)

2024

OPTIMIZACIÓN DE SISTEMAS DE AUTOCONSUMO CON BATERÍAS MEDIANTE EL ALGORITMO DE ENJAMBRE DE PARTÍCULAS. RESUMEN

El siguiente proyecto se centra en la optimización de sistemas de autoconsumo que incorporan paneles solares, aerogeneradores y baterías conectadas a la red eléctrica. El objetivo es mejorar la eficiencia y la rentabilidad de las instalaciones durante toda la vida útil de los componentes, obteniendo el mayor autoconsumo posible. La optimización se realiza mediante la aplicación del algoritmo Particle Swarm Optimization (PSO) implementado en Matlab.

El estudio se enfoca en la integración de los diferentes elementos y el algoritmo PSO se utiliza para obtener los valores óptimos de los parámetros del sistema, maximizando el autoconsumo y minimizando los costes asociados al mismo.

La metodología incluye la modelización de los componentes del sistema, la definición de los objetivos de optimización, en este caso disminuir la función coste al máximo, y la aplicación del algoritmo PSO para encontrar las soluciones más eficientes. Una vez desarrollado el código se llevan a cabo diferentes simulaciones y análisis de sensibilidad para evaluar el rendimiento del sistema en diferentes escenarios y bajo diversas condiciones.

Los resultados muestran la eficacia del enfoque propuesto, mostrando mejoras significativas tanto en el ámbito energético como económico.

OPTIMIZATION OF SELF-CONSUMPTION SYSTEMS WITH BATTERIES USING THE PARTICLE SWARM OPTIMIZATION ALGORITHM. ABSTRACT

The following Project focuses on the optimization of self-consumption systems that incorporate solar panels, wind turbines and batteries connected to the electrical grid. The objective is to improve the efficiency and profitability of the systems during the useful-life of the components. The optimization is carried out by applying the Particle Swarm Optimization (PSO) algorithm implemented in Matlab.

The study focuses on the integration of the different elements and the PSO algorithm is used with the objective to obtain the optimal values of the system parameters, maximizing self-consumption and minimizing the costs.

The methodology includes the modeling of the system components, the definition of the optimization objectives, which in this case, is reducing the cost function to the maximum, and the application of the PSO algorithm to find the most efficient solutions. Once the code has been developed, different simulations and analysis are carried out to evaluate the performance of the system in different scenarios and under different conditions.

The results show the effectiveness of the proposed approach, showing significant improvements in both energy and economic fields.

INDICE

OPTIMIZACIÓN DE SISTEMAS DE AUTOCONSUMO CON BATERÍAS MEDIANTE EL ALGORITMO DE ENJAMBRE DE PARTÍCULAS. RESUMEN	2
OPTIMIZATION OF SELF-CONSUMPTION SYSTEMS WITH BATTERIES USING THE PARTICLE SWARM OPTIMIZATION ALGORITHM. ABSTRACT	2
LISTA DE FIGURAS.....	5
LISTA DE TABLAS.....	6
1. INTRODUCCIÓN	7
2. REVISIÓN DEL ESTADO DEL ARTE	8
2.1. LEGISLACIÓN DEL AUTOCONSUMO EN ESPAÑA.....	8
2.2. ANALISIS Y OPTIMIZACION DE EDIFICIOS CON AUTOCONSUMO EN ESPAÑA.....	11
2.3. MODELOS DE OPTIMIZACIÓN (PSO) PARA PROBLEMA DE RUTAS	15
2.4. CONCLUSIONES.....	16
3. METODOLOGÍA Y OBJETIVO	18
4. AUTOCONSUMO, AGENTES Y MODALIDADES EN ESPAÑA	19
4.1. AUTOCONSUMO CON ALMACENAMIENTO	19
4.2. AGENTES	19
4.3. MODALIDADES.....	20
5. MODELOS DE OPTIMIZACIÓN.....	23
5.1. ANT COLONY OPTIMIZATION.....	23
5.2. SIMULATED ANNEALING	24
5.3. PARTICLE SWARM OPTIMIZATION.....	24
6. CASOS DE ESTUDIO Y EQUIPOS SELECCIONADOS	27
6.1. FUNCIONAMIENTO, HIPÓTESIS Y ESCENARIOS	27
6.2. ELEMENTOS SELECCIONADOS	28
6.2.1. PANELES FOTOVOLTAICOS.....	28
6.2.2. AEROGENERADORES	28
6.2.3. BATERIAS.....	30
7. DESCARGA DE DATOS PARA ALGORITMO	31
7.1. VALORES DE GENERACIÓN	31
7.1.1. GENERACIÓN SOLAR	31
7.1.2. GENERACIÓN EÓLICA	32
7.2. VALORES DE CONSUMO.....	33
7.3. VALORES DE PRECIO DE RED.....	33
8. DESARROLLO DEL PROGRAMA DE SIMULACIÓN.....	35
8.1. FUNCIÓN OBJETIVO	35
8.2. CODIGO DE OPTIMIZACIÓN	35
8.2.1. VARIABLES DEL PROBLEMA.....	35

8.2.2.	PARAMETROS DE SIMULACIÓN.....	36
8.2.3.	PRIMERA ITERACIÓN	36
8.2.4.	RESTO DE ITERACIONES	37
8.3.	ALGORITMO	38
9.	RESULTADOS SIMULACIONES.....	40
9.1.	NÚMERO ÓPTIMO DE ELEMENTOS	40
9.2.	AHORRO VIDA ÚTIL.....	41
10.	CONCLUSIONES	42
11.	REFERENCIAS BIBLIOGRÁFICAS	43
12.	ANEXOS	44
12.1.	CÓDIGO COMPLETO ALGORITMO OPTIMIZACIÓN	44
12.2.	DESCARGA DE DATOS LOAD_PROFILE_GENERATOR.....	59
12.3.	DESCARGA DE DATOS DE PRECIO DE RED	61
12.4.	FICHA TÉCNICA PANELES SOLARES	64
12.5.	FICHA TÉCNICA AEROGENERADORES	66
12.6.	FICHA TÉCNICA BATERÍAS	68

LISTA DE FIGURAS

Ilustración 1. Método utilizado para evaluar sistemas de generación. Fuente: [2].....	9
Ilustración 2. Cálculo payback simple. Fuente: [2].....	9
Ilustración 3. Ejemplo generación solar y demanda durante un año y gráfico del payback. Fuente: [2].....	10
Ilustración 4. Gráfico de los diferentes estudios a realizar en el artículo. Fuente: [1]	11
Ilustración 5. Gráfico de la gestión de la energía. Fuente: [1]	12
Ilustración 6. Irradiancia a lo largo de un año. Fuente: [1]	13
Ilustración 7. Demanda de una vivienda a lo largo de un año. Fuente: [1]	13
Ilustración 8. Gráfico precio energía eléctrica. Fuente: [1].....	14
Ilustración 9. Ejemplo gráfico y matricial de problema de rutas multimodal. Fuente: [3]	16
Ilustración 10. Conexión instalación autoconsumo a red interior. Fuente: [4]	21
Ilustración 11. Conexión instalación colectiva a red interior. Fuente: [4]	21
Ilustración 12. Esquema conexión autoconsumo individual y colectivo a través de red. Fuente: [4]	22
Ilustración 13. Ant Colony Optimization. Fuente: Google	23
Ilustración 14. Método funcionamiento Simulated Annealing. Fuente: Youtube	24
Ilustración 15. Principio de funcionamiento PSO.....	25
Ilustración 16. Instalación autoconsumo para una vivienda. Fuente: Google.....	27
Ilustración 17. Coeficientes de potencia según aerogenerador	29
Ilustración 18. NinjaRenewables. Fuente: GOOGLE.....	31
Ilustración 19. Código Matlab valores de consumo.....	33
Ilustración 20. Código Matlab valores precio de red.	34
Ilustración 21. Función coste MATLAB.....	35
Ilustración 22. Definición de las variables del problema.	36
Ilustración 23. Definición parámetros del problema.	36
Ilustración 24. Generación de variables de estado de forma aleatoria.	37
Ilustración 25. Actualización mejor posición y mejor coste.....	37
Ilustración 26. Actualización de los valores de las variables de estado.	38
Ilustración 27. Flujograma algoritmo	39
Ilustración 28. Convergencia del resultado.....	40
Ilustración 29. Resultados simulación para los diferentes escenarios.....	40
Ilustración 30. Ahorro en costes con instalación de autoconsumo.	41
Ilustración 31. Vista general Load Profile Generator.	59
Ilustración 32. Ventana cálculo perfil de consumo.	59
Ilustración 33. Combinaciones según las características de simulación.	60
Ilustración 34. Periodo de tiempo de descarga de valores de consumo.	60
Ilustración 35. Excel con perfil de consumo por minuto.....	60
Ilustración 36. Pantalla de inicio ESIOS. Fuente: ESIOS.....	61
Ilustración 37. Ventana análisis OMIE. Fuente: ESIOS.	61
Ilustración 38. búsqueda información ESIOS. Fuente: ESIOS.....	62
Ilustración 39. Precio horario de la red año 2022. Fuente: ESIOS.	62

LISTA DE TABLAS

Tabla 1. Ubicación ciudades a estudiar	28
Tabla 2. Diferentes escenarios para estudiar	28
Tabla 3. Características paneles fotovoltaicos	28
Tabla 4. Características aerogenerador	29
Tabla 5. Características baterías.....	30

1. INTRODUCCIÓN

El crecimiento exponencial de la demanda energética y la necesidad del uso de fuentes de energías renovables, han llevado a un aumento notable del uso de sistemas de autoconsumo, en los que se destacan los sistemas de autoconsumo basados en energía solar fotovoltaica. Estos sistemas permiten generar su propia electricidad a los usuarios, disminuyendo la dependencia con la red eléctrica y contribuyendo con el medio ambiente mediante la reducción de gases de efecto invernadero. Sin embargo, no siempre es posible garantizar un suministro energético constante y eficiente, siendo esto un gran desafío.

La integración de sistemas de almacenamiento de energía en baterías con sistemas de autoconsumo es una solución viable, permitiendo almacenar los excedentes de energía en periodos de alta generación para usarla durante los periodos de alta demanda o baja producción. No obstante, la gestión óptima de la energía almacenada es una tarea bastante compleja que requiere de estrategias avanzadas de optimización.

En el proyecto se propone el desarrollo de un sistema de optimización para la gestión eficiente de un sistema de autoconsumo con generación solar y eólica con baterías conectado a la red eléctrica. La optimización se realiza utilizando el algoritmo de optimización de enjambre de partículas (PSO), implementado en MATLAB. El algoritmo PSO es una técnica de optimización metaheurística inspirada en el comportamiento colectivo de enjambres en la naturaleza. Es un método adecuado para la gestión eficiente de la energía en sistemas de autoconsumo.

El principal objetivo es el desarrollo de un modelo que optimice el flujo energético entre la generación fotovoltaica, eólica, el almacenamiento en baterías y el consumo de la red, de forma que se minimicen los costes operativos del sistema, teniendo en cuenta diferentes factores como el precio de la red eléctrica, los perfiles de consumo y la generación energética.

Como resumen, se busca contribuir al campo de la gestión energética ofreciendo una solución eficaz para la integración de sistemas de autoconsumos con baterías y la red eléctrica, utilizando técnicas de optimización avanzadas, que permitan una transición hacia un modelo energético más sostenible y eficaz.

2. REVISIÓN DEL ESTADO DEL ARTE

2.1. LEGISLACIÓN DEL AUTOCONSUMO EN ESPAÑA

En los países europeos, se está llevando a cabo una transición hacia las energías renovables, aumentando significativamente las instalaciones de autoconsumo. Muchos estudios indican que en nuestro país las regulaciones y el apoyo a los sistemas fotovoltaicos siguen siendo insuficientes.

En los últimos años, se ha iniciado una transición energética con el objetivo de alcanzar un modelo basado en energías renovables para frenar el cambio climático y comenzar el periodo de descarbonización. Las nuevas regulaciones se han implementado para promover el autoconsumo y la generación con energías renovables. Los tres objetivos principales son mejorar la eficiencia energética, aumentar la participación del usuario en el mercado eléctrico y la reducción de emisión de gases de efecto invernadero.

Debido a las nuevas políticas y la tendencia al alza en el precio de la electricidad, el número de instalaciones de generación ha aumentado considerablemente. Las plantas de generación fotovoltaica son las más numerosas, aunque también destacan otras formas de generación, como por ejemplo la eólica.

En el caso de España, se han instalado una gran cantidad de plantas fotovoltaicas en los últimos años, favorecidas por la facilidad de instalación y su asequible precio. Este crecimiento también se debe, en parte, a la nueva legislación de autoconsumo, que ha mejorado y reducido el periodo de retorno de las inversiones. En uno de los artículos se comparan los diferentes escenarios incluidos en la nueva legislación de autoconsumo española en zonas rurales.

- **NUEVA LEGISLACIÓN DE AUTOCONSUMO**

En 2019 el gobierno español declaró un real decreto donde la tasa al sol quedaba suprimida, además de otros importantes cambios que se muestran a continuación.

- Eliminación de la tasa al sol
- Eliminación de los límites de potencia en una instalación. La potencia instalada puede ser mayor que la potencia contratada.
- Nuevas compensaciones que permiten al usuario vender la energía generada que no es consumida añadiéndola a la red eléctrica. Dentro de esta modalidad de venta existen dos maneras posibles, acogido a compensación o no acogido a compensación. La no acogida a compensación no tiene apenas formalidades administrativas y no hay tasas a la hora de inyectar la energía a la red. Está limitado a instalaciones fotovoltaicas de hasta 100kW. La energía sobrante se vende a la red a un precio fijo. De la otra forma para acogerse a compensación es necesario estar registrado como propiedad generadora. El usuario recibe compensación financiera según el precio del mercado.
- Se permiten las instalaciones compartidas, aunque tienen una serie de limitaciones.
- Se permite alquilar tejados a terceras partes para generar electricidad y compartir beneficios.

Esta serie de beneficios han favorecido claramente a las instalaciones privadas de generación.

- **COMPARACIÓN DEL PAYBACK PARA DIFERENTES ESCENARIOS**

En este apartado se describe el método usado para evaluar económicamente los sistemas de generación fotovoltaica establecidos en el último Real Decreto. Se muestra un esquema general del método utilizado [Figura 1]

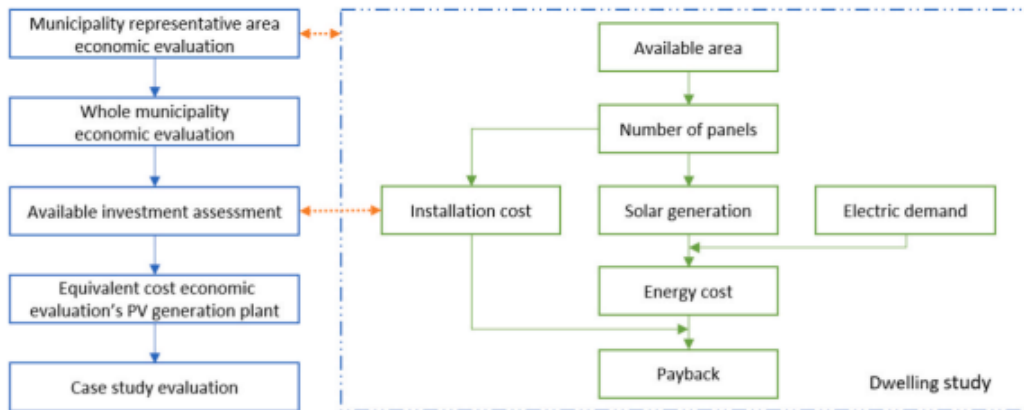


Ilustración 1. Método utilizado para evaluar sistemas de generación. Fuente: [2]

Lo primero de todo es necesario obtener el registro de los precios horarios de consumo de la vivienda de todo un año. El precio horario se obtiene de la web de Red Eléctrica de España.

Lo segundo es conocer el área disponible de la vivienda para instalación de paneles solares, lo que permite calcular la potencia pico que puede llegar a tener la instalación aplicando un factor de pérdidas para su cálculo. Por tanto, la potencia total de la instalación se calcula a partir del número de paneles, la potencia pico de cada uno de los paneles y las pérdidas totales de cada panel.

Una vez conocida la potencia pico, se tiene en cuenta un coste de instalación en función de la potencia pico de la instalación y por tanto se puede obtener rápidamente el coste inicial de la instalación. Este valor es importante conocerlo a la hora de poder calcular el periodo de retorno de la inversión.

Por último, se obtiene la generación solar de una instalación del mismo tamaño para el periodo de un año. Una vez todos estos datos están establecidos, es posible realizar el método de cálculo para los diferentes escenarios.

Se realizan los cálculos para 4 escenarios diferentes: Sin instalación de autoconsumo, con instalación sin venta de excedente, con instalación y venta acogida a compensación y con instalación y venta sin acogida a compensación.

- El primer escenario es importante conocerlo para analizar las ventajas de los siguientes escenarios que se van a estudiar. Una vez es conocida la demanda y el precio horarios de todo un año, es posible conocer el coste que el usuario va a pagar cada hora del año.
- En este segundo escenario se utiliza la energía generada por los paneles para autoconsumo en vez de comprarla a la red. Puede haber momentos con cero consumos de red, pero en este caso es un método unidireccional y no se inyecta la energía sobrante. Si la energía generada es mayor que el consumo, habrá periodos en los que la compra de energía a la red sea nula. En este caso la compra de energía a la red disminuye notablemente y por tanto la factura al final de año es menor. Para poder comparar con los diferentes escenarios se calcula el payback [Figura 2] o periodo de retorno de manera simple dividiendo la inversión realizada entre el ahorro anual.

$$PB_2 = \frac{I_0}{BC_1 - BC_2}$$

Ilustración 2. Cálculo payback simple. Fuente: [2]

- En el tercer caso se produce intercambio de energía bidireccional en el cual se consume y se inyecta energía de la red. En el momento que se ha generado más de lo que se ha consumido, el excedente se inyecta a la red, obteniendo beneficio económico, con un precio de venta fijo. En este escenario no es posible obtener un beneficio económico a final de mes, sino que como mucho se puede conseguir que el coste de tu factura sea 0. Por tanto, en este escenario el coste anual es el mismo, pero restando el valor de ahorro inyectado a la red para cada mes. Sumando el coste de todos los meses se tiene el coste anual y de la misma forma que en el escenario anterior se puede calcular el Payback.
- En el último de los casos se recibe compensación económica por kWh inyectado a la red, pero a precio variable según esté en ese momento en el operador de red español. En este caso se tiene un 7% de impuestos, y un coste de acceso de 0,5 euros/MWh. En este escenario sí que es posible obtener beneficio económico, es decir, la cuenta puede ser negativa y recibir dinero mensualmente por la energía generada. De la misma forma que antes, se puede calcular el coste anual y por lo tanto el Payback.

Se ha seleccionado el municipio Aras de Olmos para el estudio de viabilidad económica de un sistema de autoconsumo solar. Se realiza el estudio en un área representativa del pueblo para luego extrapolarlo y obtener el presupuesto total en el municipio (aproximadamente el 7% de la población).

Se determina el número de viviendas donde es posible instalar paneles fotovoltaicos y el número de paneles que se pueden instalar en cada vivienda y se calcula la potencia pico y por tanto el coste instalación. Para el estudio se han utilizado paneles de 330 Wp con un área de 1x1,9m². Conociendo las tarifas de los usuarios, se conoce el consumo anual de cada vivienda, se calcula el número de paneles, la potencia pico, la potencia útil y el coste total de inversión.

A partir de ahí, para cada una de las viviendas y según los paneles y generación de cada una, se calcula el coste anual para los 4 escenarios mencionados anteriormente. Se observa que cada escenario planteado es mejor que el anterior. Los resultados muestran que el escenario sin paneles es el más caro, mientras que el escenario con instalación de autoconsumo y venta del excedente según el precio del mercado el más barato. Obteniendo un periodo de retorno en torno a 30 años con instalación fotovoltaica, de 20 años con venta de excedente a precio fijo y de 15 años con venta de excedente según mercado.

A continuación, se muestra un ejemplo de la demanda y generación típica de una vivienda. Cabe destacar de la siguiente imagen, que la generación solar a últimas horas del día es bastante más baja que la demanda. También se muestra un gráfico de los valores de retorno obtenidos a partir de la media y desviaciones típicas de los resultados [Figura 3].

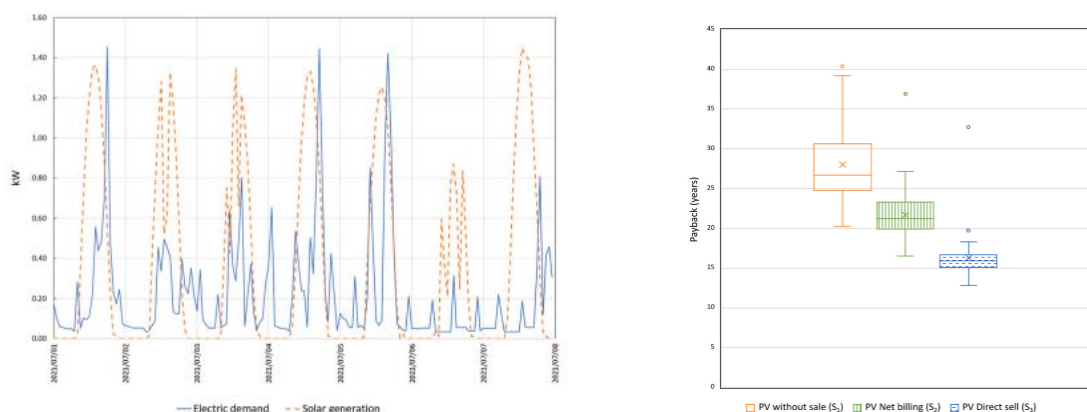


Ilustración 3. Ejemplo generación solar y demanda durante un año y gráfico del payback. Fuente: [2]

2.2. ANALISIS Y OPTIMIZACION DE EDIFICIOS CON AUTOCONSUMO EN ESPAÑA

La mayoría de los países, especialmente en la Unión Europea están preocupados por el aumento del consumo eléctrico y la transición del carbón a las energías renovables. Muchos aspectos animan a incrementar el nivel de autoconsumo en edificios residenciales. Por un lado, los consumidores hacen una inversión más rentable, ya que se benefician de precios de luz más bajos y periodos de retorno menores. Además, el uso de aplicaciones inteligentes permite crear un horario de demandas para incrementar el ratio de autoconsumo y disminuir los picos de autoconsumo. Las principales ventajas del autoconsumo son el uso de energías renovables y la disminución de pérdidas en las redes de distribución entre muchas cosas y aunque una única red que cubra las necesidades de todas las cargas no es viable aún, es importante remarcar que es el camino a seguir.

Se ha demostrado la viabilidad de estos proyectos de autoconsumo tal y como se ha comentado en el artículo anterior [2], aunque aún hay falta de apoyo legislativo en España. Se han realizado una gran cantidad de estudios tanto teniendo en cuenta los subsidios como no, y para una gran cantidad de escenarios, llegando a la conclusión de que el marco actual promueve y apoya una estructura de autoconsumo con energías renovables, almacenamiento y comunidades energéticas. Durante la realización del proyecto este trabajo se va a realizar la comprobación de si el uso de baterías o no en las instalaciones de autoconsumo son rentables del todo o no.

• METODOLOGÍA DEL ESTUDIO

El estudio se centra en un edificio compuesto por 12 viviendas, donde se analiza un sistema de autoconsumo colectivo desde diferentes puntos de vista con el objetivo de comparar los dos tipos de autoconsumo, evaluar el papel de programar los electrodomésticos y optimizar los coeficientes para compartir la energía generada.

Inicialmente el estudio fue dividido en dos tipos de instalaciones, individual y colectiva (12 viviendas) y aplicando las dos modalidades de autoconsumo para cada instalación. Se ha calculado con modelo diario y con modelo anual donde el uso del precio horario justifica la comparación entre ambos modelos. Se realizan múltiples estudios para diferentes escenarios con el objetivo de sacar conclusiones una vez obtenidos los resultados. Un resumen de los estudios se muestra a continuación [Figura 4].

Table 1
Different scenarios based on dwelling, self-consumption and time model types.

Study cases	Type of dwelling		Self-consumption modality		Time model		Number of scenarios
	Individual	Collective	without surpluses	with surpluses	Average daily model	Whole year model	
Without self-consumption	yes	yes	no	no	yes	yes	2 ²
Sizing optimisation	yes	yes	yes	yes	yes	yes	2 ³
Sizing and	yes	yes	yes	yes	no	yes	2 ²
- Optimisation	yes	yes	yes	yes	no	yes	2 ²
Scheduling	yes	yes	yes	yes	no	yes	2 ²
- Irrad. strategy	yes	yes	yes	yes	no	yes	2 ²
- Price strategy	yes	yes	yes	yes	no	yes	2 ²
Sharing	no	yes	yes	yes	no	yes	2
- Optimisation	no	yes	yes	yes	no	yes	2
- Pro-rata demand	no	yes	yes	yes	no	yes	2
- Equal	no	yes	yes	yes	no	yes	2

Ilustración 4. Gráfico de los diferentes estudios a realizar en el artículo. Fuente: [1]

Para cada uno de los casos el objetivo es conocer el coste de la energía. Para la optimización se ha utilizado un método heurístico en Matlab (similar a lo que se va a realizar en el proyecto). A continuación, se explica un poco el procedimiento a seguir.

1. Modelo de las cargas de la vivienda

Para las cargas de las viviendas, se ha utilizado un modelo típico para cada una de las viviendas. Se encuentran tres patrones en la curva de demanda:

- Una serie de pulsos de diferentes anchuras
- Cargas que se pueden aproximar como curvas periódicas
- Cargas constantes, normalmente bajas, distribuidas a lo largo de todo el día

Por tanto, el modelo propuesto contiene estas 3 funciones para modelar los diferentes tipos de cargas para cada vivienda.

2. Modelo del sistema de gestión de energía

Se realiza un balance de energía entre los elementos que constituyen el sistema híbrido (energías renovables, almacenamiento y carga de red). En la siguiente imagen [Figura 5] se muestra la estructura simple para la gestión de energía de este sistema híbrido.

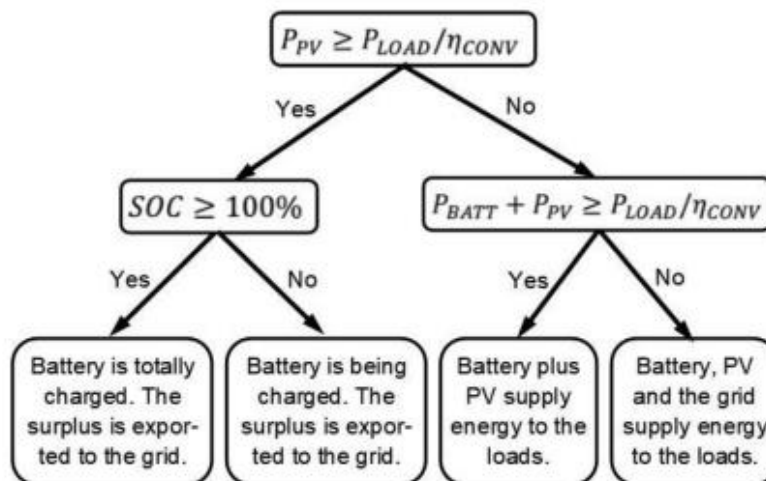


Fig. 2. Energy management scheme for a dwelling.

Ilustración 5. Gráfico de la gestión de la energía. Fuente: [1]

Este modelo, según la carga, la energía generada y el estado de carga de las baterías, gestiona de la forma más adecuada posible para satisfacer la demanda. La metodología a seguir del modelo de gestión de energía que se va a realizar en el proyecto es muy similar, aunque con algunas variaciones que se comentarán en su momento.

3. Evaluación económica

Para evaluar el proyecto, es necesario simular el sistema para un año. Una vez simulado para un año se puede extrapolar para la viabilidad del proyecto en un periodo más largo, teniendo en cuenta diferentes efectos como la inflación. En este caso se han tenido en cuenta una gran serie de costes para el cálculo, como por ejemplo los equipos, las reparaciones, costes de mantenimientos. En el estudio a realizar en el proyecto los costes se realizarán de una manera más sencilla.

• DATOS COMUNES EN TODOS LOS ESCENARIOS

A continuación, se describen todos los datos necesarios para realizar las simulaciones y optimizaciones. Todos los escenarios tienen los mismos datos de partida en cuanto a irradiancia, carga y precio de la energía.

1. Irradiancia

Los datos de irradiancia cubren un año. Se puede observar [Figura 6] que la zona central tiene más zonas de baja irradiancia, debido a que se corresponde con invierno, mientras que, en los extremos, que corresponde a verano tiene mayor irradiación. Esto indica que habrá más

generación fotovoltaica en verano que invierno tal y como era de esperar. En este caso los datos descargados provienen del software PVGIS. El perfil de irradiancia obtenido será similar al del proyecto, aunque dependerá en gran parte de la localización de la instalación.

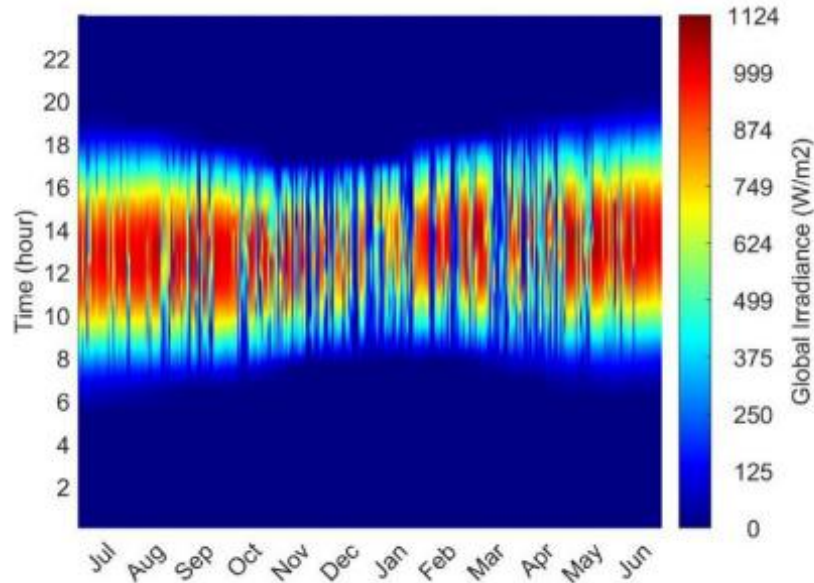


Ilustración 6. Irradiancia a lo largo de un año. Fuente: [1]

2. Datos de demanda de la vivienda

Se obtienen las cargas de cada casa a partir de la demanda media de cada una de las viviendas [Figura 7]. En la siguiente figura se muestra la demanda de una de las viviendas. Para el caso del autoconsumo colectivo se generan diferentes valores de demanda para cada una de las viviendas modificando ciertas variables aleatorias. Se observa que las horas donde menos demanda se produce es por la noche, principalmente debido a que los usuarios están durmiendo y los momentos donde más en torno a las 20h. También se observa que la demanda aumenta en los meses de invierno probablemente debido al aumento en el uso de la calefacción.

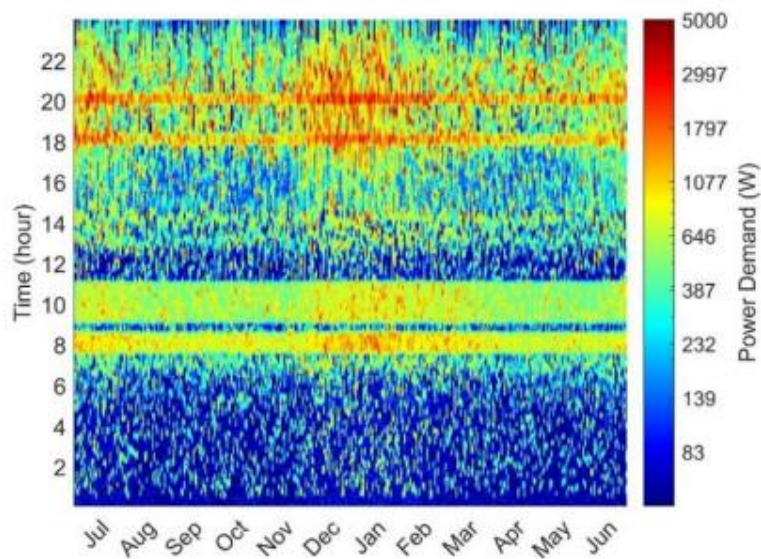


Ilustración 7. Demanda de una vivienda a lo largo de un año. Fuente: [1]

3. Precios de la energía eléctrica

En 2020 los precios eran mucho más bajos comparados con los últimos años. En 2021 salió una nueva tarifa con 3 precios diferentes de energía. La siguiente imagen [Figura 8] muestra el aumento en el precio y la volatilidad.

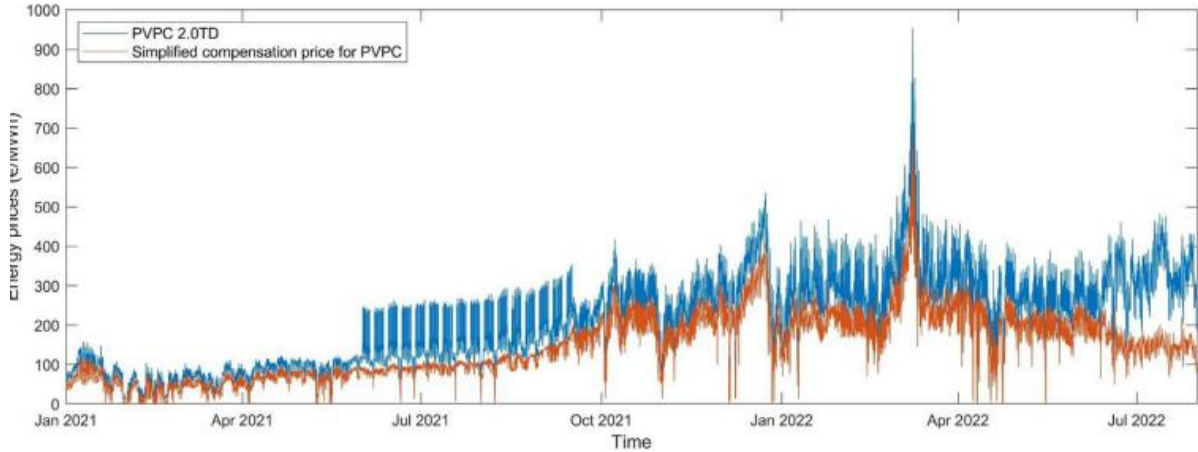


Fig. 6. PVPC 2.0TD and simplified compensation price of energy surpluses.

Ilustración 8. Gráfico precio energía eléctrica. Fuente: [1]

En el proyecto a realizar, a la hora de hacer las simulaciones, se van a partir de los mismos parámetros que en el estudio, aunque según la simulación que se este haciendo (depende de la ubicación y del consumo) los valores obtenidos serán diferentes.

- **RESULTADOS**

En cuanto a los resultados sin instalación de autoconsumo, se comparan utilizando los modelos mencionados, tanto el de valores de un día como el de valores de un año entero. En este caso el coste es únicamente el coste de energía por la cantidad de energía. Para este caso los resultados muestran que no hay apenas diferencia entre ambos modelos y los resultados son similares, por lo que lo óptimo sería utilizar el modelo diario debido al menor esfuerzo computacional.

En el caso de optimizar el tamaño de paneles y baterías, de la misma forma que antes, se obtienen los resultados de optimización tanto para el modelo de un día y para el del año entero. En este caso los resultados obtenidos diferencian significativamente entre ellos. En estos casos usar el modelo de día no es recomendable ya que se obtienen errores significativos.

En cuanto a los resultados obtenidos, sin tener en cuenta la modalidad de autoconsumo elegida, el coste de la energía presenta un menor valor para el caso colectivo que para el individual. En cuanto a las modalidades de autoconsumo, el autoconsumo con venta de excedente presenta un coste de la energía menor que sin venta del excedente.

En cuanto a la programación horaria de los electrodomésticos, se siguen dos estrategias. La primera consiste en utilizar los electrodomésticos en los momentos pico de generación de los paneles solares. La segunda es usarlos en los momentos en los que el precio de la red esté lo más barato posible, es decir en las horas valle. Echando un ojo a los resultados obtenidos se muestra que el coste de la energía para cualquier caso es muy similar y por tanto no es un método de ahorro significativo.

2.3. MODELOS DE OPTIMIZACIÓN (PSO) PARA PROBLEMA DE RUTAS

La optimización metaheurística ha ido ganando interés y popularidad entre los investigadores a lo largo de las últimas décadas. Los estudios en este campo avanzan rápidamente debido a que las técnicas necesarias para desarrollarse son introducidas matemáticamente y hacen estos métodos de optimización accesibles a todo el mundo. Tienen el objetivo de aumentar la efectividad de las soluciones que se buscan. Pueden ser tanto uniobjetivo como multiobjetivo y existen diferentes métodos entre los que destacan Particle Swarm Optimization (PSO) y Ant Colony Optimization (ACO) entre otros.

La optimización de rutas está dentro de los problemas habituales que puede tener un usuario. En el artículo mencionado se aplican este tipo de optimizaciones al problema de rutas, donde los usuarios no siempre conocen cual es el recorrido óptimo para llegar a su destino, ya que tienen en cuenta más cosas que solo la ruta más corta, sino que tiene en cuenta cosas como el tráfico la comodidad y la seguridad.

El problema de rutas, incluso en las optimizaciones mono-objetivo, es un problema de optimización complejo porque incluye un gran abanico de soluciones posibles. En este tipo de problemas, el espacio de búsqueda es muy amplio debido a numerosas soluciones, por tanto, no se pueden resolver con los métodos clásicos en un corto periodo de tiempo. Muchos estudios han tratado este tipo de algoritmos debido a la necesidad de resolver problemas complicados con un gran abanico de soluciones. Inspirados en la naturaleza, los algoritmos metaheurísticos basados en poblaciones y partículas inteligentes han sido desarrollados para resolver estos problemas de optimización complejos. El PSO está entre esos algoritmos y es muy fuerte en resolver este tipo de problemas.

A día de hoy, los sistemas de transporte público han hecho grandes progresos en las ciudades, por tanto, se han convertido en sistemas más complicados de analizar. La existencia del transporte multi modal puede mejorar la calidad del servicio y resolver los problemas de transporte urbano.

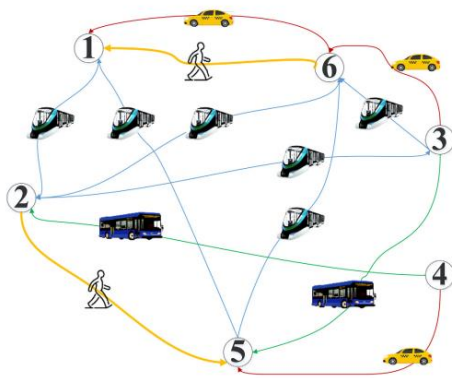
Se han realizado muchas investigaciones con el objetivo de resolver el problema de rutas. El caso más simple de este tipo de problemas es cuando la red tiene únicamente un objetivo y un modo de transporte. El algoritmo de PSO multiobjetivo fue desarrollado por primera vez por Coello, donde la salida de resultados te muestra una serie de soluciones y el usuario elige la más adecuada en función de sus prioridades.

- **METODOLOGÍA**

Se presenta el estudio en el que va a consistir el modelado de rutas multimodales en cuatro pasos. El primer paso la base de datos, donde se incluyen objetivos, modos y los criterios. El segundo paso la innovación, donde se proponen innovaciones con el objetivo de mejorar el algoritmo de optimización, en el tercer paso se incluye el problema de rutas multimodal y por último se evalúa y se analizan los resultados.

- **PROBLEMA DE RUTA MULTIMODAL**

Hoy en día, el transporte en áreas urbanas consiste en una red multimodal, donde la gente puede elegir diferentes medios de transporte para llegar a su destino. Con el objetivo de ayudar a la gente a encontrar su ruta optima la optimización de una red multimodal ha ido ganando enteros. Una ruta multimodal consiste en una serie de rutas monomodales y con una serie de constantes y condiciones de limitación del problema se puede obtener el resultado. A la hora de modelar este tipo de casos se suelen utilizar gráficos o matrices [Figura 9]. En este caso se utiliza un gráfico para representar las formas de llegar de un lugar a otro y a su vez la matriz que nos indica si hay conexiones de un lugar a otro y el método de llegar al sitio.



$$\begin{bmatrix} 0 & 1 & 2 & 0 & 1 & 3 \\ 1 & 0 & 1 & 4 & 3 & 1 \\ 2 & 1 & 0 & 0 & 4 & 1 \\ 0 & 4 & 0 & 0 & 2 & 0 \\ 1 & 3 & 4 & 2 & 0 & 1 \\ 3 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Ilustración 9. Ejemplo gráfico y matricial de problema de rutas multimodal. Fuente: [3]

- **METODO DE OPTIMIZACIÓN PSO**

El PSO se creó para resolver problemas uniobjetivos con variables continuas en un espacio de n dimensiones donde cada dimensión genera un algoritmo. La solución completa del algoritmo es un punto concreto de una de esas dimensiones del espacio de partículas generadas. Los cambios en los valores de la solución se obtienen moviéndose alrededor del espacio de partículas buscando el punto óptimo. Por tanto, inicialmente se generan soluciones aleatorias en cuanto a posición y velocidad de las partículas en el espacio de partículas y se va calculando el resultado. Cada movimiento de posición de una partícula por el espacio es un cambio de la solución final a un resultado mejor. Cada partícula guarda su mejor posición en cuanto a obtener la función objetivo se refiere, e informa y es informada del resto de valores de mejor posición para actualizarse en cada una de las iteraciones. Es decir, el cambio de posición de las partículas es influenciado por el mejor resultado de cada partícula y el mejor resultado global de las partículas en todo el espacio.

La optimización multiobjetivo de este tipo surge en 2002 donde un concepto llamado archivo se añade al algoritmo básico de PSO. La diferencia entre el uniobjetivo y la multiobjetivo está en la selección de una partícula líder y en la mejor solución global. En el caso multiobjetivo las soluciones no se pueden comparar linealmente.

2.4. CONCLUSIONES

En el primer estudio mencionado, se ha realizado una evaluación económica de las instalaciones de autoconsumo fotovoltaica en una zona rural, según la nueva legislación española. Se han estudiado y comparado los diferentes casos de autoconsumo en España con un escenario base sin instalación fotovoltaica. Se concluye que la legislación en zonas rurales no hace tan atractiva la instalación de paneles fotovoltaicos para autoconsumo, ya que se obtienen valores de retorno de la inversión muy elevados. Se observan 3 cosas interesantes. Primero, se concluye que la venta de excedente reduce notablemente el payback de 28 años a un rango de 16-22. Segundo, la subida de precios de la red resulta en una disminución notable del payback en instalaciones donde se vende el excedente según el precio del mercado, lo que lo hace mejor solución a pesar de los procedimientos administrativos. Por último, la solución más atractiva es la comunidad energética, donde un grupo de vecinos construyen un campo fotovoltaico con el mismo valor de instalación y con el objetivo de vender todo lo generado, sin beneficiarse de la legislación de autoconsumo.

A pesar de que la legislación de instalaciones para autoconsumo ha mejorado, el artículo muestra que sigue sin ser del todo económicamente rentable del todo a medio plazo, proponiéndose una serie de modificaciones para apoyar estas instalaciones de autoconsumo.

- Eliminar la limitación económica para los autoconsumos acogidos a compensación
- Establecer unos rangos para el precio fijo pagado en los autoconsumos acogidos a compensación basados en el precio actual del mercado.
- Reducir y formalizar las formalidades en los autoconsumos no acogidos a compensación.
- Eliminar las restricciones burocráticas a la hora de crear comunidades energéticas.

Aunque la legislación no favorezca del todo a estas instalaciones, debido a la situación de los altos precios del coste de la energía, hace el uso del autoconsumo, tanto individual como colectivo, asequible. Las dos conclusiones principales del segundo estudio nos indica que el autoconsumo con venta del excedente es mucho más rentable que sin venta del excedente y que las instalaciones colectivas son más rentables que las individuales independientemente del tipo de autoconsumo. En el proyecto se estudiará la eficiencia con el uso de baterías.

Se concluye que el modelo diario no es asequible con el autoconsumo, ya que se obtienen resultados sustancialmente diferentes que los obtenidos con el modelo anual y los sobreestima.

Y por último en cuanto a los métodos de optimización se concluye que el algoritmo multiobjetivo de enjambre de partículas, que se utiliza para obtener la ruta óptima en función de ciertos criterios en el último estudio, es muy útil para una gran cantidad de resolución de problemas, como el que se va a resolver durante el proyecto. También se ha comparado con otra serie de algoritmos de optimización concluyendo que la convergencia del algoritmo mencionado es mejor que en otros métodos de optimización y registra mayor reproducibilidad. En resumen, el algoritmo muestra un buen ratio de implementación y una buena eficiencia de convergencia para resolver problemas complejos y extensos.

3. METODOLOGÍA Y OBJETIVO

El objetivo y propósito final del proyecto es optimizar y mejorar la eficiencia de los sistemas de autoconsumo que integran paneles solares, aerogeneradores y baterías conectados a la red eléctrica, considerando un consumo específico y una ubicación determinada. La optimización se ha realizado utilizando el método de optimización de enjambre de partículas (PSO) implementado en Matlab. A continuación, se explica la metodología seguida para la realización del proyecto.

El proyecto se inicia con un estudio y análisis detallado del autoconsumo, concretamente en España, evaluando los diversos métodos existentes. Posteriormente, se lleva a cabo un estudio comparativo de los diferentes métodos de optimización, determinando que el PSO es el más adecuado para la resolución de este tipo de problemas.

A continuación, se seleccionan los componentes necesarios a utilizar en la simulación, es decir, los paneles, aerogeneradores y baterías que se van a utilizar en el problema, con sus características específicas para tenerlas en cuenta a la hora de simular. También se planifica la obtención de los datos necesarios para el desarrollo de la simulación, los cuales son los datos de consumo (LoadGenerator), la generación solar y eólica (NinjaRenewable) y los precios de compra de la red eléctrica.

El desarrollo del código de optimización parte de una base obtenida del software de YARPIZ. Se implementa el código para la obtención de datos requeridos y se desarrolla el algoritmo progresivamente. Inicialmente se realiza la simulación únicamente con paneles solares, y una vez funciona correctamente se incorporan los aerogeneradores. Finalmente se incorporan las baterías de forma que se tenga el código con el sistema de autoconsumo completo.

Una vez se tiene todo el código integrado y en funcionamiento, se desarrolla el algoritmo final con las condiciones necesarias para que el sistema responda adecuadamente a cada situación específica.

Con el código desarrollado, se realizan simulaciones bajo distintos escenarios con diferentes condiciones meteorológicas, que permitan evaluar el buen funcionamiento y desempeño del sistema y de la optimización. Finalmente se elaboran las conclusiones con los resultados obtenidos.

4. AUTOCONSUMO, AGENTES Y MODALIDADES EN ESPAÑA

4.1. AUTOCONSUMO CON ALMACENAMIENTO

Se conoce como una instalación de autoconsumo a todas aquellas instalaciones que producen energía eléctrica para su propio consumo, mediante generación solar o cualquier tipo de sistema de generación renovable y, en algunos casos, permitiendo el almacenamiento de la energía excedentaria. La gran parte del consumo eléctrico del hogar se cubre con la energía generada por la instalación de autoconsumo, dando lugar a un ahorro importante en la factura de la luz y generando una menor dependencia de la red, ante la volatilidad del precio de la electricidad, además de un menor consumo de energía fósil, ayudando a reducir de manera directa las emisiones de gases de efecto invernadero a la atmósfera.

El autoconsumo es una práctica viable, principalmente debido a la eliminación del impuesto al sol, tal y como se comentó en la revisión científica. Las instalaciones de autoconsumo están formadas por la parte de generación y el contador. Todo el sistema está conectado a la red eléctrica de forma que cuando la propia instalación genera energía se consume de ella, pero cuando no, la energía sigue viniendo de la red. De la misma forma, la energía excedente no utilizada puede ser inyectada a la red recibiendo una compensación por ello.

El uso de baterías en instalaciones de autoconsumo conectado a la red, a priori, puede parecer innecesario, pero realmente es una opción muy interesante. En este caso, cuando se quiera inyectar excedente a la red, pero también se quiera utilizar el almacenamiento en baterías, será necesario utilizar un controlador (inversor) que sea compatible con la inyección a red y con las baterías. Este controlador se encarga de gestionar y sincronizar tanto la producción solar, el consumo de la casa, la inyección de energía y la carga/descarga de las baterías. Las ventajas de la acumulación de energía son bastantes, ya que es la mejor forma de rentabilizar al máximo la instalación, y obtener mayor ahorro tanto energético como económico. El objetivo principal es maximizar el autoconsumo, de forma que se pueda obtener un menor valor de tiempo de retorno.

Teniendo en cuenta que en torno al 40% del consumo en los hogares se realiza por la noche cuando no se produce generación fotovoltaica, al instalar baterías, el excedente generado durante el día lo aprovechas por las noches, en vez de inyectarlo a la red por un bajo precio, para horas más tarde, comprarlo a la red por un precio mayor. Se puede concluir que el tiempo de amortización con baterías es mayor que sin las baterías, pero a largo plazo el ahorro es mucho mayor, ya que una vez recuperada la inversión el grado de autoconsumo de la instalación está en torno al 85% a diferencia del 40% que tendría la instalación sin baterías.

Las baterías a utilizar en estas instalaciones son baterías de litio. El estado de carga mínimo de cada una de ellas varía según el tipo específico de batería y la recomendación del fabricante. El SOC (Estado de carga) mínimo suele variar generalmente entre el 20-30%. Descargar las baterías más allá de esos valores puede afectar negativamente a su vida útil y rendimiento. Durante el desarrollo del proyecto se utilizará el valor recomendado por el fabricante de la batería seleccionada.

A continuación, se comentan los agentes involucrados en este tipo de instalaciones de autoconsumo y las modalidades de autoconsumo existentes en España.

4.2. AGENTES

En una instalación de autoconsumo intervienen diferentes actores, entre los cuales destacan:

- **Consumidor.** Es el consumidor de la energía eléctrica en un punto de suministro que tiene asociadas unas instalaciones de generación próximas.

- **Titular de la instalación de generación.** El que esté inscrito en los registros de autoconsumo. Sin excedentes, el titular es el consumidor. Con excedentes, el titular será el sujeto productor.
- **Productor.** El que produce la energía. El consumidor y productor pueden ser la misma persona.
- **Propietario de la instalación de generación.** Puede ser una persona diferente del consumidor y del productor.
- **Empresa comercializadora.** Realizan la venta de energía a los consumidores a través de los contratos de suministro. Las de mercado libre ofertan el suministro eléctrico al precio que libremente se pacte. Las de referencia están obligadas a ofertar el suministro al Precio Voluntario al Pequeño Consumidor.
- **Empresa distribuidora.** Propietarias de la red de distribución de energía eléctrica y prestan el servicio de distribución y son responsable de su gestión, operación y mantenimiento. Responsable de proporcionar a las empresas comercializadoras los datos necesarios para facturación.
- **Empresa instaladora.** Personas especializadas en instalaciones eléctricas.
- **Operador del sistema.** Es Red Eléctrica de España y tiene como función garantizar la continuidad y seguridad del suministro eléctrico.

4.3. MODALIDADES

En el Real Decreto 244/2019 se regulan las condiciones administrativas, técnicas y económicas del autoconsumo de energía eléctrica. Todo lo dispuesto en este Real Decreto es de obligatoria aplicación a las instalaciones de autoconsumo de energía eléctrica definidas en el artículo 9 de la ley 24/2013. También, el Real Decreto regula las modalidades de autoconsumo eléctrico definidas en el mismo artículo mencionado previamente.

Por tanto, queda definido que cualquier instalación de autoconsumo debe pertenecer a una de las siguientes modalidades que se mencionan a continuación:

- **Autoconsumo sin excedentes.** Son instalaciones de autoconsumo conectadas a la red de distribución o transporte, que disponen de un sistema anti vertido de tal forma que se impida la inyección del excedente eléctrico generado a la propia red.
- **Autoconsumo con excedentes.** Son instalaciones que, aparte de suministrar la energía eléctrica generada para autoconsumo, permiten la inyección del excedente eléctrico a las redes de transporte o de distribución. Dentro de este tipo de instalaciones se distinguen dos modalidades:
 - **Autoconsumo con excedentes acogida a compensación:** En estos casos, el consumidor utiliza la energía procedente de la instalación de autoconsumo, pudiendo comprar energía de la propia red siempre que sea necesario en los momentos en los que la generación no sea suficiente para satisfacer demanda. Del mismo modo, cuando la energía generada es mayor que la necesaria para el consumo, esta se puede inyectar a la red y posteriormente en la factura emitida, la comercializadora compensará el coste de la energía comprada con la inyectada. En ningún caso, el resultado de la factura podrá ser negativo. Para ello es necesario que se cumplan una serie de condiciones. La fuente de energía primaria debe ser renovable, deben ser potencias de instalación inferiores a 100 kW, existencia de un contrato de suministro con la comercializadora, existencia de un contrato de compensación de excedentes y no está sujeta a la percepción de un régimen retributivo.
 - **Autoconsumo con excedentes no acogida a compensación:** Son las instalaciones de autoconsumo que no cumplen con las condiciones mencionadas anteriormente

o que de forma voluntario no quieran acogerse a ellas. El excedente se vende directamente en el mercado eléctrico.

Para acogerse a cualquier tipo de modalidad, es necesario que el usuario disponga de un contrato de suministro de electricidad con la comercializadora, además de un contrato de acceso para las instalaciones de autoconsumo con la empresa distribuidora. En el contrato de suministro se suscribe el abastecimiento de la electricidad que va a tener el usuario, mientras que, en el contrato de acceso a la red, el consumidor contrata a la empresa distribuidora la conexión y el uso de las instalaciones de la compañía.

Independientemente de la modalidad de autoconsumo, este podrá calificarse como individual, si solo hay un consumidor asociado a la instalación, o colectivo, en el caso de que haya más de un usuario asociado. También podrán instalarse cualquier tipo de elemento de almacenamiento, como por ejemplo baterías.

Tanto para consumo individual como colectivo, las instalaciones podrán estar conectadas en la red interior de los consumidores, o conectadas a través de la red. A continuación, se muestran una serie de ejemplos de conexiones según el tipo de conexión [Figura 10]:

- Autoconsumo individual con conexión a red interior

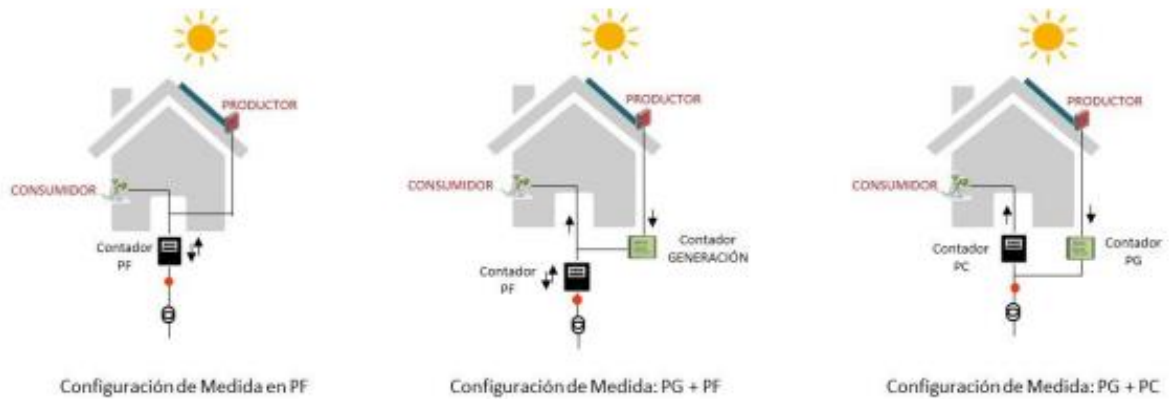


Ilustración 10. Conexión instalación autoconsumo a red interior. Fuente: [4]

- Autoconsumo colectivo con conexión a red interior

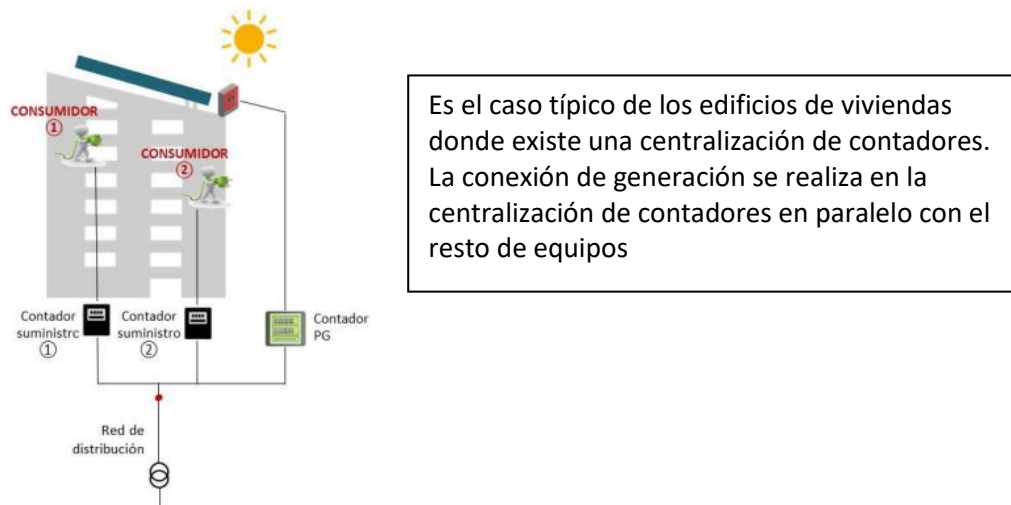
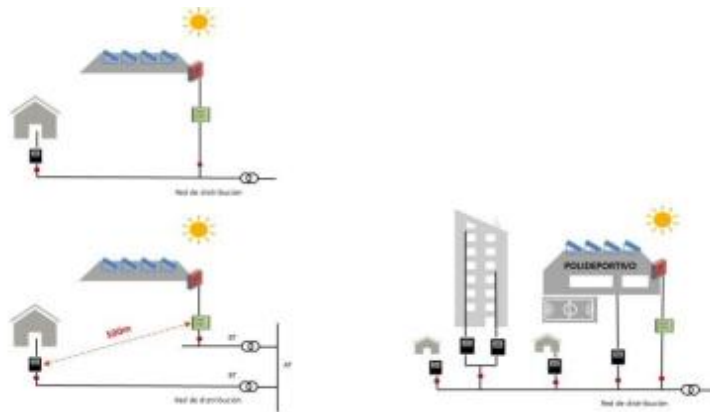


Ilustración 11. Conexión instalación colectiva a red interior. Fuente: [4]

- Autoconsumo individual y colectivo a través de red



Se conecta directamente a la red de distribución y se instala un contador para generación. Este tipo de casos no pueden acogerse a la modalidad de compensación de excedentes.

Ilustración 12. Esquema conexión autoconsumo individual y colectivo a través de red. Fuente: [4]

Por último, se describen los pasos a seguir para la tramitación de instalaciones de autoconsumo para cualquier tipo de modalidad. Según el tipo de autoconsumo y del tamaño de la instalación muchos pasos de los que mencionan a continuación no son de obligatoria aplicación.

Inicialmente se realiza el diseño de la instalación, mediante una memoria o proyecto técnico y a continuación se piden una serie de permisos (Permisos de acceso, autorizaciones ambientales, autorizaciones administrativas, licencia de obras). Una vez obtenidos todos los permisos necesarios se comienza la ejecución, donde se irán realizando inspecciones periódicas y se obtendrán finalmente los certificados de instalación y fin de obra. Una vez ejecutada la instalación se firman los contratos de acceso con la distribuidora y los de suministro con la comercializadora, además de obtener la licencia de actividad y firmar el contrato de compensación de excedentes. Para finalizar, se realiza la inscripción en los diferentes registros (autonómico, administrativo).

5. MODELOS DE OPTIMIZACIÓN

Se plantean y estudian diferentes métodos de optimización con el objetivo de encontrar cual es el modelo más adecuado para nuestro caso concreto. Estos modelos se obtienen la mayoría de las veces de observar lo que sucede en la naturaleza. Entre los diferentes modelos destacan Ant Colony Optimization, Simulated Annealing y Particle Swarm Optimization.

5.1. ANT COLONY OPTIMIZATION

Es un método de optimización, donde la fuente de inspiración es el comportamiento real de las hormigas a la hora de buscar comida. El método de búsqueda de comida de las hormigas consiste en explorar el área de forma aleatoria y tan pronto encuentran algo la llevan a su nido. Durante su regreso al nido, las hormigas dejan feromonas en el ambiente, de forma que sirva de guía para el resto de las hormigas y sean capaces de encontrar la comida por la ruta más corta posible. Esta característica mencionada se utiliza para resolver problemas de optimización en los que se necesiten mejorar los tiempos de cómputo.

Este algoritmo es un programa constructivo donde en cada iteración realizada, cada hormiga construye una solución al problema a través de un grafo. Cada arista del grafo tiene asociada cierta información. En cada iteración cada hormiga elige el camino de forma probabilística, y cada vez que una hormiga vuelve por un camino, esta actualiza estos valores probabilísticos de forma que la siguiente hormiga que vaya a buscar comida, tenga mayor probabilidad de ir por el camino más rápido.

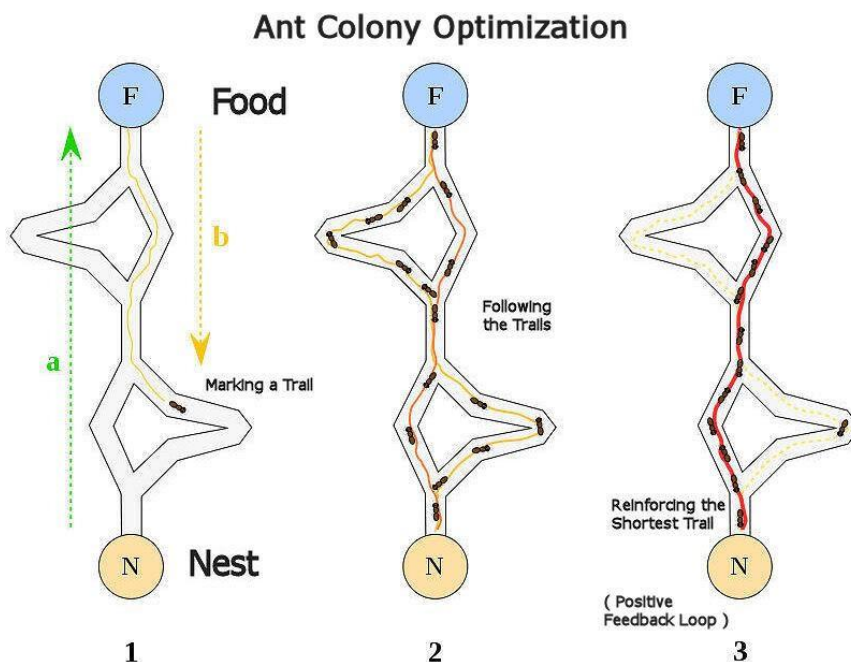


Ilustración 13. Ant Colony Optimization. Fuente: Google

En la anterior imagen [Figura 13] se observa lo mencionado anteriormente, donde inicialmente las hormigas están distribuidas por toda la zona en busca de comida, mientras que finalmente, con la ayuda de la información proporcionada por el resto de las hormigas, todas van por el camino más rápido.

No se ha seleccionado este método de optimización, ya que se considera un método más apropiado para otro tipo de problemas, que no el que se realiza en este proyecto, como puede ser por ejemplo un problema de optimización de rutas.

5.2. SIMULATED ANNEALING

El recocido simulado o simulated annealing es otro método conocido de optimización. El recocido es un tratamiento térmico que se utiliza en la metalurgia que altera las propiedades físicas y químicas de un material. En la gran mayoría de los casos, el proceso se realiza calentando el material y posteriormente dejándolo enfriar lentamente al aire.

El recocido simulado es un algoritmo basado en trayectorias que utiliza una metáfora termodinámica como método de decisión a la hora de aceptar soluciones. Consiste en evitar atascarse en óptimos locales, aceptando soluciones que a priori son peores, de forma controlada, mediante un parámetro Temperatura.

El algoritmo interpreta el enfriamiento lento como un lento descenso en la probabilidad de aceptar peores soluciones a medida que se explora el espacio de soluciones [Figura 14]. En este caso se aplica la filosofía de diversificar y explorar en las etapas iniciales del algoritmo e intensificar la búsqueda al final de este.

Se considera un algoritmo estocástico y sin memoria en el que, partiendo de una solución inicial, el algoritmo se ejecuta en varias iteraciones buscando movimientos que mejoren la solución objetivo, llegando incluso a aceptar como válidas soluciones peores, tal y como se ha mencionado anteriormente.

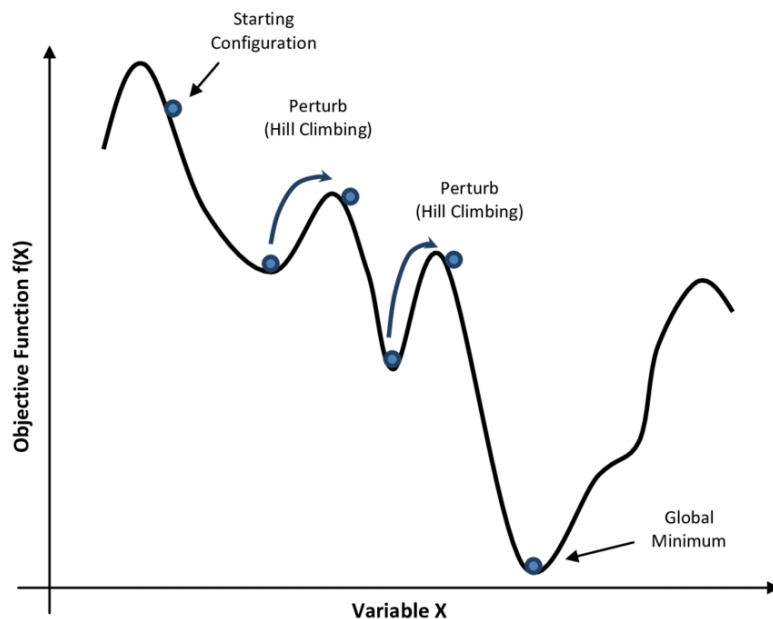


Ilustración 14. Método funcionamiento Simulated Annealing. Fuente: Youtube

Es un método similar al PSO, pero con una naturaleza distinta, por lo que se podría haber seleccionado como método de optimización para el problema a resolver de optimización de una instalación de autoconsumo.

5.3. PARTICLE SWARM OPTIMIZATION

Es un algoritmo de optimización inteligente basado en el paradigma de un enjambre, inspirado en el comportamiento de los animales en grupo, como puede ser un banco de peces. Este método de optimización puede ser aplicado a varios componentes de la ciencia y de la ingeniería. Es un método muy simple, pero con poderosos resultados. La base de este método de optimización es la comunicación y el aprendizaje entre partículas.

Este método fue introducido por Russel Eberhart y James Kennedy en 1995 mientras estudiaban un modelo para describir el comportamiento social de los animales en grupo. Mientras

realizaban sus estudios se dieron cuenta que el modelo era capaz de realizar diferentes tipos de optimizaciones.

La optimización por enjambre de partículas contiene una población de soluciones factibles, llamadas enjambre de soluciones. El algoritmo busca la solución factible al problema de optimización que se va a resolver. Cualquier partícula tiene una posición en el espacio de búsqueda de soluciones. Ese espacio es el conjunto de todas esas posibles soluciones y el objetivo es encontrar la mejor solución de todo ese espacio.

Para cada partícula, en cada iteración, la posición se denota mediante un vector, que pertenece al espacio de búsqueda de soluciones de la siguiente forma:

$$\text{Posición} = \vec{X}_i(t) \in x$$

De la misma forma, para cada una de las partículas, en cada iteración, se tiene un vector velocidad, que se encuentra en el mismo espacio que la posición, y describe el movimiento de la partícula en cuanto a sentido de dirección, distancia y tamaño

$$\text{Velocidad} = \vec{V}_i(t)$$

A continuación, se muestra un gráfico del principio de funcionamiento del método de optimización [Figura 15].

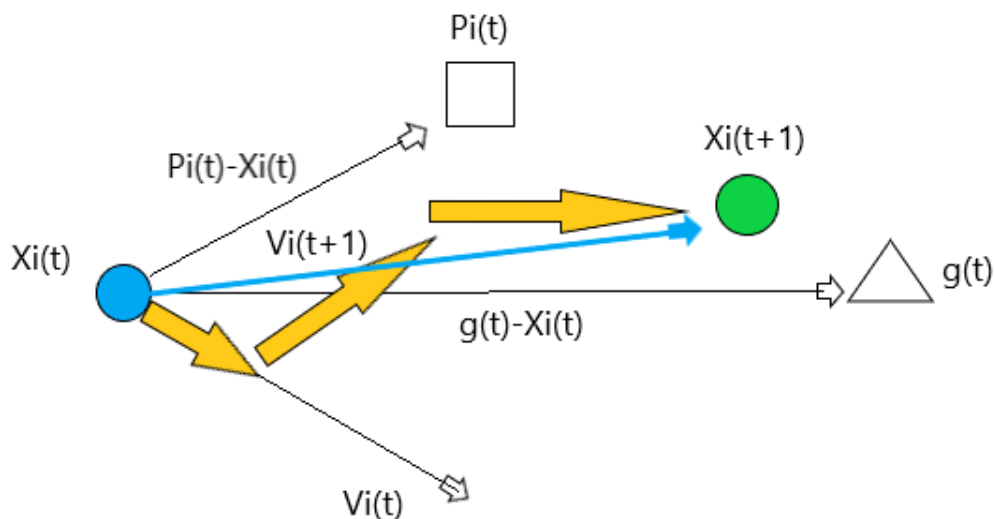


Ilustración 15. Principio de funcionamiento PSO.

Se parte de una partícula i en el espacio que se encuentra en una posición inicial $\vec{X}_i(t)$ que se mueve en una dirección y velocidad inicial $\vec{V}_i(t)$. Esto describe el movimiento simple de una partícula en el espacio.

Pero esta partícula es un miembro de todo un enjambre de partículas, y tal y como se ha mencionado anteriormente, estas partículas están continuamente interactuando y aprendiendo entre ellas, siguiendo una serie de reglas simples que les ayude a encontrar la mejor solución dentro del espacio de soluciones. Además de la posición y de la velocidad, cada partícula tiene un recuerdo de su propia mejor posición, es decir, de la posición en la que se ha obtenido mejor resultado, que se describe como $\vec{P}_i(t)$. También se tiene una mejor posición común de todas las partículas, que se describe como $g(t)$.

Para realizar el modelo matemático, se describe un vector desde la posición inicial de la partícula, hasta la mejor posición de la partícula, $\vec{P}_i(t) - \vec{X}_i(t)$, y un vector desde la posición inicial de la partícula, hasta la mejor posición global, $g(t) - \vec{X}_i(t)$.

En la siguiente iteración, la partícula se mueve hacia una nueva posición, pero utiliza los 3 vectores mencionados para llegar a ella. Inicialmente la partícula se mueve paralela al vector de velocidad ($\vec{V}_i(t)$), algo paralela al vector que conecta la posición de la partícula con la mejor posición de la partícula ($\vec{P}_i(t) - \vec{X}_i(t)$) y algo paralelo al vector que conecta la posición de la partícula con la mejor posición global.

Finalmente se obtiene la nueva posición de la partícula, denominada $\vec{X}_i(t+1)$ y su nueva velocidad que es un vector que conecta las 2 posiciones de la partícula, denominado $\vec{V}_i(t+1)$. Probablemente la nueva posición de partícula sea una mejor que la anterior, ya que utiliza para desplazarse, su mejor posición propia y la mejor posición global de todo el enjambre.

El modelo matemático puede describirse por tanto mediante la ecuación de actualización de posición de partículas y la ecuación de actualización de la velocidad:

$$X_i(t+1) = X_i(t) + V_i(t+1)$$
$$V_i(t+1) = WV_i(t) + C_1(P_i(t) - X_i(t)) + C_2(g(t) - X_i(t))$$

El primer componente de la ecuación de velocidad se conoce como término de inercia, el segundo componente de la ecuación de velocidad se conoce como componente cognitivo, asociado a la experiencia personal y el último de ellos se conoce como componente social, asociado a la mejor experiencia global.

6. CASOS DE ESTUDIO Y EQUIPOS SELECCIONADOS

6.1. FUNCIONAMIENTO, HIPÓTESIS Y ESCENARIOS

El objetivo principal es la optimización de sistemas de autoconsumo compuestos por paneles fotovoltaicos, aerogeneradores y baterías, conectados a la red. Un esquema similar del caso de estudio es el que se observa en la siguiente imagen [Figura 16].

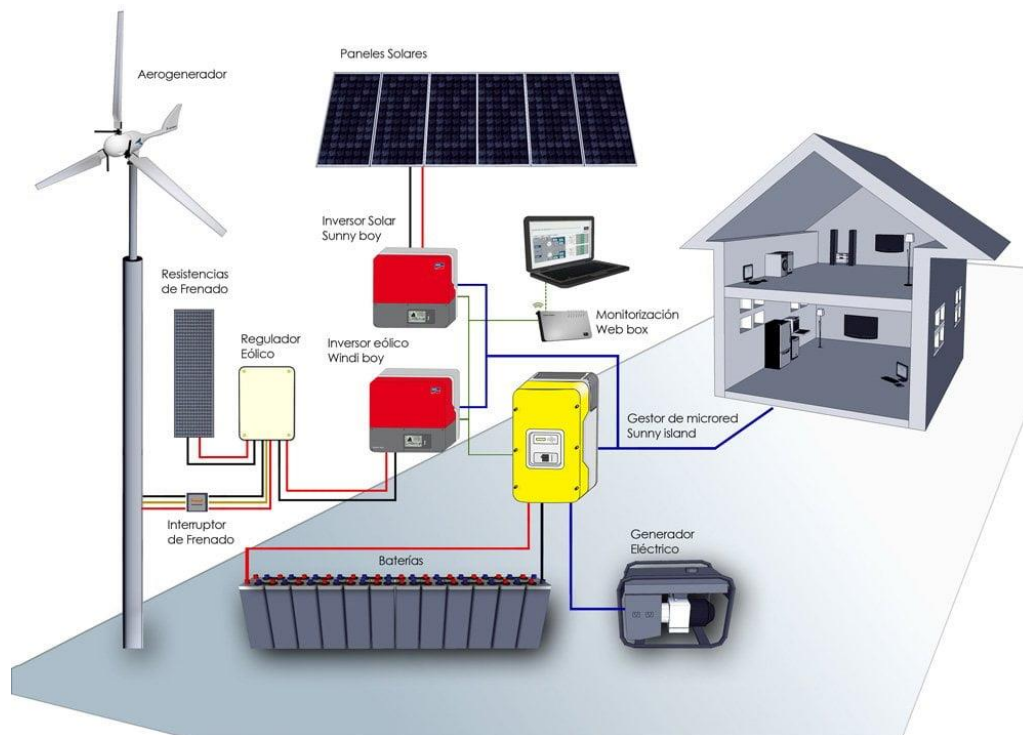


Ilustración 16. Instalación autoconsumo para una vivienda. Fuente: Google

La diferencia de la imagen anterior con el caso a estudiar es que en el caso de estudio la instalación está también conectada a la red en caso de apoyo por falta de generación o para recargar las baterías. La instalación está compuesta por paneles, aerogeneradores, baterías y los elementos necesarios para el correcto funcionamiento como los reguladores o inversores.

El objetivo es obtener los valores óptimos de los elementos mencionados anteriormente, de forma que, el coste de toda la instalación durante la vida útil de la misma sea el mínimo posible. La optimización se realiza conociendo la generación fotovoltaica por minuto durante un año, la generación eólica por minuto durante un año, el consumo por minuto durante un año y el precio de la energía eléctrica de la red por minuto durante un año. Siempre que se pueda el consumo se cubre con la energía generada, y en caso contrario se utilizan las baterías para cubrir el déficit. En el caso de que no se pueda cubrir la energía ni con la autogeneración, ni con las baterías, la energía necesaria se consume de la red al precio que se encuentre en ese momento. También se pretende recargar las baterías en momentos en los que el precio de la energía sea bajo, de forma que sea una instalación con el mayor autoconsumo posible y al menor coste.

Para ello se realizan una serie de suposiciones y aclaraciones:

- Para el cálculo del coste de la instalación no se tienen en cuenta elementos como inversores o reguladores.
- Para el cálculo del coste no se tienen en cuenta posibles roturas ni costes de mantenimiento

- No se tiene en cuenta el desgaste ni la pérdida de eficiencia por uso de ninguno de los elementos de la instalación
- Se supone una vida útil de la instalación de 20 años
- Las baterías inicialmente tienen una carga del 30 %
- Se considera la carga de baterías comprando a la red para precios inferiores a 0,07 euros/kWh
- Se considera el mismo precio de energía de la red para todas las simulaciones

Se realizan simulaciones en 4 localidades diferentes, Zaragoza, Oslo, Phoenix y Wellinton y para 3 consumos diferentes, uno de ellos de una persona soltera con trabajo, otro de una familia con un hijo y el último el de una familia numerosa. Las ciudades seleccionadas son de climas diferentes con el objetivo de ver la diferencia entre los resultados y poder comprobar que el programa de simulación funciona correctamente. A continuación, se muestran los valores de longitud y latitud de las ciudades sobre las cuales se va a realizar el estudio [Tabla 1].

	Latitud	Longitud
Wellinton	-47,28	174,77
Oslo	59,91	10,74
Phoenix	33,45	-112,07
Zaragoza	41,65	-0,88

Tabla 1. Ubicación ciudades a estudiar

Por tanto, se obtienen un total de 12 escenarios diferentes para realizar la simulación [Tabla 2].

	Soltero	Familia	Familia Numerosa
Wellinton	Escenario 1	Escenario 2	Escenario 3
Oslo	Escenario 4	Escenario 5	Escenario 6
Phoenix	Escenario 7	Escenario 8	Escenario 9
Zaragoza	Escenario 10	Escenario 11	Escenario 12

Tabla 2. Diferentes escenarios para estudiar

6.2. ELEMENTOS SELECCIONADOS

A continuación, se comentan los diferentes elementos seleccionados para hacer el estudio.

6.2.1. PANELES FOTOVOLTAICOS

El panel seleccionado para el estudio es el EM550-PH de la marca Tenste con las siguientes características [Tabla 3]:

Características	Valor
Tipo de módulo	550M Half cell Mono PERC
Máxima potencia	550 Wp
Corriente de potencia máxima	13,12 A
Voltaje de potencia máxima	41,95 V
Dimensiones	2,279x1,134x0,035 m
Precio	139,44 euros

Tabla 3. Características paneles fotovoltaicos

6.2.2. AEROGENERADORES

A la hora de instalar aerogeneradores en instalaciones domésticas, hay que tener en cuenta una serie de factores, como puede ser el ruido producido por la turbina o el impacto visual que pueda tener.

Los aerogeneradores de eje vertical son muy utilizados para instalaciones de autoconsumo, debido a que tienen un tamaño inferior a los aerogeneradores de eje horizontal y que el sonido producido por los mismos es prácticamente nulo. Las ventajas del uso de aerogeneradores verticales para autoconsumo respecto a los aerogeneradores horizontales son las siguientes:

- Más silenciosos
- Menor velocidad de giro y, por tanto, más seguros
- Funcionamiento a menores velocidades de viento
- Menores coste de instalación

El aerogenerador seleccionado finalmente para la realización del estudio es el modelo ATLAS 2.0, que debido a su forma compacta genera energía a valores menores de velocidad de viento. Este modelo tiene una gran desventaja, que es el bajo valor del coeficiente de potencia, debido a que tiene rotor Savonius, en este caso de 0,25 [Figura 17].

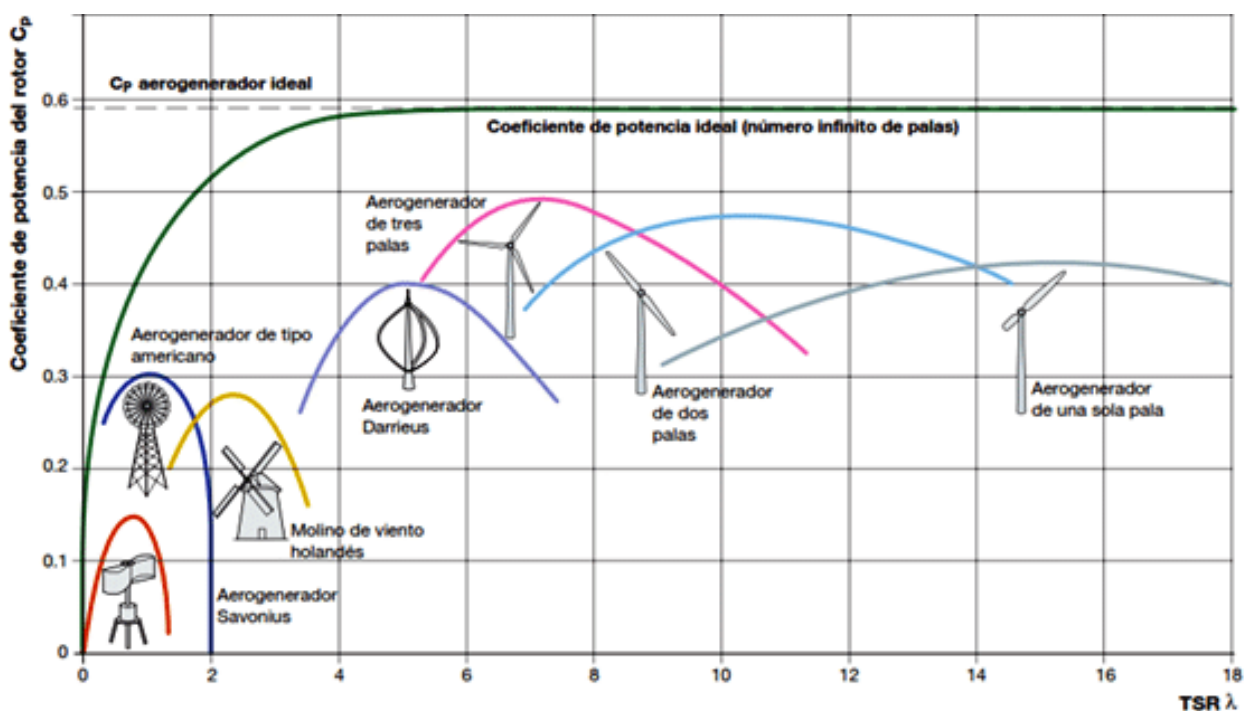


Ilustración 17. Coeficientes de potencia según aerogenerador

Las especificaciones del aerogenerador seleccionado son las siguientes [Tabla 4]:

Características	Valor
Modelo	TESUP ATLAS 2.0
Potencia Pico	2 kW
Numero de palas	3
Velocidad mínima de viento	2 m/s
Diámetro	0,6 m
Max RPM	650 m/s
Ruido	30 dB
Cp	0,25
Precio	690

Tabla 4. Características aerogenerador

6.2.3. BATERIAS

En cuanto a las baterías seleccionadas para la simulación, se han elegido baterías de litio ferroso (LiFePO₄) basado en celdas cuadradas de 2.560 Wh (24V 100Ah). Es una batería de alta densidad energética, con alta resistencia a las temperaturas y humedad, con protecciones integradas y una sencilla instalación.

Generalmente, las baterías, fallan prematuramente al funcionar durante un tiempo en modo de déficit, o si se deja muy descargada. Este caso, al ser batería de litio ferroso, no es necesario estar completamente cargada. Es una batería con una alta eficiencia, en donde la eficiencia energética del ciclo completo puede llegar a ser del 92 %, en comparación a una de plomo-acido que está en torno al 80%. Debido a esta alta eficiencia y a una alta densidad energética, se consigue ahorrar un alto porcentaje en tamaño y peso de las baterías. Las baterías, en el momento de envío, están cargadas al 30%, parámetro que se tomará como valor inicial, a la hora de realizar el código de simulación en Matlab.

La ficha técnica de las baterías se muestra en la siguiente tabla [Tabla 5]:

Características	Valor
Referencia	NCL-LFPO-24100
Composición	LiFePO ₄
Voltaje nominal	25.6V
Resistencia interna	< 30mΩ
Capacidad nominal	50Ah
Energía nominal	2.560Ah
Corriente max Carga	60A (recomendado) 100A (Máximo)
Corriente max Descarga	60A (recomendado) 100A (Máximo)
Voltaje carga	25.6-29.2 V
Voltaje descarga	20-29.2 V
Dimensiones	482x400x134 mm
Peso	28 kg
Precio	490 euros

Tabla 5. Características baterías

7. DESCARGA DE DATOS PARA ALGORITMO

Para llevar a cabo la optimización del sistema, es indispensable recopilar una variedad de datos e información que posibiliten la realización de las simulaciones. Estos datos incluyen los valores de generación solar y eólica, los valores de consumo y los precios de la red eléctrica para un año completo.

El algoritmo está pensado para simularse durante todos los minutos de un año completo, lo que significa, que se requiere obtener toda esta información por minuto, es decir, hay que conocer la generación cada minuto, el consumo cada minuto y el coste de la red cada minuto. A continuación, se detalla el procedimiento para obtener los datos.

7.1. VALORES DE GENERACIÓN

En un sistema de autoconsumo con paneles solares y aerogeneradores, los usuarios son capaces de generar su propia electricidad, reduciendo la dependencia con la red eléctrica. Los valores de generación dependen de varios factores, como, por ejemplo, la ubicación de la instalación, los elementos seleccionados o la potencia de la instalación. Son valores imprescindibles para el desarrollo del algoritmo.

Para obtener estos valores se utiliza NinjaRenewable. Es un software online que, dados unos valores de entrada, como la ubicación o la potencia de instalación, el programa descarga los valores de generación. A continuación, se muestra una imagen del Software [Figura 18]. Como se observa, se pueden obtener valores tanto de generación eólica como solar.

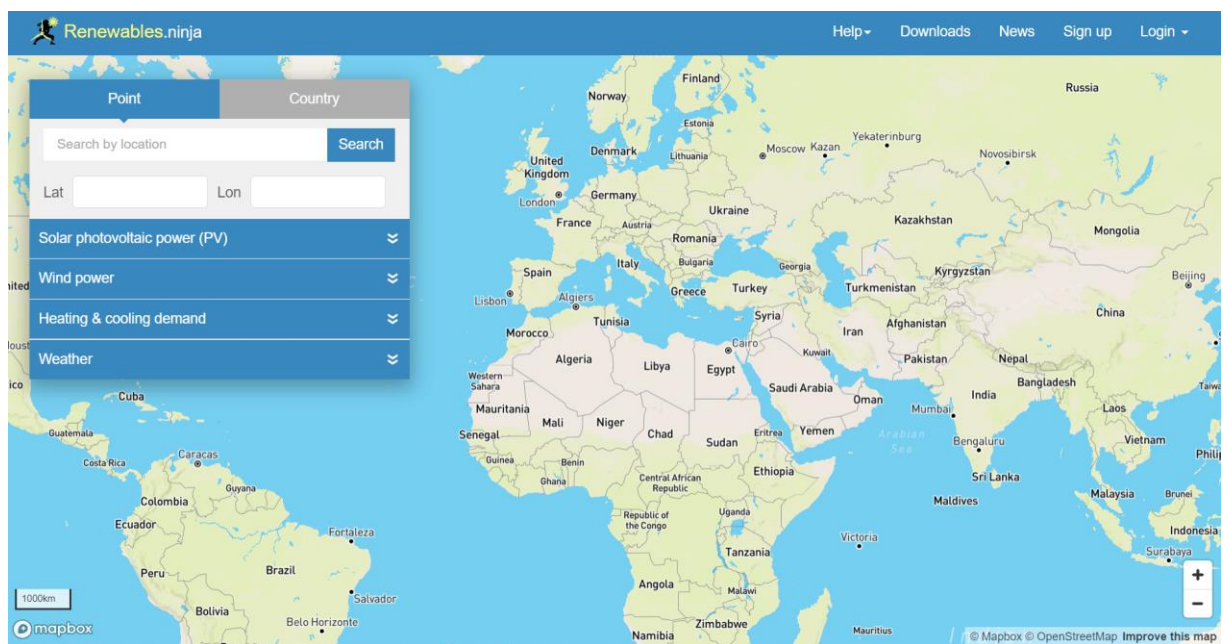


Ilustración 18. NinjaRenewables. Fuente: GOOGLE.

En este caso, se ha desarrollado un código de Matlab que descarga los datos de la web [Anexo 1]. La descarga de datos de generación solar y eólica se ha realizado de la misma forma, con una pequeña diferencia que se comenta a continuación.

7.1.1. GENERACIÓN SOLAR

La producción energética fotovoltaica se obtiene a partir del software online Ninja Renewable, en el cual, introduciendo una serie de valores como la ubicación, potencia de la instalación, porcentaje de pérdidas o inclinación, el programa te devuelve la generación horaria durante el periodo de tiempo que se le pida.

En este caso, los datos se descargan a través de un código de Matlab que conecta con la web y según los datos iniciales introducidos, se descargan automáticamente los valores de generación. Uno de los parámetros de entrada que hay que introducir en el código para su posterior descarga de datos es la potencia de la instalación, y para ello es necesario conocer la potencia y características de los paneles a instalar, de forma que según el número de paneles que seleccione el programa, y conocido el valor de la potencia, se puedan descargar correctamente los valores de generación fotovoltaica.

Desarrollando el código en Matlab, finalmente se obtiene la generación solar para cada minuto del año.

7.1.2. GENERACIÓN EÓLICA

En cuanto a la energía eólica, se descarga del mismo modo que la fotovoltaica, usando el software Ninja Renewable, con la diferencia de que en este caso no se descarga el valor de energía directamente, sino que se descargan los valores del viento horarios de la zona seleccionada. Esto se debe a que en el software Ninja Renewable no hay disponible ningún aerogenerador para autoconsumo, entonces hay que calcular la energía generada a partir de los datos de viento. Los datos de viento, de la misma forma que antes, se descargan utilizando un código de Matlab que conecta con el servidor de Ninja para los parámetros de entrada que se hayan seleccionado.

Una vez conocidos los valores por minuto de viento y el aerogenerador seleccionado, la energía eólica producida se calcula mediante una serie de fórmulas que se muestran a continuación. Todos los cálculos se han realizado también a través de Matlab.

La potencia generada se calcula como el mínimo entre el punto de corte de la curva de potencia del aerogenerador con la curva de velocidad de viento y la potencia teórica. Los datos de potencia del aerogenerador se obtienen de la hoja del fabricante.

Para el cálculo de la potencia teórica es necesario hacer unos cálculos previos, ya que la velocidad del viento varía en función de la altura, y por tanto hay que ajustarla a los valores de la altura a la que se encuentre el rotor. La fórmula para ajustar las velocidades del viento es la siguiente:

$$velocidad_{ajustada} = velocidad_{viento} * \left(\frac{altura_{rotor}}{altura_{medida}} \right) * alfa$$

La velocidad del viento se ha obtenido con NinjaRenewable para una altura dada, la altura del rotor se considera 15m y el parámetro alfa es el coeficiente de rugosidad que se considera de 0,2.

A continuación, se calcula el área barrida por el aerogenerador del rotor Savonius, que es el aerogenerador seleccionado, sabiendo que el diámetro del rotor elegido es de 0,6 metros. La fórmula es la siguiente:

$$Area_{barrida} = altura_{rotor} * diametro_{rotor}$$

Conocida toda esa información, y sabiendo que el coeficiente de potencia típico de un rotor Savonius es de 0,25 tal y como ya se había comentado, se calcula la potencia teórica generada con la siguiente fórmula:

$$Potencia\ Teórica = \frac{1}{2} * \rho * Area * v^3 * Cp$$

De esta forma, una vez conocida la potencia teórica para cada minuto de un año completo, y las curvas de potencia del aerogenerador en función de la velocidad del viento, se puede conocer la generación eólica anual para cada minuto del año.

7.2. VALORES DE CONSUMO

Para obtener los datos de consumo, se utiliza la aplicación LoadProfileGenerator. Es una herramienta que permite la creación de diferentes perfiles de carga detallados para el consumo eléctrico de hogares, oficinas u otro tipo de edificios. La aplicación genera perfiles de consumo, a partir de datos de comportamiento y diferentes patrones de uso, capaces de representar la demanda real de la electricidad en diferentes momentos del día y del año.

Estos perfiles son muy útiles para realizar simulaciones, ya que facilitan información sobre patrones de consumo energético, los cuales pueden ser utilizados en estudios proporcionando datos para modelar el comportamiento de los consumidores y sus efectos en la demanda energética.

Es una herramienta con un comportamiento muy realista, ya que utiliza modelos detallados del comportamiento humano generando perfiles de carga precisos. Permite la personalización de los perfiles de forma que se puedan adaptar los perfiles de consumo a las necesidades. Tiene una interfaz amigable y sencilla de usar.

Para este estudio, se han descargado 3 tipos diferentes de perfiles de consumo muy diferenciados para realizar las simulaciones. Se ha descargado el perfil de consumo de una persona soltera, que viva sola, y, por tanto, con un bajo consumo energético. Se ha descargado también el perfil de consumo de una familia con un hijo y por último el perfil de consumo de una familia numerosa. El objetivo de realizar las simulaciones con unos perfiles de carga tan diferenciados es comprobar el buen funcionamiento del código, ya que, para perfiles de consumo tan diferenciados, se tienen que obtener resultados muy dispares, y poder demostrar la utilidad del programa para cualquier tipo de situación y escenario.

En resumen, se descargan los 3 perfiles de consumo en Excel con información de consumo de todos los minutos de un año completo. El código de Matlab obtiene la información de consumo leyéndola de Excel [Figura 19], según el tipo de simulación que se esta realizando en cada caso.

```
%% CONSUMO POR MINUTO DURANTE UN AÑO NATURAL

file = 'Consumo familia numerosa.csv';           % Se leen los
datos de consumo previamente descargados de un archivo CSV
data = readtable(file, 'VariableNamingRule', 'preserve');
Tiempo = data.Timestep;
Hora = data.Time;                               % Se lee el día y
hora del consumo
consumo = data.Valor;                           % Se lee el valor
del consumo
```

Ilustración 19. Código Matlab valores de consumo.

Los pasos para la descarga de datos se explican mas a fondo en el Anexo [Anexo 2].

7.3. VALORES DE PRECIO DE RED

Otro de los datos necesarios para el correcto desarrollo del programa es el valor del precio de compra de energía eléctrica a la red. Es un parámetro necesario, ya que cuando haya déficit energético con la generación renovable será necesario obtener la diferencia inyectándola de la red para poder satisfacer la demanda.

Para la descarga de datos se utiliza la web ESIOS (Sistema de Información del Operador del Sistema Eléctrico). ESIOS es una plataforma gestionada por la Red Eléctrica Española. Se proporciona datos e información en tiempo real sobre el sistema eléctrico español, incluyendo precios eléctricos, demanda, generación y otros indicadores relacionados. ESIOS permite el

acceso a históricos de forma que se descarga el precio para cada minuto del año 2022 en formato Excel [Anexo 3].

Una vez se ha descargado el archivo Excel con la información necesaria, se genera un código de Matlab que lea los valores del Excel [Figura 20], de la misma forma que se hace con los valores de consumo. Se utilizan los mismos valores de precio de compra de red para cada uno de los escenarios de simulación.

Una vez seleccionados los elementos a utilizar con sus características y definida la forma de obtener los valores necesarios para poder realizar las simulaciones, se comenta el desarrollo del código del programa.

```
%% PRECIO DE RED POR MINUTO DURANTE UN AÑO NATURAL

file = 'Precio Red Horario.csv'; % Se leen los datos
de precio previamente descargados en un archivo CSV
data = readtable(file, 'VariableNamingRule', 'preserve');
HoraPrecio = data.datetime; % Se lee el día y
hora del precio
Precio = data.value; % Se lee el valor
del precio de red

% Se convierten los valores horarios obtenidos en valores por minuto
precio_minuto = zeros(525600,1);
precio_minuto = repmat(Precio(1),1,60);
for k=2:8760
    precio_minuto = [precio_minuto repmat(Precio(k),1,60)];
end
precioRed = reshape(precio_minuto, [525600, 1]); % Vector final de
precio por minuto durante un año natural
```

Ilustración 20. Código Matlab valores precio de red.

8. DESARROLLO DEL PROGRAMA DE SIMULACIÓN

8.1. FUNCIÓN OBJETIVO

Como se ha comentado anteriormente, el objetivo del algoritmo es minimizar la función coste de la instalación, de forma que se obtenga el número óptimo de paneles, aerogeneradores y baterías, para que sea lo más eficiente tanto energéticamente como económicamente durante la vida útil del sistema, es decir, lo más barato posible.

Para ello, lo más importante es definir la función objetivo que se quiere minimizar durante la optimización. Esta función objetivo es la función coste. Como se ha comentado anteriormente en las hipótesis, la función coste se calcula de forma simplificada, es decir, no se tienen en cuenta ni operaciones de mantenimiento y sustitución, ni actualización del valor del dinero con el paso del tiempo.

La función coste se calcula multiplicando el número de cada uno de los elementos, del sistema, previamente seleccionados, por su precio, y el coste de la energía eléctrica comprada a la red multiplicado por la vida útil del sistema, que en este caso se considera de 20 años. La función coste, por tanto, queda de la siguiente forma [Figura 18].

```
function z = COSTE_codigoFinal(x, compra)

paneles=x(1);
baterias=x(2);
aerogeneradores=x(3);

z = 139.44*paneles+490*baterias+690*aerogeneradores+20*compra;
```

Ilustración 21. Función coste MATLAB

Se define la función coste, como una función con dos parámetros de entrada, el vector “x” donde se define la cantidad de cada uno de los elementos del sistema (paneles, aerogeneradores o baterías) y el parámetro “compra”, que indica el valor del coste de la energía eléctrica comprada de la red. De esta forma, conocidos ambos parámetros el sistema es capaz de calcular la función de coste Z.

Una vez definida la función coste, se desarrolla el código de optimización.

8.2. CODIGO DE OPTIMIZACIÓN

8.2.1. VARIABLES DEL PROBLEMA

El código de optimización, como se ha comentado, está basado en el método de optimización de enjambre de partículas (PSO). Una vez definida la función de coste a optimizar y las variables necesarias a utilizar en el programa (generación, consumo y precio red) se define el problema y los parámetros necesarios para realizar la optimización. La base del código desarrollado ha sido obtenida de la web Yarpiz, pero ha sido modificada para adaptarla a este problema en concreto.

En cuanto al código de optimización lo primero que se hace es definir las variables del problema [Figura 22]. Para ello se crea una estructura llamada “problem” a la cual se le asocia la siguiente información:

- Función coste. Se asocia la función de coste previamente creada con el problema
- Variables de estado. Se asocia el número de variables de estado del problema. Estas son las variables que se quieren optimizar, que en este caso son 3, los paneles solares, los aerogeneradores y las baterías.

- Valores máximos y mínimos. Se limitan los valores máximos y mínimos de las variables de estado del problema. En este caso, al ser autoconsumo en hogares, se limita a un valor máximo de 20 unidades de cada uno de los elementos y a un valor mínimo de 0.

```
%% DEFINICION DEL PROBLEMA. Se define la funcion de coste, numero de variables y
valor máximo y mínimo de las variables

problem.CostFunction = @(x, compra) COSTE_codigoFinal(x, compra); % Cost Function
problem.nVar = 3; % Number of
Unknown (Decision) Variables
problem.VarMin = 0; % Lower Bound of
Decision Variables
problem.VarMax = 20; % Upper Bound of
Decision Variables
```

Ilustración 22. Definición de las variables del problema.

8.2.2. PARAMETROS DE SIMULACIÓN

A continuación, se definen las condiciones de la optimización de la simulación [Figura 23] y, de la misma forma, se crea la estructura “params” para definir los parámetros de la optimización. A la estructura se le asocia la siguiente información:

- Número de iteraciones. Se define el número de iteraciones que se van a realizar. En cada una de las iteraciones se reajustan los valores de las variables de estado, de forma que, tras cada una de las iteraciones los valores están cada vez más cerca de sus valores óptimos. Con muchas iteraciones, se puede asegurar que el problema converge a la solución óptima.
- Tamaño de población. Se define la población de partículas. Cada partícula tiene un valor de la solución en cada una de las iteraciones. Cuantas más partículas haya, antes se converge en la solución final, ya que las partículas interactúan entre ellas para poder obtener la mejor solución final.
- Parámetros optimización PSO. Se definen una serie de parámetros necesarios a partir de los cuales se actualizan los valores de las variables de estado entre iteración e iteración.

```
%% PARAMETROS DEL PSO. Se definen los parametros necesarios para la optimizacion

params.MaxIt = 7; % Maximum Number of Iterations
params.nPop = 6; % Population Size (Swarm Size)
params.w = 1; % Inertia Coefficient
params.wdamp = 0.99; % Damping Ratio of Inertia Coefficient
params.c1 = 2; % Personal Acceleration Coefficient
params.c2 = 2; % Social Acceleration Coefficient
params.ShowIterInfo = true; % Flag for Showing Iteration Informatin
```

Ilustración 23. Definición parámetros del problema.

8.2.3. PRIMERA ITERACIÓN

Una vez definidas las condiciones del problema y los parámetros de optimización, se comienza con la simulación. La primera iteración es diferente al resto, ya que los valores de las variables de estado se definen de forma aleatoria [Figura 24].

```

for i=1:nPop

    particle(i).Position = round(unifrnd(VarMin, VarMax, VarSize)); % Se
generan valores aleatorios reales de las variables de estado

```

Ilustración 24. Generación de variables de estado de forma aleatoria.

Una vez generadas las variables de estado para cada una de las partículas de la población, se aplica el algoritmo para todas ellas de forma que se calcula la función coste siguiendo una serie de condiciones según los valores de generación, consumo y precio de la red. También se almacenan las variables de estado que han generado el mejor resultado de la función coste.

```

    % Initialize Velocity (Dar valor inicial 0 de velocidad a todas las partículas)
    particle(i).Velocity = zeros(VarSize);

    % Evaluation (Calcular la función coste para los valores iniciales de partícula
introduciendo las 3 variables de estado y el coste total de compra)
    particle(i).Cost = CostFunction(particle(i).Position, SumacosteRed(i));

    % Update the Personal Best (Se crea la mejor posición y el mejor coste de cada
partícula, en este caso las iniciales)
    particle(i).Best.Position = particle(i).Position;
    particle(i).Best.Cost = particle(i).Cost;

    % Update Global Best (Se actualiza el valor de coste y posición de la mejor
partícula
    if particle(i).Best.Cost < GlobalBest.Cost
        GlobalBest = particle(i).Best;
    end

end

% Array to Hold Best Cost Value on Each Iteration
BestCosts = zeros(MaxIt, 1);

```

Ilustración 25. Actualización mejor posición y mejor coste.

8.2.4. RESTO DE ITERACIONES

La diferencia con la primera iteración es que las variables de estado nuevas de la siguiente iteración no se generan actualmente. Aquí se observa el fundamento de la optimización de enjambre de partículas, donde los siguientes valores de las variables de estado se calculan a partir de los mejores resultados obtenidos en la iteración anterior.

Cada una de las partículas recalcula su posición a partir de los mejores valores propios y a partir del mejor resultado global de todas las partículas. Una vez obtenidos los nuevos valores de las variables de estado se vuelve a ejecutar el algoritmo para cada una de las partículas y de la misma forma que antes se vuelven a almacenar los mejores resultados en posición y coste.

El procedimiento es el mismo para cada una de las iteraciones, obteniendo al final de ellas, los resultados óptimos para el problema estudiado.

Finalmente, se estudian los resultados obtenidos, calculando el coste final y el ahorro que supone utilizar un sistema de autoconsumo en vez de comprar únicamente la energía de la red.

```

        for i=1:nPop                                % Para cada partícula en cada
iteracion                                         iteración

        % Update Velocity (Se actualizan los valores de la velocidad en función
de los mejores resultados hasta el momento
        particle(i).Velocity = w*particle(i).Velocity ...
        + c1*rand(VarSize).*(particle(i).Best.Position - particle(i).
Position) ...
        + c2*rand(VarSize).*(GlobalBest.Position - particle(i).Position);

        % Apply Velocity Limits (Se aplican los límites máximos y mínimos de
velocidad
        particle(i).Velocity = max(particle(i).Velocity, MinVelocity);
        particle(i).Velocity = min(particle(i).Velocity, MaxVelocity);

        % Update Position (Se actualizan las variables de estado de la
partícula en función de los mejores resultados hasta el momento
        particle(i).Position = round(particle(i).Position + particle(i).
Velocity);

        % Apply Lower and Upper Bound Limits (Se aplican los valores máximos y
mínimos de las variables de estado)
        particle(i).Position = max(particle(i).Position, VarMin);
        particle(i).Position = min(particle(i).Position, VarMax);

```

Ilustración 26. Actualización de los valores de las variables de estado.

8.3. ALGORITMO

Una vez se tiene toda la información necesaria para cada minuto del año (Consumo, generación y precio de red) se aplica el algoritmo. El algoritmo se aplica para cada minuto del año, y según los valores que se tengan en cada minuto se realiza una cosa u otra. En el flujograma [Figura 27] se muestra el algoritmo de una forma más visual.

Para cada minuto del año, lo primero que se compara es la generación con el consumo. En el caso de que la generación, tanto eólica como solar, sea mayor que el consumo, la demanda se satisface completamente con la generación renovable, sin necesidad de utilizar baterías ni comprar energía a la red. A continuación, se calcula la energía excedente una vez se satisface la demanda. Esa energía excedentaria se utiliza para recargar las baterías. El valor del estado de carga de las baterías se actualiza.

En el caso de que el consumo sea mayor que la generación, no se puede satisfacer la demanda únicamente con la generación renovable, de forma que hay que utilizar las baterías y si fuese necesario, comprar energía eléctrica a la red. Se calcula el déficit energético y se satisface con la energía de las baterías. En el caso de que no haya suficiente energía en las baterías para satisfacer la demanda se procede a comprar la diferencia a la red eléctrica. El valor del estado de carga de las baterías se actualiza.

Por último, para cada minuto del año, se revisa el precio de compra de energía a la red. Si el precio está por debajo de un valor límite, y además las baterías no están llenas, se compra energía a la red a un precio barato para llenar un porcentaje las baterías. De esa forma se compra energía eléctrica a bajo precio, aunque en ese momento no sea necesario, para posteriormente utilizarla cuando no haya suficiente generación renovable.

Con toda la información y aplicando el algoritmo, finalmente se obtiene el número óptimo de elementos a colocar en el sistema de autoconsumo y el coste final del sistema para toda su vida útil.

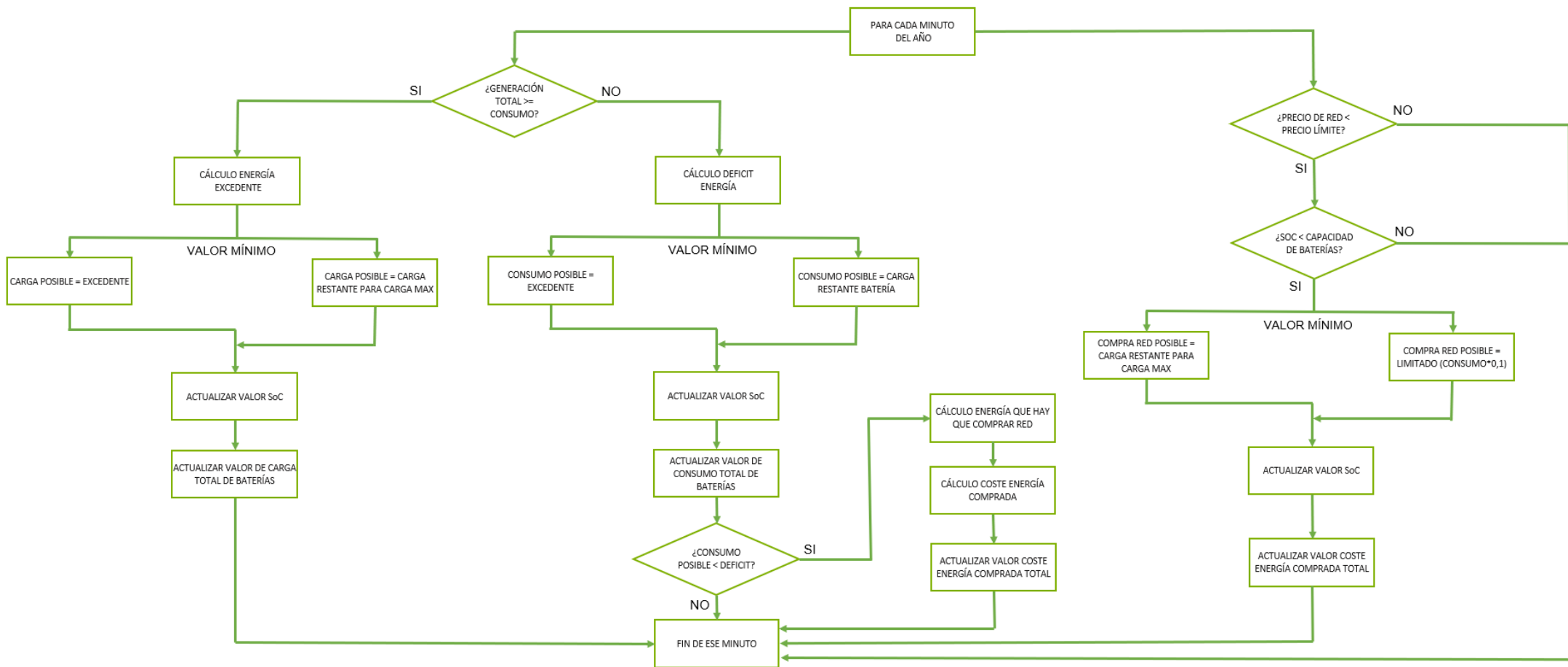


Ilustración 27. Flujograma algoritmo

9. RESULTADOS SIMULACIONES

Una vez explicado el código, se procede a realizar las simulaciones pertinentes para los diferentes escenarios mencionados anteriormente, con el objetivo de comprobar si funciona correctamente.

9.1. NÚMERO ÓPTIMO DE ELEMENTOS

Para cada uno de los escenarios, se realizan las simulaciones con el número de iteraciones necesarias para asegurar que el resultado converge al resultado óptimo, tal y como se muestra en la siguiente imagen [Figura 28]. Son gráficas que muestran el valor de la función coste en función del número de iteraciones que se realizan de la simulación. Se observa que para cada iteración se obtienen valores de coste cada vez menores, hasta llegar a un punto en el que la gráfica se mantiene en un valor constante, lo que hace indicar que se ha obtenido el resultado óptimo para ese escenario.

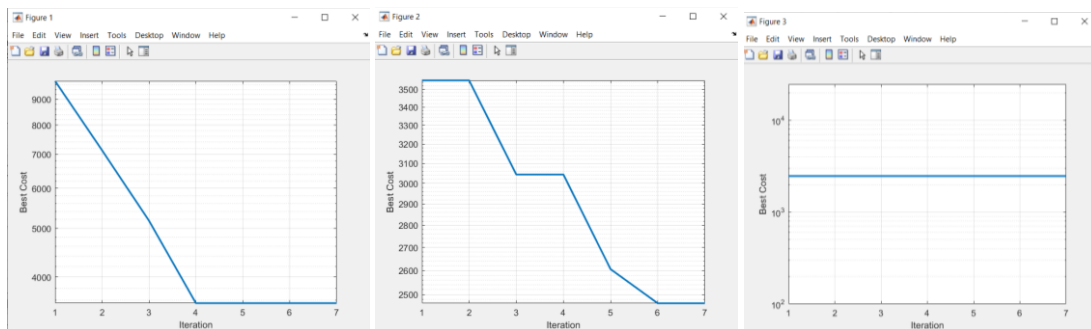


Ilustración 28. Convergencia del resultado.

A continuación, se muestran los resultados obtenidos para los diferentes escenarios, con el número de elementos óptimos y el coste asociado para toda la vida útil del sistema de autoconsumo [Figura 29].

		SOLTERO			FAMILIA			FAMILIA NUMEROSA		
		Paneles	Baterías	Aerogeneradores	Paneles	Baterías	Aerogeneradores	Paneles	Baterías	Aerogeneradores
WELLINGTON	Elementos óptimos	6	2	0	5	3	1	10	7	1
	Coste vida útil	2.464,00			3.916,50			6.932,40		
OSLO	Elementos óptimos	4	1	0	10	3	0	20	5	0
	Coste vida útil	3.206,70			6.804,60			13.127		
PHOENIX	Elementos óptimos	2	2	0	5	3	0	10	6	0
	Coste vida útil	1.552,90			2.892,20			5.564		
ZARAGOZA	Elementos óptimos	3	2	0	7	4	0	10	5	1
	Coste vida útil	1.758,40			3.562,50			5.721		

Ilustración 29. Resultados simulación para los diferentes escenarios.

En cuanto a los resultados obtenidos, se concluye que son válidos y lógicos. En Wellington, sitio caracterizado por el viento, para dos de los escenarios se obtiene que lo óptimo sería colocar al menos un aerogenerador, mientras que, en Phoenix, sitio caracterizado por una gran cantidad de irradiación solar durante todo el año, se obtiene que es donde menos cantidad de paneles solares son necesarios, ya que, con menos, se genera la energía necesaria.

En cuanto a los costes, se observa que los mayores precios se obtienen en Oslo, tal y como cabría esperar, ya que es un lugar que no se caracteriza ni por mucho viento ni mucho sol, y, por tanto, habría que colocar una mayor cantidad de elementos para generar la misma cantidad de energía que en los otros sitios. El coste aumenta notablemente cuando se aumenta el consumo de los hogares, tal y como cabría esperar.

9.2. AHORRO VIDA ÚTIL

A continuación, se muestra una tabla con el ahorro que se obtiene implementando los sistemas de autoconsumo para cada uno de los escenarios mencionados anteriormente [Figura 30].

	SOLTERO		FAMILIA		FAMILIA NUMEROSA	
	Sin autoconsumo	Ahorro	Sin autoconsumo	Ahorro	Sin autoconsumo	Ahorro
WELLINGTON	5.215,10	2.751,10	11.123,30	7.206,80	21.391,40	14.459
OSLO		2.008,40		4.318,70		8.264,40
PHOENIX		3.662,20		8.231,10		15.827,40
ZARAGOZA		3.456,70		7.560,80		15.670,40

Ilustración 30. Ahorro en costes con instalación de autoconsumo.

El coste sin autoconsumo es igual para cada uno de los escenarios de consumo, ya que para todos los casos se utiliza el mismo perfil de consumo y los mismos precios de compra de la red energética. Esto se calcula suponiendo que todo el consumo se satisface consumiendo la energía eléctrica directamente de la red sin nada de generación renovable de autoconsumo.

Se observa que, para todos los escenarios, se produce un ahorro notablemente elevado, por lo que se puede concluir que, para cualquier caso, es mucho más eficiente, tanto energética como económicamente, tener un sistema de autoconsumo. Se observa que donde menos ahorro se produce es en Oslo, ya que, el número de elementos necesarios para satisfacer la demanda energética es mucho mayor que en el resto de los sitios y, por tanto, el coste de instalación es mucho mayor.

10. CONCLUSIONES

En el proyecto se ha demostrado la viabilidad y eficacia del método de optimización de enjambre de partículas para la gestión e implementación de sistemas de autoconsumo con baterías, además de sentar las bases para futuras investigaciones y aplicaciones dentro del sector energético. Las principales conclusiones obtenidas son las siguientes:

- Se ha desarrollado e implementado de forma exitosa el código de optimización, donde se simula de manera efectiva el comportamiento de sistemas de autoconsumo con baterías conectado a la red eléctrica, permitiendo una evaluación precisa y detallada del rendimiento y eficacia del sistema para diferentes escenarios.
- Las simulaciones realizadas demuestran que el método de optimización seleccionado es adecuado para este tipo de problemas de optimización de sistemas de autoconsumo.
- Los resultados, tal y como se esperaba, muestran mejoras significativas en la eficiencia energética de los hogares y en la reducción de costes.
- Se destaca la flexibilidad del código, con una gran adaptabilidad a diferentes configuraciones y condiciones, de forma que puede ser aplicado a una gran cantidad de escenarios diferentes, e incluso permitiendo la optimización personalizada para diferentes usuarios.
- Potencial para investigación científica, generando un marco teórico y práctico que puede servir como base para futuras investigaciones. El proyecto abre la puerta para la elaboración de un artículo científico que pueda contribuir significativamente al campo de la eficiencia energética y las energías renovables.

Las simulaciones realizadas demuestran que el método PSO es adecuado para la optimización de sistemas de autoconsumo con baterías. Los resultados indican mejoras significativas en la eficiencia energética y en la reducción de costos operativos, validando la efectividad del enfoque adoptado.

11. REFERENCIAS BIBLIOGRÁFICAS

- [1] Gil Mena, Nasimba Medina, Bouakkaz, Haddad. "Analysis and optimisation of collective self-consumption in residential buildings in Spain". Energy and buildings 283 (2023). <https://doi.org/10.1016/j.enbuild.2023.112812>
- [2] Dasí Crespo, Roldán Blay, Escrivá Escrivá, Roldán Porta. "Evaluation of the spanish regulation on self consumption photovoltaic installations. A case study base don a rural municipality in Spain" Renewable Energy 204 (2023) 788-802 <https://doi.org/10.1016/j.renene.2023.01.055>
- [3] Afrasyabi, Saadi Mesgari, M. El-kenawy, Kaveh, Abdelhammeed Ibrahim, Khodadadi. "A crossover-based multi-objective discrete particle optimization model for solving multi-modal routing problems". Decision Analytics Journal 9 (2023). <https://doi.org/10.1016/j.dajour.2023.100356>
- [4] Guía profesional de tramitación del autoconsumo. https://www.idae.es/sites/default/files/documentos/publicaciones_idae/2023-01-10_Guia_Profesional_Tramitacion_autoconsumo_v.5.1_NIPO-2023.pdf
- [5] Definición de autoconsumo. <https://www.idae.es/tecnologias/energias-renovables/oficina-de-autoconsumo/que-es-el-autoconsumo>
- [6] Particle Swarm Optimization. <https://yarpiz.com/440/ytea101-particle-swarm-optimization-pso-in-matlab-video-tutorial>
- [7] Ant Colony Optimization. <https://revistas.ucc.edu.co/index.php/in/article/download/454/459>
- [8] Simulated Annealing Optimization. <https://www.youtube.com/watch?v=OdoM37ZEY4g>
- [9] Precio de la red eléctrica <https://www.esios.ree.es/es>
- [10] Pflugradt et al, (2022). Load Profile Generator: An Agent-Based Behavior Simulation for Generating Residential Load Profiles. Journal of Open Source Software, 7(71),3574. <https://doi.org/10.21105/joss.03574>

12. ANEXOS

12.1. CÓDIGO COMPLETO ALGORITMO OPTIMIZACIÓN

```
%% SE RESETEAN TODAS LAS VARIABLES

clc;
clear;
close all;

%% DEFINICION DEL PROBLEMA. Se define la funcion de coste, numero de variables y
valor máximo y mínimo de las variables

problem.CostFunction = @(x, compra) COSTE_codigoFinal(x, compra); % Cost Function
problem.nVar = 3; % Number of
Unknown (Decision) Variables
problem.VarMin = 0; % Lower Bound of
Decision Variables
problem.VarMax = 20; % Upper Bound of
Decision Variables

%% PARAMETROS DEL PSO. Se definen los parametros necesarios para la optimizacion

params.MaxIt = 7; % Maximum Number of Iterations
params.nPop = 6; % Population Size (Swarm Size)
params.w = 1; % Inertia Coefficient
params.wdamp = 0.99; % Damping Ratio of Inertia Coefficient
params.c1 = 2; % Personal Acceleration Coefficient
params.c2 = 2; % Social Acceleration Coefficient
params.ShowIterInfo = true; % Flag for Showing Iteration Informatin

%% SE INICIALIZAN LAS VARIABLES A UTILIZAR DURANTE EL CODIGO

for i=1:params.nPop % Los siguientes parametros se inicializan para
cada una de las particulas
genFotovoltaica = zeros(525600,i); % Se inicializa el vector generacion
fotovoltaica por minuto con valores iguales a 0
genEolica = zeros (525600,i); % Se inicizaliza el vector generación eólica
por minuto con valores iguales a 0
consumoBat = zeros(525600,i); % Se inicializa el vector consumo de las
baterias por minuto con valores de 0
cargaBat = zeros(525600,i); % Se inicializa el vector carga de las baterias
con excedente por minuto con valores de 0
compraRed = zeros (525600,i); % Se inicializa el vector compra a red por
minuto con valores de 0
costecompraRed = zeros (525600,i); % Se inicializa el vector coste de energia
comprada por minuto con valores de 0
cargaCompraRed = zeros (525600,i); % Se inicializa el vector carga de las baterias
con compra de red por minuto con valores de 0
SumaconsumoBat(i) = 0; % Se inicializa el valor del total de energia
consumida de la bateria con valor de 0
SumacargaBat(i) = 0; % Se inicializa el valor del total de energia
usada para cargar las baterias con valor de 0
SumacompraRed(i) = 0; % Se inicializa el valor del total de energia
comprada a la red con valor de 0
SumacosteRed(i) = 0; % Se inicializa el valor del total del coste a
pagar por la energia comprada a la red con valor de 0
```

```

end
consumo = zeros(525600,1);           % Se inicializa el vector demanda por minuto
con valores iguales a 0
precioRed = zeros(525600,1);        % Se inicializa el vector precio de red por
minuto con valores iguales a 0

%% CONSUMO POR MINUTO DURANTE UN AÑO NATURAL

file = 'Consumo familia numerosa.csv';           % Se leen los
datos de consumo previamente descargados de un archivo CSV
data = readtable(file, 'VariableNamingRule', 'preserve');
Tiempo = data.Timestep;
Hora = data.Time;                               % Se lee el día y
hora del consumo
consumo = data.Valor;                             % Se lee el valor
del consumo

%% PRECIO DE RED POR MINUTO DURANTE UN AÑO NATURAL

file = 'Precio Red Horario.csv';                 % Se leen los datos
de precio previamente descargados en un archivo CSV
data = readtable(file, 'VariableNamingRule', 'preserve');
HoraPrecio = data.datetime;                       % Se lee el día y
hora del precio
Precio = data.value;                               % Se lee el valor
del precio de red

% Se convierten los valores horarios obtenidos en valores por minuto
precio_minuto = zeros(525600,1);
precio_minuto = repmat(Precio(1),1,60);
for k=2:8760
    precio_minuto = [precio_minuto repmat(Precio(k),1,60)];
end
precioRed = reshape(precio_minuto, [525600, 1]); % Vector final de
precio por minuto durante un año natural

%% SE DESCARGAN LOS VALORES DE VIENTO POR HORA DURANTE UN AÑO NATURAL

% Copyright (c) 2023 José Luis Bernal Agustín
% All rights reserved
%{
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

    * Redistributions of source code must retain the above copyright
      notice, this list of conditions and the following disclaimer.

    * Redistributions in binary form must reproduce the above copyright
      notice, this list of conditions and the following disclaimer in
      the documentation and/or other materials provided with the distribution

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

```

AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
%}
```

```
% Reemplaza 'your_token_here' por tu token de acceso
token = '180cc9517ffcc58dd9496e01a3ff6785f8ceb945';
```

```
% Configurar los parámetros de la consulta para eólica
api_base = 'https://www.renewables.ninja/api/data/wind'; % Base URL para la API
eólica de Renewable Ninja                               % Latitud de la
lat = 41.65;                                           % Longitud de la
lon = -0.88;                                           % Fecha de inicio del
date_from = '2019-01-01';                             % Fecha de fin del
date_to = '2019-12-31';                               % Conjunto de datos a
dataset = 'merra2';                                   % Capacidad de la
capacity = 1.0;                                       % Altura de la turbina
instalación en kW                                     % Modelo de la turbina
height = 10;                                          % Formato de los datos
turbine = 'Vestas V80 2000';
eólica
format = 'csv';
de salida
raw = 'true';
```

```
% Construir la URL de la consulta
args = sprintf('lat=%f&lon=%f&date_from=%s&date_to=%s&dataset=%s&capacity=%f&height=%d&turbine=%s&format=%s&raw=%s', lat, lon, date_from, date_to, dataset, capacity, height, urlencode(turbine), format, raw);
url = strcat(api_base, '?', args);
```

```
% Configurar las opciones de solicitud
options = weboptions;
options.HeaderFields = {'Authorization', sprintf('Token %s', token)}; % Añadir el
token de acceso a la cabecera de la solicitud
```

```
% Realizar la consulta a la API
disp(url) % Mostrar la URL de la
consulta
data = webread(url, options); % Descargar los datos
de la URL
```

```

% Procesar los datos descargados
resultados = zeros (3,8760); % Se inicializan los
valores descargados de viento % Separar las filas de
data = splitlines(data); % Eliminar las primeras
datos % Eliminar ultima fila
data(1:4) = []; % Eliminar las primeras
filas que no contienen datos % Eliminar ultima fila
data(8761) = []; % Eliminar ultima fila
que no tiene datos % Obtener los nombres
header = split(data(1), ','); % Obtener los nombres
de las columnas % Se obtienen los
resultados = split(data, ','); % Se obtienen los
valores separados por columnas
time = resultados(:,1);
electricity = resultados(:,2);
wind_speed = zeros (8760,1);
wind_speed = str2double(resultados(:,3)); % Se obtiene el valor
del viento en m/s para cada hora

%% SE DEFINEN LAS VARIABLES DEL PROBLEMA A PARTIR DE LOS VALORES INTRODUCIDOS
PREVIAMENTE

CostFunction = problem.CostFunction; % Cost Function
nVar = problem.nVar; % Number of Unknown (Decision)
Variables
VarSize = [1 nVar]; % Matrix Size of Decision Variables
VarMin = problem.VarMin; % Lower Bound of Decision Variables
VarMax = problem.VarMax; % Upper Bound of Decision Variables

%% SE DEFINEN LAS VARIABLES DE LA OPTIMIZACIÓN A PARTIR DE LOS VALORES INTRODUCIDOS
PREVIAMENTE

MaxIt = params.MaxIt; % Maximum Number of Iterations
nPop = params.nPop; % Population Size (Swarm Size)
w = params.w; % Inertia Coefficient
wdamp = params.wdamp; % Damping Ratio of Inertia Coefficient
c1 = params.c1; % Personal Acceleration Coefficient
c2 = params.c2; % Social Acceleration Coefficient
ShowIterInfo = params.ShowIterInfo; % The Flag for Showing Iteration
Information
MaxVelocity = 0.2*(VarMax-VarMin); % Valor máximo del parametro velocidad
MinVelocity = -MaxVelocity; % Valor mínimo del parametro velocidad

%% INICIALIZACIÓN DE PARAMETROS A PARTIR DE VARIABLES DE ESTADO ALEATORIAS (1ER
CÁLCULO)

% The Particle Template (Se crea un vector de particula vacia con toda la
informacion que tiene que tener cada una de las particulas)
empty_particle.Position = []; % En la variable position se
almacenan los valores de las variables de estado
empty_particle.Velocity = []; % En la variable velocidad se
almacena el valor de la velocidad para cada particula

```

```

empty_particle.Cost = []; % En la variable coste se
almacena el valor del coste obtenido segun las variables de estado
empty_particle.Best.Position = []; % En la variable best position
se almacena el mejor valor de posicion de la particula
empty_particle.Best.Cost = []; % En la variable best cost se
almacena el mejor valor de coste de la particula

% Create Population Array (Se crea un vector de particula igual al de la particula
vacía tantas veces como sea el tamaño de la población)
particle = repmat(empty_particle, nPop, 1);

% Initialize Global Best (Se inicializa el coste de la mejor particula como un
numero muy elevado)
GlobalBest.Cost = inf;

% Initialize Population Members (Para cada una de las particulas)
for i=1:nPop % Para cada una de las
partículas se inicializan una serie de valores necesarios para la simulación
deficit= zeros(525600,i); % Se inicia el valor del
decifit de energía para cada particula con valor 0 para cada minuto
excedente= zeros(525600,i); % Se inicia el valor del
excedente de energía para cada particula con valor a 0 para cada minuto
CargaPosible=zeros(525600,i); % Se inicia el valor de carga
posible de energía para cada particula con valor a 0 para cada minuto
ConsumoPosible=zeros(525600,i); % Se inicia el valor de consumo
posible de energía para cada particula con valor a 0 para cada minuto
end

for i=1:nPop

    particle(i).Position = round(unifrnd(VarMin, VarMax, VarSize)); % Se
generan valores aleatorios reales de las variables de estado

    % Se calcula la generación solar de cada particula por minuto

    Auxiliar1 = particle(i).Position; % Variable
auxiliar que se utilizará más adelante

    % Reemplaza 'your_token_here' por tu token de acceso
    token = '180cc9517ffcc58dd9496e01a3ff6785f8ceb945';

    % Configurar los parámetros de la consulta
    api_base = 'https://www.renewables.ninja/api/data/pv'; % Base URL de
la API de Renewable Ninja
    lat = 41.65; % Latitud de la
ubicación
    lon = -0.88; % Longitud de la
ubicación
    date_from = '2014-01-01'; % Fecha de
inicio del periodo de consulta
    date_to = '2014-12-31'; % Fecha de fin
del periodo de consulta
    dataset = 'merra2'; % Conjunto de

```

```

datos a consultar
    capacity = 0.55*(Auxiliar1(1)); % Capacidad de
la instalación en kW
    system_loss = 0.10; % Pérdidas del
sistema
    tracking = 0; % Seguimiento
solar
    tilt = 35; % Inclinación
de los paneles solares
    azim = 180; % Orientación
de los paneles solares
    format = 'csv'; % Formato de
los datos de salida
    metadata = 'false'; % Indicar si se
incluye información adicional (metadata) en la respuesta

    % Construir la URL de la consulta
    args = sprintf('lat=%f&lon=%f&date_from=%s&date_to=%s&dataset=%s&capacity=%f&system_loss=%f&tracking=%d&tilt=%d&azim=%d&format=%s', lat, lon, date_from, date_to, dataset, capacity, system_loss, tracking, tilt, azim, format);
    url = strcat(api_base, '?', args);

    % Configurar las opciones de solicitud
    options = weboptions;
    options.HeaderFields = {'Authorization', sprintf('Token %s', token)}; %
Añadir el token de acceso a la cabecera de la solicitud

    % Realizar la consulta a la API
    disp(url) % Mostrar la URL de la consulta
    data = webread(url, options); % Descargar los datos de la URL

    % Procesar los datos descargados
    data = splitlines(data); %
Separar las filas de datos
    data(1:4) = []; %
Eliminar las cuatro primeras filas que no contienen datos
    header = split(data(1), ','); %
Obtener los nombres de las columnas
    formatSpec = '%{yyyy-MM-dd HH:mm}D%f'; %
Especificar el formato de los datos
    C = textscan(strjoin(data, '\n'), formatSpec, 'Delimiter', ','); %
Convertir los datos de texto a variables numéricas y de fecha
    time = C{1}; %
Obtener los valores de fecha y hora
    electricity = C{2}; %
Obtener los valores de energía generada

    % Se convierten los valores horarios en valores por minuto
    generacion_minuto = zeros(525600,1);
    generacion_minuto = repmat(electricity(1),1,60);
    for k=2:8760
        generacion_minuto = [generacion_minuto repmat(electricity(k),1,60)];
    end

```

```

    generacion_minuto = generacion_minuto/(60);
    genFotovoltaica(:,i) = reshape(generacion_minuto, [525600, 1]); %
Se almacena la generacion fotovoltaica por minuto para cada una de las particulas

% Se calcula la generaci3n eolica por minuto para cada particula

    altura_medicion = height; %
% Altura a la que se mide la velocidad del viento (m)
    altura_rotor = 15; %
% Altura del rotor de savonius (m)
    diametro_rotor = 0.6; %
% Diametro del rotor Savonius (m)
    coeficiente_CP = 0.25; %
% Coeficiente de potencia del Savonius Valor tipico para rotor Savonius
    densidad_aire = 1.225; %
% Densidad del aire (kg/m3) Asumiendo condiciones estandar de temperatura y nivel
del mar
    velocidades_curva = [0, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
19, 20]; % Velocidades en la curva de potencia del Savonius (m/s) Valores hoja
fabricante
    potencia_curva = [0, 0.024, 0.096, 0.0188, 0.326, 0.516, 0.77, 1.094, 1.384,
1.615, 1.785, 1.935, 2.007, 2.045, 1.072, 2.090, 2.105, 2.115]; % Potencia
correspondiente a la curva de potencia
    alfa = 0.2; %
% Coeficiente de rugosidad del terreno Valor tipico de terreno abierto
    velocidades_viento_ajustadas = wind_speed*(altura_rotor/altura_medicion)^alfa; %
% Ajustar velocidades del viento a la altura del rotor
    area_barrida = altura_rotor*diametro_rotor; %
% Area barrida por el Savonius (m2)
    energia_generada = zeros(8760,1); %
% Inicializar vector de energa generada (kWh)

    % Calcular energa generada
    for g=1:8760
        velocidad_viento_rotor = velocidades_viento_ajustadas(g); %
% Ajustar velocidad del viento a altura del rotor
        potencia_teorica = (0.5*
*densidad_aire*area_barrida*velocidad_viento_rotor^3*coeficiente_CP)/1000; %
% Formula de la potencia, se divide entre 1000 para pasar a kW
        potencia_generada = min(interp1(velocidades_curva,potencia_curva,
velocidad_viento_rotor,'linear','extrap'),potencia_teorica); % Encontrar potencia
generada limitada por la curva de potencia Savonius
        energia_generada(g) = potencia_generada*1; %
% Se multiplica por 1 para conocer la energa cada hora
    end

    % Convertir los valores a minutos
    eolica_minuto = zeros(525600,1); % Se
inicializa el vector por minuto
    eolica_minuto = repmat(energia_generada(1),1,60);
    for k=2:8760
        eolica_minuto = [eolica_minuto repmat(energia_generada(k),1,60)];
    end

```

```

    eolica_minuto = eolica_minuto/(60);
    genEolica(:,i) = reshape(eolica_minuto, [525600, 1])*Auxiliar1(3); % Se
obtiene el valor de energía por minuto a partir del viento para cada aerogenerador
y se multiplica por el numero de aerogeneradores

% Se calcula lo que pasa en la instalacion para cada minuto

    SOCmin=0.25; % Porcentaje minimo de carga que
pueden alcanzar las baterias
    precioLimite = 0.07; % Se da valor al precio limite para
poder comprar energía de la red para recarga de bateria
    capacidadBaterias(i)=Auxiliar1(2)*2.4; % Numero de baterias por energia
maxima de cada bateria
    SOC(i)=capacidadBaterias(i)*0.3; % Las baterias tienen inicialmente
un estado de carga del 30%

    % Para cada minuto del año se calcula todo segun las variables
for k=1:525600

    if genFotovoltaica(k,i)+genEolica(k,i) >= consumo(k)
% Si hay más generación que consumo se cargan baterias
        excedente(k,i) = genFotovoltaica(k,i)+genEolica(k,i)-consumo(k);
% Se calcula el excedente generado por minuto
        CargaPosible(k,i) = min(excedente(k,i), (capacidadBaterias(i)-SOC(i)));
% Se calcula el valor de carga posible segun SOC y capacidad de baterias para cada
minuto
        SOC(i) = SOC(i) + CargaPosible(k,i);
% Se actualiza el valor de SOC
        cargaBat(k,i) = CargaPosible(k,i);
        SumacargaBat(i) = SumacargaBat(i)+cargaBat(k,i);
% Se actualiza el valor total de energia cargada
    end

    if consumo(k) > genFotovoltaica(k,i)+genEolica(k,i)
% Si más consumo que generación se descargan las baterias
        deficit(k,i) = consumo(k)-(genFotovoltaica(k,i)+genEolica(k,i));
% Se calcula el deficit de energia por minuto
        ConsumoPosible(k,i) = min(deficit(k,i), (SOC(i)-capacidadBaterias(i)
*SOCmin)); % Se calcula el consumo posible de las baterias
        SOC(i) = SOC(i)-ConsumoPosible(k,i);
% Se actualiza el valor de SOC
        consumoBat(k,i) = ConsumoPosible(k,i);
        SumaconsumoBat(i) = SumaconsumoBat(i)+consumoBat(k,i);
% Se actualiza el valor total de energia consumida

        if ConsumoPosible(k,i) < deficit(k,i)
% Si no hay suficiente con las baterias para cubrir el deficit se compra red
            compraRed(k,i) = deficit(k,i)-ConsumoPosible(k,i);
% Se calcula la cantidad de energia que hay que comprar a red por minuto
            costecompraRed(k,i) = compraRed(k,i)*precioRed(k);
% Se calcula el coste a pagar por la energia de red por minuto
            SumacompraRed(i) = SumacompraRed(i)+compraRed(k,i);

```

```

        SumacosteRed(i) = SumacosteRed(i)+costecompraRed(k,i); %
% Se calcula el coste total de la energía comprada
        end
    end

if precioRed(k) < precioLimite && SOC(i) < capacidadBaterias(i) %
% Si precio red barato se recargan baterias
    cargaCompraRed(k,i) = min(capacidadBaterias(i)-SOC(i), consumo(k)*0.1);
    SOC(i)=SOC(i)+cargaCompraRed(k,i);
    SumacosteRed(i)=SumacosteRed(i)+cargaCompraRed(k,i)*precioRed(k); %
% Se suma el coste total de la energía comprada para recargar las baterias
end
end

% Initialize Velocity (Dar valor inicial 0 de velocidad a todas las particulas)
particle(i).Velocity = zeros(VarSize);

% Evaluation (Calcular la funcion coste para los valores iniciales de particula %
introduciendo las 3 variables de estado y el coste total de compra)
particle(i).Cost = CostFunction(particle(i).Position,SumacosteRed(i));

% Update the Personal Best (Se crea la mejor posicion y el mejor coste de cada %
particula, en este caso las iniciales)
particle(i).Best.Position = particle(i).Position;
particle(i).Best.Cost = particle(i).Cost;

% Update Global Best (Se actualiza el valor de coste y posicion de la mejor %
particula
if particle(i).Best.Cost < GlobalBest.Cost
    GlobalBest = particle(i).Best;
end

end

% Array to Hold Best Cost Value on Each Iteration
BestCosts = zeros(MaxIt, 1);

%% UNA VEZ INICIALIZADAS LAS PARTICULAS SE REALIZAN VARIAS ITERACIONES

for it=1:MaxIt % Para cada iteracion

for i=1:nPop % Para cada particula se reinician %
ciertos valores de interes, para que no se vayan sumando todo el rato
    deficit= zeros(525600,i);
    excedente= zeros(525600,i);
    CargaPosible=zeros(525600,i);
    ConsumoPosible=zeros(525600,i);
    SumacargaBat(i)=0;
    SumaconsumoBat(i)=0;
    SumacosteRed(i)=0;
    SumacompraRed(i)=0;

end
end

```

```

        for i=1:nPop                                % Para cada partícula en cada
iteración

            % Update Velocity (Se actualizan los valores de la velocidad en función
de los mejores resultados hasta el momento
            particle(i).Velocity = w*particle(i).Velocity ...
            + c1*rand(VarSize).*(particle(i).Best.Position - particle(i).
Position) ...
            + c2*rand(VarSize).*(GlobalBest.Position - particle(i).Position);

            % Apply Velocity Limits (Se aplican los límites máximos y mínimos de
velocidad
            particle(i).Velocity = max(particle(i).Velocity, MinVelocity);
            particle(i).Velocity = min(particle(i).Velocity, MaxVelocity);

            % Update Position (Se actualizan las variables de estado de la
partícula en función de los mejores resultados hasta el momento
            particle(i).Position = round(particle(i).Position + particle(i).
Velocity);

            % Apply Lower and Upper Bound Limits (Se aplican los valores máximos y
mínimos de las variables de estado)
            particle(i).Position = max(particle(i).Position, VarMin);
            particle(i).Position = min(particle(i).Position, VarMax);

            % Se calcula la generación solar de cada partícula por minuto

            Auxiliar1 = particle(i).Position;        % Variable
auxiliar que se utilizará más adelante

            % Reemplaza 'your_token_here' por tu token de acceso
            token = '180cc9517ffcc58dd9496e01a3ff6785f8ceb945';

            % Configurar los parámetros de la consulta
            api_base = 'https://www.renewables.ninja/api/data/pv';    % Base URL de
la API de Renewable Ninja
            lat = 41.65;        % Latitud de la
ubicación
            lon = -0.88;        % Longitud de la
ubicación
            date_from = '2014-01-01';    % Fecha de
inicio del periodo de consulta
            date_to = '2014-12-31';    % Fecha de fin
del periodo de consulta
            dataset = 'merra2';        % Conjunto de
datos a consultar
            capacity = 0.55*(Auxiliar1(1));    % Capacidad de
la instalación en kW
            system_loss = 0.10;        % Pérdidas del
sistema
            tracking = 0;        % Seguimiento
solar

```

```

        tilt = 35; % Inclinación
de los paneles solares
        azim = 180; % Orientación
de los paneles solares
        format = 'csv'; % Formato de
los datos de salida
        metadata = 'false'; % Indicar si se
incluye información adicional (metadata) en la respuesta

        % Construir la URL de la consulta
        args = sprintf('lat=%f&lon=%f&date_from=%s&date_to=%s&dataset=%s&capacity=%
f&system_loss=%f&tracking=%d&tilt=%d&azim=%d&format=%s', lat, lon, date_from,
date_to, dataset, capacity, system_loss, tracking, tilt, azim, format);
        url = strcat(api_base, '?', args);

        % Configurar las opciones de solicitud
        options = weboptions;
        options.HeaderFields = {'Authorization', sprintf('Token %s', token)}; %
Añadir el token de acceso a la cabecera de la solicitud

        % Realizar la consulta a la API
        disp(url) % Mostrar la URL de la consulta
        data = webread(url, options); % Descargar los datos de la URL

        % Procesar los datos descargados
        data = splitlines(data);
Separar las filas de datos
        data(1:4) = [];
Eliminar las cuatro primeras filas que no contienen datos
        header = split(data(1), ',');
Obtener los nombres de las columnas
        formatSpec = '%{yyyy-MM-dd HH:mm}D%f';
Especificar el formato de los datos
        C = textscan(strjoin(data, '\n'), formatSpec, 'Delimiter', ',');
Convertir los datos de texto a variables numéricas y de fecha
        time = C{1};
Obtener los valores de fecha y hora
        electricity = C{2};
Obtener los valores de energía generada

        % Se convierten los valores horarios en valores por minuto
        generacion_minuto = zeros(525600,1);
        generacion_minuto = repmat(electricity(1),1,60);
        for k=2:8760
            generacion_minuto = [generacion_minuto repmat(electricity(k),1,60)];
        end
        generacion_minuto = generacion_minuto/(60);
        genFotovoltaica(:,i) = reshape(generacion_minuto, [525600, 1]);
Se almacena la generacion fotovoltaica por minuto para cada una de las particulas

        % Se calcula la generación eolica por minuto para cada particula

        altura_medicion = height;

```

```

% Altura a la que se mide la velocidad del viento (m)
    altura_rotor = 15; %
% Altura del rotor de savonius (m)
    diametro_rotor = 0.6; %
% Diametro del rotor Savonius (m)
    coeficiente_CP = 0.25; %
% Coeficiente de potencia del Savonius Valor típico para rotor Savonius
    densidad_aire = 1.225; %
% Densidad del aire (kg/m3) Asumiendo condiciones estandar de temperatura y nivel
del mar
    velocidades_curva = [0, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
19, 20]; % Velocidades en la curva de potencia del Savonius (m/s) Valores hoja
fabricante
    potencia_curva = [0, 0.024, 0.096, 0.0188, 0.326, 0.516, 0.77, 1.094, 1.384,
1.615, 1.785, 1.935, 2.007, 2.045, 1.072, 2.090, 2.105, 2.115]; % Potencia
correspondiente a la curva de potencia
    alfa = 0.2; %
% Coeficiente de rugosidad del terreno Valor típico de terreno abierto
    velocidades_viento_ajustadas = wind_speed*(altura_rotor/altura_medicion)^alfa; %
% Ajustar velocidades del viento a la altura del rotor
    area_barrida = altura_rotor*diametro_rotor; %
% Area barrida por el Savonius (m2)
    energia_generada = zeros(8760,1); %
% Inicializar vector de energía generada (kWh)

    % Calcular energía generada
    for g=1:8760
        velocidad_viento_rotor = velocidades_viento_ajustadas(g); %
% Ajustar velocidad del viento a altura del rotor
        potencia_teorica = (0.5*
*densidad_aire*area_barrida*velocidad_viento_rotor^3*coeficiente_CP)/1000; %
% Formula de la potencia, se divide entre 1000 para pasar a kW
        potencia_generada = min(interp1(velocidades_curva,potencia_curva,
velocidad_viento_rotor,'linear','extrap'),potencia_teorica); % Encontrar potencia
generada limitada por la curva de potencia Savonius
        energia_generada(g) = potencia_generada*1; %
% Se multiplica por 1 para conocer la energía cada hora
    end

    % Convertir los valores a minutos
    eolica_minuto = zeros(525600,1); % Se
inicializa el vector por minuto
    eolica_minuto = repmat(energia_generada(1),1,60);
    for k=2:8760
        eolica_minuto = [eolica_minuto repmat(energia_generada(k),1,60)];
    end
    eolica_minuto = eolica_minuto/(60);
    genEolica(:,i) = reshape(eolica_minuto, [525600, 1])*Auxiliar1(3); % Se
obtiene el valor de energía por minuto a partir del viento para cada aerogenerador
y se multiplica por el numero de aerogeneradores

% Se calcula lo que pasa en la instalacion para cada minuto

```

```

SOCmin=0.25; % Porcentaje minimo de carga que
pueden alcanzar las baterias
precioLimite = 0.07; % Se da valor al precio límite para
poder comprar energía de la red para recarga de bateria
capacidadBaterias(i)=Auxiliar1(2)*2.4; % Numero de baterias por energia
maxima de cada bateria
SOC(i)=capacidadBaterias(i)*0.3; % Las baterias tienen inicialmente
un estado de carga del 30%

% Para cada minuto del año se calcula todo segun las variables
for k=1:525600

    if genFotovoltaica(k,i)+genEolica(k,i) >= consumo(k) %
% Si hay más generación que consumo se cargan baterias
        excedente(k,i) = genFotovoltaica(k,i)+genEolica(k,i)-consumo(k); %
% Se calcula el excedente generado por minuto
        CargaPosible(k,i) = min(excedente(k,i), (capacidadBaterias(i)-SOC(i))); %
% Se calcula el valor de carga posible segun SOC y capacidad de baterias para cada
minuto
        SOC(i) = SOC(i) + CargaPosible(k,i); %
% Se actualiza el valor de SOC
        cargaBat(k,i) = CargaPosible(k,i);
        SumacargaBat(i) = SumacargaBat(i)+cargaBat(k,i); %
% Se actualiza el valor total de energía cargada
    end

    if consumo(k) > genFotovoltaica(k,i)+genEolica(k,i) %
% Si más consumo que generación se descargan las baterias
        deficit(k,i) = consumo(k)-(genFotovoltaica(k,i)+genEolica(k,i)); %
% Se calcula el deficit de energia por minuto
        ConsumoPosible(k,i) = min(deficit(k,i), (SOC(i)-capacidadBaterias(i)
*SOCmin)); % Se calcula el consumo posible de las baterias
        SOC(i) = SOC(i)-ConsumoPosible(k,i); %
% Se actualiza el valor de SOC
        consumoBat(k,i) = ConsumoPosible(k,i);
        SumaconsumoBat(i) = SumaconsumoBat(i)+consumoBat(k,i); %
% Se actualiza el valor total de energia consumida

        if ConsumoPosible(k,i) < deficit(k,i) %
% Si no hay suficiente con las baterias para cubrir el deficit se compra red
            compraRed(k,i) = deficit(k,i)-ConsumoPosible(k,i); %
% Se calcula la cantidad de energía que hay que comprar a red por minuto
            costecompraRed(k,i) = compraRed(k,i)*precioRed(k); %
% Se calcula el coste a pagar por la energia de red por minuto
            SumacompraRed(i) = SumacompraRed(i)+compraRed(k,i);
            SumacosteRed(i) = SumacosteRed(i)+costecompraRed(k,i); %
% Se calcula el coste total de la energia comprada
        end
    end

end

if precioRed(k) < precioLimite && SOC(i) < capacidadBaterias(i) %
% Si precio red barato se recargan baterias

```

```

    cargaCompraRed(k,i) = min(capacidadBaterias(i)-SOC(i), consumo(k)*0.1);
    SOC(i)=SOC(i)+cargaCompraRed(k,i);
    SumacosteRed(i)=SumacosteRed(i)+cargaCompraRed(k,i)*precioRed(k);
% Se suma el coste total de la energía comprada para recargar las baterias
end
end

% Evaluation (Para cada iteracion se calculan los valores de la funcioncoste
    particle(i).Cost = CostFunction(particle(i).Position,SumacosteRed(i));

% Update Personal Best (Actualizar los mejores valores de cada particula)
    if particle(i).Cost < particle(i).Best.Cost

        particle(i).Best.Position = particle(i).Position;
        particle(i).Best.Cost = particle(i).Cost;

% Update Global Best (Actualizar los mejores valores globales)
    if particle(i).Best.Cost < GlobalBest.Cost
        GlobalBest = particle(i).Best;
    end

end

end

end

% Store the Best Cost Value (Almacenar los mejores valores de coste)
    BestCosts(it) = GlobalBest.Cost;

    % Display Iteration Information
    if ShowIterInfo
        disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCosts
(it))]);
    end

    % Damping Inertia Coefficient
    w = w * wdamp;

end

%out.pop = particle;
%out.BestSol = GlobalBest;
%out.BestCosts = BestCosts;

%% CALCULO DEL COSTE SIN INSTALACION DE AUTOCONSUMO PARA PODER COMPARAR

coste_sin_panel = zeros(525600,1);
sumacoste_sin_panel = 0;
for k=1:525600
    coste_sin_panel(k) = consumo(k)*precioRed(k);
    sumacoste_sin_panel = (sumacoste_sin_panel+coste_sin_panel(k));
end
sumacoste_sin_panel = 20*sumacoste_sin_panel; % Calculo

```

```
del coste total sin instalacion de autoconsumo
ahorro_vida_util=sumacoste_sin_panel-GlobalBest.Cost;

%% RESULTADOS GRÁFICOS

figure;
plot(BestCosts, 'LineWidth', 2);
semilogy(BestCosts, 'LineWidth', 2);
xlabel('Iteration');
ylabel('Best Cost');
grid on;
```

12.2. DESCARGA DE DATOS LOAD_PROFILE_GENERATOR

Para la descarga de perfiles de consumo utilizados en la simulación, se ha utilizado el software Load Profile Generator. Es una herramienta de acceso libre desde internet, muy útil para conseguir modelos de consumo energético en viviendas. A continuación, se muestra la vista general del software [Figura 31].

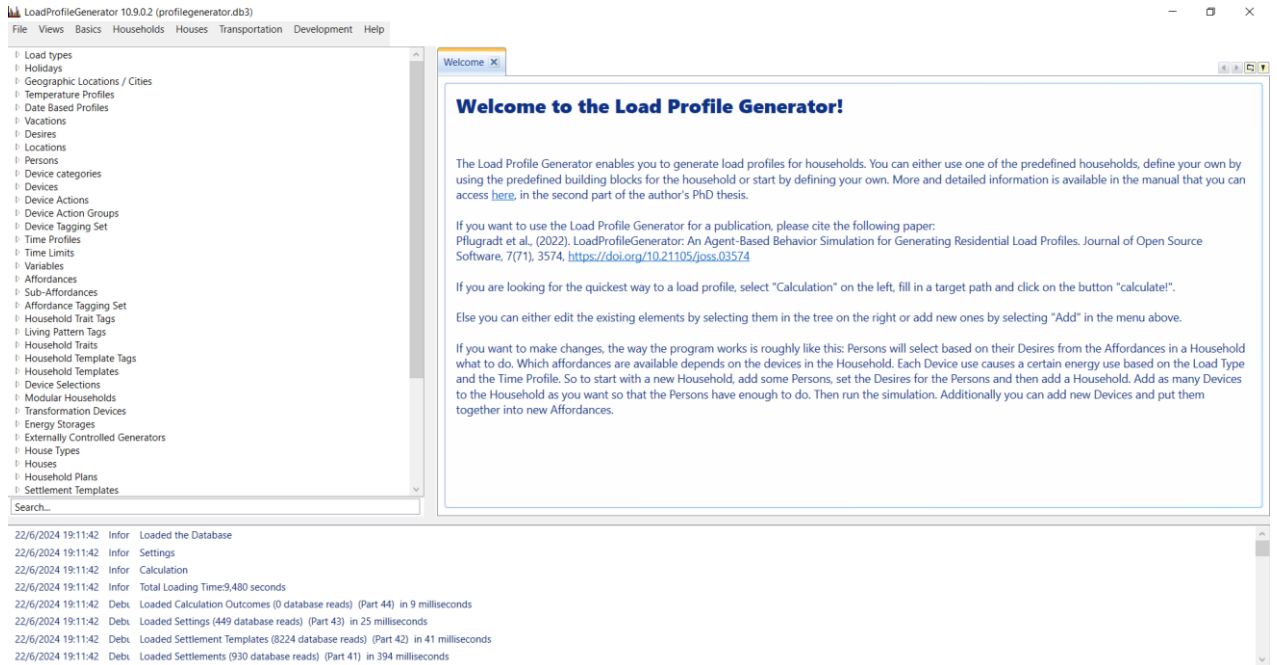


Ilustración 31. Vista general Load Profile Generator.

Para descargar un perfil de consumo, se pulsa en el botón de calculation de la pestaña de la izquierda y se abre la siguiente ventana [Figura 32].

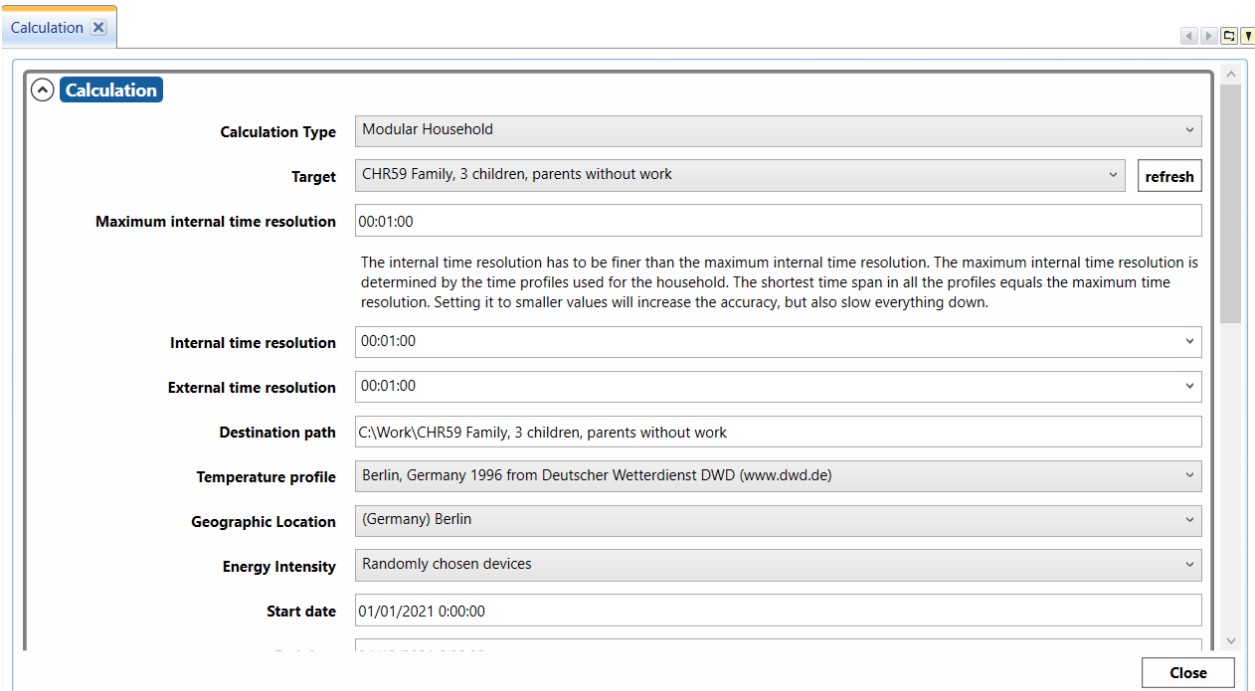


Ilustración 32. Ventana cálculo perfil de consumo.

A continuación, se rellena la información necesaria para obtener el perfil de consumo que se quiera. Existen más de 50 combinaciones diferentes de simulación dentro de esta herramienta en función de las características que se quieran simular [Figura 33]. Se puede distinguir entre parámetros como el número de convivientes, si tienen empleo, estudian, edad.

- CHR01 Couple both at Work
- CHR02 Couple, 30 - 64 age, with work
- CHR03 Family, 1 child, both at work
- CHR04 Couple, 30 - 64 years, 1 at work, 1 at home
- CHR05 Family, 3 children, both with work
- CHR06 Jak Jobless
- CHR07 Single with work
- CHR08 Single woman, 2 children, with work
- CHR09 Single woman, 30 - 64 years, with work
- CHR10 Single man, 30 - 64 age, shift worker
- CHR11 Student, Female, Philosophy
- CHR12 Student 2, Male, Philosophy
- CHR13 Student with Work

Ilustración 33. Combinaciones según las características de simulación.

En este proyecto se ha descargado la información de consumo de una persona soltera, de una familia que trabaja con un hijo y de una familia numerosa.

Por último, se rellenan el resto de los parámetros relevantes, entre los que destacan el periodo de tiempo del cual se quiere obtener el perfil de consumo [Figura 34].

Start date

End date

Ilustración 34. Periodo de tiempo de descarga de valores de consumo.

Una vez se ha relleno toda la información necesaria, se pulsa el botón de calcular y se obtiene el perfil de consumo en un documento Excel, con información del consumo para cada minuto del año según las características seleccionadas [Figura 35].

Los Excel descargados son los que se utilizan directamente en el código de simulación para leer la información de consumo para cada uno de los diferentes escenarios.

	A	B	C
1	Timestep	Time	Valor
2	0	01/01/2021 0:00	0.0005683630090290866
3	1	01/01/2021 0:01	0.0005595209299222224
4	2	01/01/2021 0:02	0.0005716490446679981
5	3	01/01/2021 0:03	0.0005609170074943872
6	4	01/01/2021 0:04	0.0005336728665323769
7	5	01/01/2021 0:05	0.0005721654202538088
8	6	01/01/2021 0:06	0.0005751645900540534

Ilustración 35. Excel con perfil de consumo por minuto.

12.3. DESCARGA DE DATOS DE PRECIO DE RED

La descarga de datos de la web se realiza a través de la web ESIOS, sistema de información desarrollado por la Red Eléctrica Española (REE). ESIOS es un sistema de información del operador del sistema para realizar tareas de información y gestión relacionadas con el mercado eléctrico.

Para la descarga de datos lo primero que hay que hacer es entrar en la propia web. En la web se puede encontrar información de todo tipo relacionada con el mercado eléctrico. Se puede obtener información de generación, de consumo, de precios y muchas cosas más de cualquier día que se quiera. A continuación, se muestra la pantalla de inicio de la web [Figura 36].



Ilustración 36. Pantalla de inicio ESIOS. Fuente: ESIOS.

Para la descarga de datos de precio de la red durante un año completo se pincha en la pestaña de Análisis [Figura 37]. Una vez en la ventana de análisis, aparece un buscador, en el cual hay que encontrar que es lo que se quiere analizar.

Se pueden analizar cualquier tipo de información relacionada con la generación y consumo, con la gestión de la demanda, con el mercado y los precios del mercado, intercambios...



Ilustración 37. Ventana análisis OMIE. Fuente: ESIOS.

Se puede analizar cualquier tipo de información relacionada con la generación y consumo, con la gestión de la demanda, con el mercado y los precios del mercado, intercambios...

En este caso, como se ha comentado anteriormente, interesa la descarga de datos de precios de la red eléctrica durante un año entero. Para encontrar lo que interesa se utiliza el buscador. Buscando "Precio medio horario mercado diario", aparece el listado de información relacionada

con la búsqueda. Dentro del listado se selecciona la que pone “Precio medio horario componente mercado diario contratación libre”, tal y como se observa en la siguiente imagen [Figura 38].



Ilustración 38. búsqueda información ESIOS. Fuente: ESIOS.

Se abre una ventana con un gráfico que muestra el precio horario de la red eléctrica. Hay que seleccionar los filtros de búsqueda que interesan. En este caso se quiere obtener el precio de la red de un año completo, por lo que se filtra para obtener los datos del año 2022. Una vez se ha seleccionado el rango que se quiere descargar para obtener los datos en formato Excel se pulsa en el botón exportar y a continuación a CSV [Figura 39].

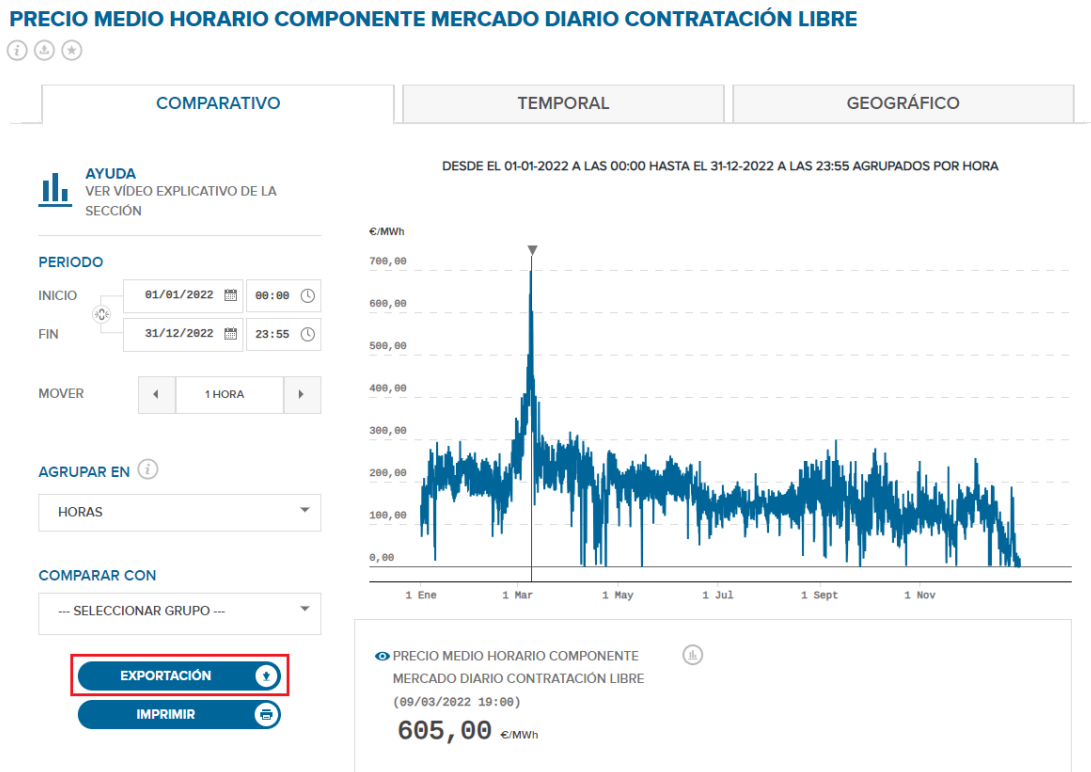


Ilustración 39. Precio horario de la red año 2022. Fuente: ESIOS.

Se obtiene, por tanto, un archivo Excel con toda la información requerida disponible para usar en el código de optimización.

12.4. FICHA TÉCNICA PANELES SOLARES



EM550-PH





144 Células MBB 72S 2P



Tecnología Half Cell Mono PERC



Mayor potencia de salida



Diseño ligero



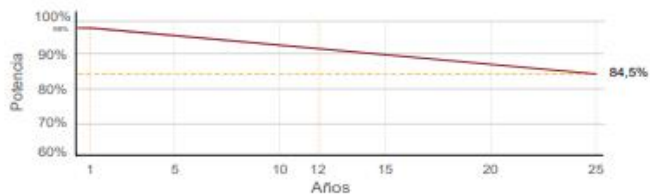
Rendimiento con poca luz



Mayor eficiencia de conversión del módulo

GARANTÍA

Garantía lineal de Potencia



- + Tolerancia positiva de vatios
- 12 Años de garantía del producto
- 25 Años de garantía de potencia lineal

Tensite
 info@tensite-energy.com
 www.tensite-energy.com



v-ES-1
 Patrocinador oficial de

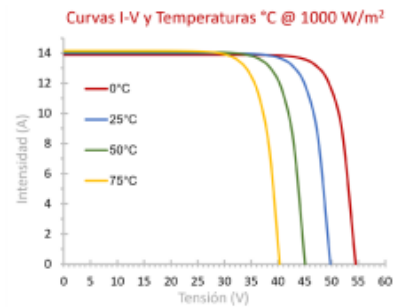
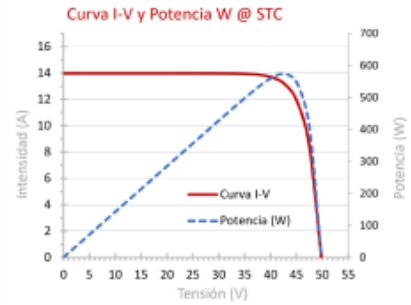


Datos Eléctricos STC

EM550-PH

Tipo de módulo	550M Half cell Mono PERC
Máxima potencia (Wp)	550 Wp
Corriente de potencia máxima (I_{mp})	13,12 A
Voltaje de potencia máxima (V_{mp})	41,95 V
Corriente de cortocircuito (I_{sc})	13,98 A
Voltaje de circuito abierto (V_{oc})	49,80 V
Eficiencia del módulo	21%
Fusible de serie máxima	25 A
Número de Diodos	3
Tolerancia positiva de potencia	0+3%
Condiciones de prueba estándar	1.000 W/m ² , 25 °C, AM 1.5
Voltaje máximo del sistema DC	1.500 V
Coefficiente de temperatura I_{sc}	0,048% / °C
Coefficiente de temperatura V_{oc}	-0,270% / °C
Coefficiente de temperatura P_{mp}	-0,350% / °C
Rango temperatura funcionamiento	-40°C / +85°C
Temperatura operación célula (TONC)	45°C ±2
Capacidad carga frontal del módulo	5.400 Pa IEC61215 (nieve)
Capacidad carga trasera del módulo	2.400 Pa IEC61215 (viento)

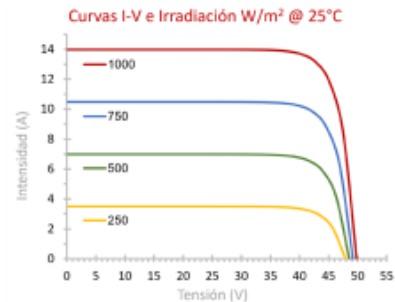
*Condiciones Estandar de Medida STC: Irradiación 1.000 W/m², espectro AM1.5, célula a 25°C.



Valores en condiciones TONC**

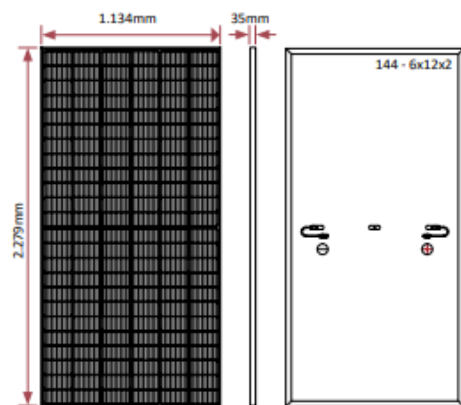
Potencia máxima TONC (P_{max})	416 W
Voltaje de potencia máxima (V_{mp} TONC)	39,65 V
Corriente de potencia máxima (I_{mp} TONC)	10,51 A
Voltaje de circuito abierto (V_{oc} TONC)	46,80 V
Corriente de cortocircuito (I_{sc} TONC)	11,11 A

**Condiciones TONC: Irradiación de 800 W/m², AM1.5, temperatura ambiente 20 °C y viento de 1 m/s.

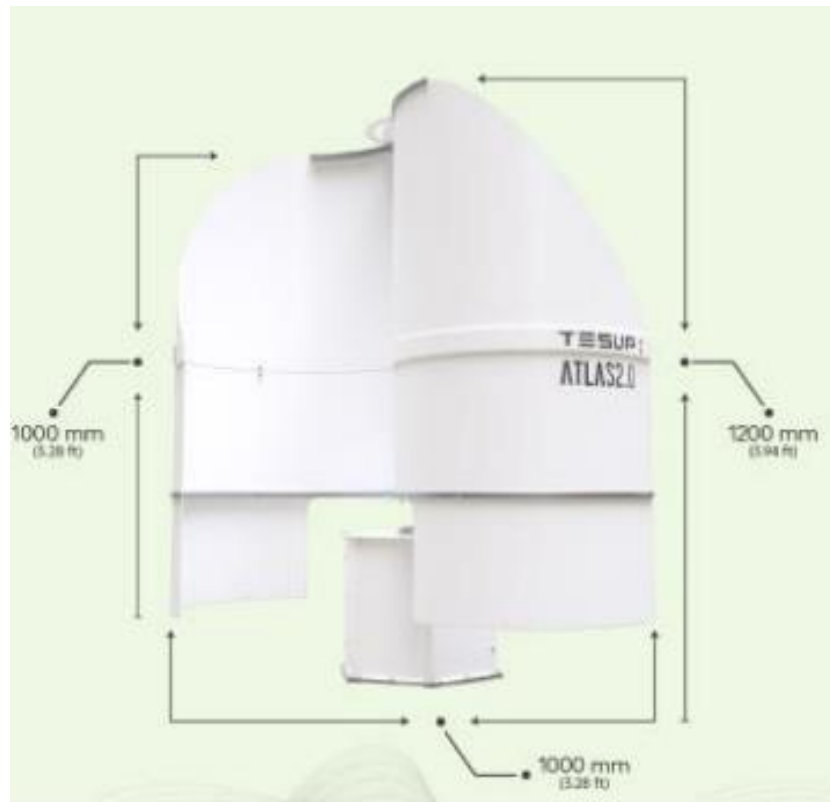


Características mecánicas

Cubierta frontal (material/espesor)	Vidrio templado / 3.2mm
Peso del módulo	27,2 kg
Dimensiones del módulo (L / W / H)	2.279 x 1.134 x 35mm
Lámina de protección posterior	TPT en blanco
Células (cantidad/material)	144 (6x12x2) / Silicio mono
Marco (material/color)	Aluminio anodizado / Plata
Grado protección caja de conexiones	≥ IP68
Cables y conectores	4mm ² , long. 1.400mm
Clasificación de calidad	Clase A
Clase de protección eléctrica	Clase II
Clase de seguridad contra incendios	Clase C

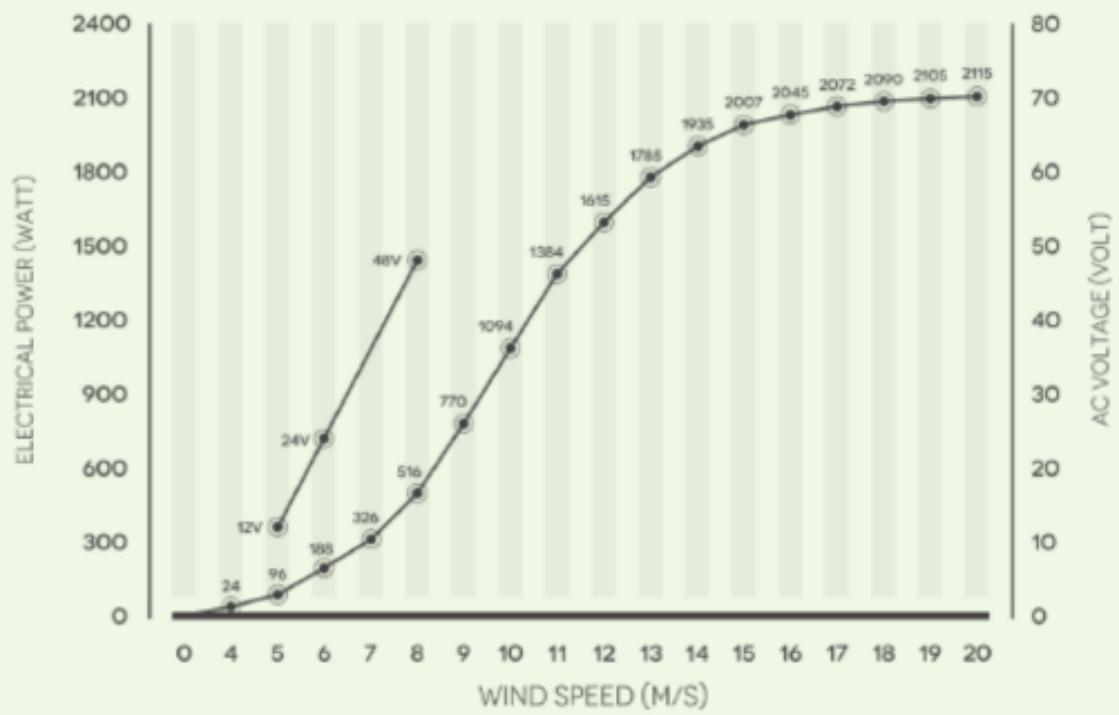


12.5. FICHA TÉCNICA AEROGENERADORES



SPECIFICATIONS	
DESIGNATION	12V to 48V
GENERATOR	
TYPE	2KW vertical axis wind permanent magnet generator
WEIGHT	20kg
MAX. POWER	2kW
OPERATING CIRCUIT VOLTAGE	0-100V
CURRENT	3-Phase
START OF CHARGING	3m/s
BASE PLATE MATERIAL	Sheetmetal
DIRECTION OF ROTATION	Clockwise
TEST STANDARDS	EN 61000-6-1 (electromagnetic compatibility - immunity) EN 61000-6-3 (electromagnetic compatibility - emissions)
ROTOR BLADES	
MATERIAL	Composites
HUB FLANGE	Aluminum
DIAMETER	600 (1.97 Feet)
WEIGHT PER ROTOR BLADES	750 g (1.65 lbs)
DIRECTION OF ROTATION	Counter-Clockwise
STARTING WIND SPEED	2 m/s
NO. OF BLADES	3
MAX RPM	650
MAX SPEED	50m/s
NOISE	30 dB

ATLAS2.0 WIND TURBINE ELECTRICAL POWER CURVE



12.6. FICHA TÉCNICA BATERÍAS



MANUAL DE PRODUCTO

BATERÍA DE LITIO-FOSFATO DE HIERRO,
LITIO FERROSO O LIFEPO4

24V 100AH

4 INSTALACIÓN

Las baterías deben instalarse siempre en posición horizontal.

4.1 Conexiones

- En serie: Se pueden conectar hasta 15 baterías en serie.
- En paralelo: Las baterías NO se pueden conectar en paralelo.

4.2 Protección contra cortocircuitos

En instalación de una sola batería o conectadas en serie con dispositivos como inversores solares se debe instalar un relé de seguridad.

4.3 Antes de usar

En el momento de su envío, las baterías están cargadas al 30% aproximadamente.

Al cargar baterías conectadas en serie, la tensión de las baterías o celdas con el estado de carga inicial más alto llegarán antes al estado de carga completa, dejando atrás las baterías o celdas con un estado de carga inicial inferior. Esto podría resultar en una sobretensión de las baterías o celdas más con el estado de carga inicial más alto, por lo que el BMS interrumpiría el proceso de carga.

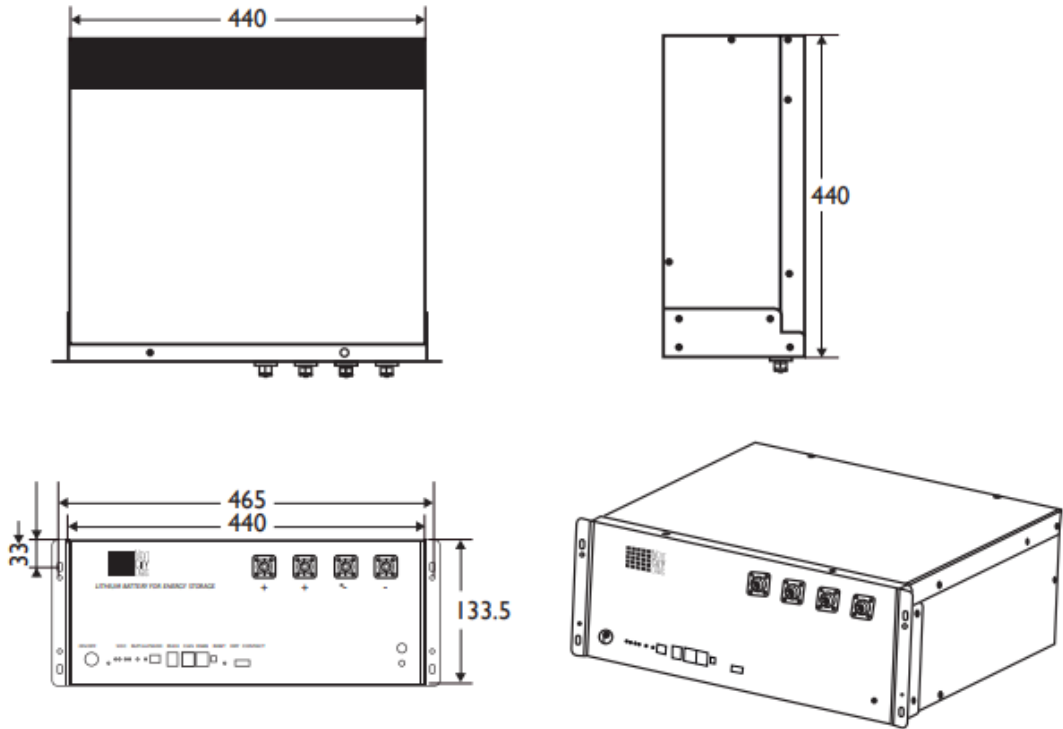
Es por ello que desde NextCity Labs® recomendamos encarecidamente que se carguen las baterías nuevas completamente antes de incluirlas en una configuración en serie.

Esto puede llevarse a cabo cargando las baterías individualmente a un ritmo bajo con un cargador o una fuente de alimentación según los valores de carga indicados en la ficha técnica. Para equilibrar completamente las celdas se recomienda añadir un periodo de absorción de varias horas al voltaje recomendado en la ficha técnica.

5 VARIABLES TÉCNICAS

FICHA TÉCNICA 24V 100AH	
Referencia	NCL-LFPO-24100
Composición	LiFePO4
Voltaje nominal	25.6V
Resistencia interna	≤30mΩ
Capacidad nominal	50AH
Energía nominal	2560WH
Voltaje carga	25.6-29.2V
Voltaje descarga	20-29.2V
Corriente máx. Carga	60A (recomendada) y 100A (Máxima)
Corriente máx. descarga	60A (recomendada) y 100A (Máxima)
Dimensiones	482x400x134mm
Peso	28Kg
Consumo BMS	<2W (operativo) / <100mW (stand by)
Monitoreo BMS	Voltaje sistema, corriente celda, voltaje celda, temperatura celda, temperatura sistema
SOC	Algoritmo inteligente
Comunicaciones	CAN/RS-485
Temperatura operativa	-10°C-50°C
Temperatura almacenaje	-20°C-45°C
Humedad	15%-85% (sin condensación)
Vida útil@80%DOD	>6000 ciclos
Garantía del producto	5 años
Vida útil estimada	+10 años (25°C)

6 DIMENSIONES



7 FUNCIONAMIENTO BÁSICO

