

7

**PROPUESTA DE RESOLUCIÓN
MEDIANTE ALGORITMOS
BASADOS EN NUBES DE
PARTÍCULAS**

7.1 INTRODUCCIÓN.

En el capítulo 5 se ha desarrollado un modelo de análisis de un sistema SOFC-TG alimentado con gases procedentes de la gasificación de biomasa residual. Este modelo se utiliza en el capítulo 6 como parte del análisis técnico, económico y geográfico global que le corresponde a la optimización del tamaño y la ubicación de un sistema SOFC-TG cuyo combustible son residuos forestales dispersos.

El primer modelo planteado en el apartado 6.2 (optimización de la potencia de generación desde el punto de vista del inversor en condiciones técnicas y geográficas constantes) requiere para su simulación de un tiempo computacional relativamente bajo, ya que su resolución se reduce a obtener la función objetivo (IR) en función de la potencia del sistema, o dicho de otra manera, calcular la función objetivo para un rango de superficies circulares de suministro de biomasa (o de radios) y de entre los resultados elegir el valor que maximiza a la función objetivo.

Sin embargo, el segundo modelo desarrollado en el apartado 6.3 persigue, además del cálculo del tamaño óptimo, también su localización, y plantea el problema considerando que una determinada región se divide en numerosas parcelas, cada una de ellas con unas características técnicas y energéticas diferentes.

En este capítulo se desarrolla un método para la resolución del segundo modelo planteado que no tenga un coste computacional elevado. La técnica elegida es un algoritmo basado en nubes de partículas. Se elige esta técnica por ser relativamente reciente su aplicación en la optimización de problemas complejos y el creciente interés que suscita en varios campos de la comunidad investigadora.

Esta Tesis no tiene como principal objetivo la comparación de varios métodos y la búsqueda de la mejor técnica metaheurística para resolver el problema de optimización aquí planteado, sino que pretende proponer un algoritmo bioinspirado

(como ejemplo de técnica metaheurística), relativamente novedoso, basado en nubes de partículas (algoritmo PSO), para la resolución de un problema de aprovechamiento energético de biomasa forestal residual con un sistema de generación eléctrica SOFC-TG. No obstante, se emplean los algoritmos genéticos para comparar los resultados obtenidos y demostrar las ventajas de la técnica propuesta.

En las referencias consultadas [56, 65, 139, 162] no se ha encontrado ningún método diseñado para resolver este tipo de problemas con técnicas no exactas. En estos trabajos la optimización se acomete partiendo de bases de datos pertenecientes a Sistemas de Información Geográfica (SIG) correspondientes a regiones reales. Los datos iniciales utilizados no están publicados, únicamente, en algunos casos se muestran mapas con poca resolución representando el potencial energético de la biomasa residual dispersa de la región a estudio. Además, para reducir la complejidad de los cálculos se utilizan simplificaciones tales como limitar la distancia máxima de recogida de los residuos, fijar de inicio unas pocas posibles ubicaciones de la planta, establecer la potencia de generación, etc.

7.2 PROBLEMÁTICA DE LA RESOLUCIÓN

La optimización del modelo propuesto parte de un gran conjunto de datos y una serie de condiciones que dificultan la utilización de métodos exhaustivos. La dificultad se presenta principalmente por la complejidad de los cálculos y la duración de éstos. Mediante el uso de métodos exactos se deberían de realizar los cálculos correspondientes a cada parcela (la biomasa utilizable, la energía que se genera a partir de esa biomasa teniendo en cuenta las características del sistema SOFC-TG, los costes de extracción, etc.) y el análisis económico global para diferentes tamaños del área de suministro de biomasa, es decir, para diferentes tamaños del conjunto de parcelas seleccionadas. El proceso se ha de repetir para todas las posibles ubicaciones de la planta en la región de estudio.

Por ejemplo, para afrontar conjuntamente el problema de la localización y del tamaño de la planta mediante la utilización de una técnica exhaustiva en una región que está compuesta por 128x128 parcelas, se tendrían que analizar 16384 posibles ubicaciones de la planta y habría que estudiar cada una de esas ubicaciones para diferentes tamaños. El tamaño máximo de la superficie de producción o área de suministro dependería de la naturaleza discreta del problema (número y dimensiones de las parcelas, forma geométrica del área, etc.) y de la capacidad máxima de la planta. Por ejemplo, considerando un radio de acción máximo de 64 parcelas (la mitad del mapa) desde la parcela donde hipotéticamente se sitúa la planta, la dimensión del problema ascendería a 1048576 simulaciones completas del modelo propuesto.

Si la ubicación de la planta estuviera definida, el problema sería abordable con métodos exhaustivos sin discusión, los cálculos serían mayores al primer método propuesto tantas veces como número de parcelas seleccionadas para extraer la biomasa residual, ya que cada parcela posee diferentes condiciones de extracción y tipos de residuos.

El problema aumenta enormemente por marcarse como objetivo de búsqueda la localización óptima del sistema, es decir, la parcela donde se sitúa la planta que maximiza la rentabilidad del modelo técnico, económico y geográfico global. En concreto, la dimensión del problema se amplía al número total de parcelas que componen toda la región a estudio y que no han sido descartadas para la ubicación. La dimensión del problema crece exponencialmente al número total de parcelas que componen el mapa o región del estudio.

7.3 JUSTIFICACIÓN DEL ALGORITMO DE RESOLUCIÓN

El problema está planteado para la búsqueda de la solución en un espacio discreto compuesto por unidades parcelarias cuadradas de superficies iguales, separadas unas de otras, con posiciones definidas y distribuidas de forma regular. Representando

el mapa con dos ejes de coordenadas, X e Y, la posición de una parcela se puede definir mediante dos números (enteros o binarios) que son las coordenadas X e Y. Para pasar de una parcela a otra hay que “saltar” en el espacio discreto y no se pueden adoptar posiciones intermedias. El espacio de búsqueda es discreto porque no se puede tener información geográfica precisa de cualquier punto del mapa acerca, por ejemplo, del tipo de vegetación, sino que es más práctico trabajar con valores medios en áreas de dimensiones relativamente pequeñas que en este caso se denominan parcelas.

El tamaño del problema va a depender de la superficie total de la región a optimizar y de las dimensiones de cada parcela, pero lo cierto es que cada parcela lleva asociados varios parámetros que la hacen diferente al resto, de ellos el principal es la posición inequívoca de cada una de ellas.

La posición de una parcela en un espacio geográfico discreto se puede definir como un vector binario y el algoritmo utilizado para la resolución del problema debe estar diseñado para utilizar la codificación binaria.

7.4 CODIFICACIÓN DEL PROBLEMA PARA EL ALGORITMO PSO BINARIO

Para la aplicación del algoritmo PSO se hace necesario definir o codificar la partícula y especialmente su posición. En cuanto a la función objetivo se elige entre el *IR* o el *VAN*.

Por otro lado, la superficie de producción o área de suministro se compone del conjunto de parcelas seleccionadas de las cuales se extrae la biomasa forestal. La forma del área de suministro se fija como una región cuadrada situándose la planta en el centro geométrico de este cuadrado como se puede observar en la figura 7.1. Se debe recordar que se pretende optimizar, además de la posición, el tamaño de la planta y, por tanto, el área de suministro de la biomasa residual.

La representación y codificación de una partícula depende de la solución que se pretende conseguir, es decir, una posición dentro de un mapa y una superficie determinada de la que se obtiene una cantidad de biomasa. De ahí que se defina la partícula como el conjunto de tres partes:

- 1) Componente X de la situación de la planta.
- 2) Componente Y de la situación de la planta.
- 3) Tamaño del área de la que se extrae la biomasa.

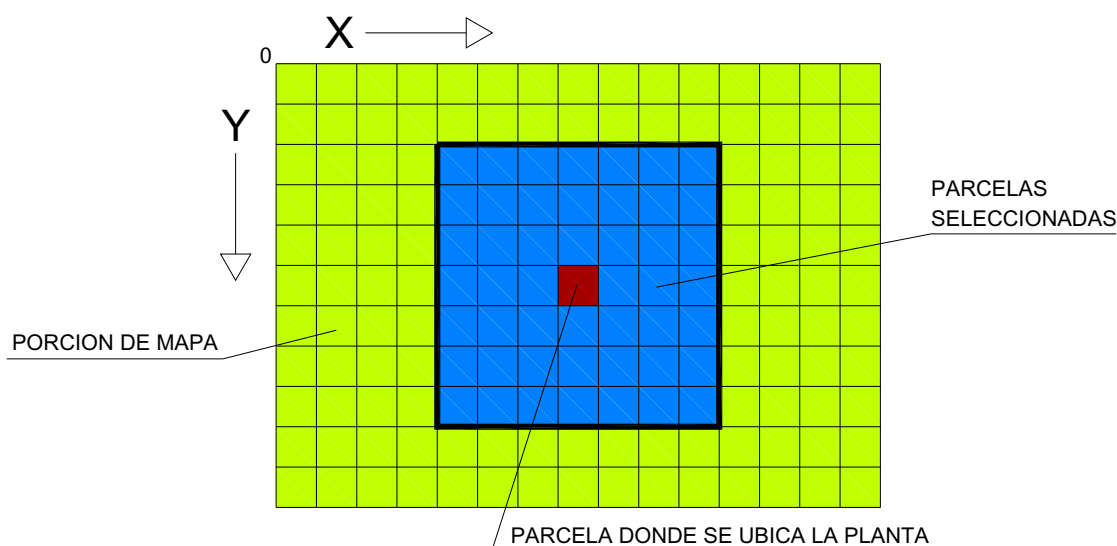


Figura 7.1: Representación gráfica de la partícula en una porción de mapa determinada.

En la figura 7.1 los componentes X e Y vienen dados por la posición de la parcela donde se ubica la planta (parcela roja). La superficie de producción es la región cuadrada que contiene las parcelas azules, de éstas se extrae la biomasa residual forestal.

Estos tres componentes son codificados a binario utilizando el llamado Código Gray que es un tipo de código binario que se caracteriza porque la representación de dos números consecutivos varía únicamente en un bit. Esta codificación no tiene nada que

ver con un sistema de cuantificación, la representación binaria en Gray de una variable no denota un valor matemático real sino más bien una posición o un estado. Por ello, la codificación en Gray es apropiada para utilizarse en problemas con espacios de búsqueda geográficos en las que la representación de posiciones cercanas difiere en muy pocos bits.

Junto con el código Gray se usa la distancia de Hamming, vista en el capítulo 3. La utilización de estos dos conceptos proporciona al algoritmo PSO binario empleado una mayor eficiencia y convergencia a la solución óptima, para este tipo de problemas y espacios de búsqueda considerados.

Volviendo a los tres componentes de la partícula, se ha considerado un espacio de búsqueda rectangular, con $x \in [1, L_X]$ e $y \in [1, L_Y]$, siendo L_X y L_Y los tamaños máximos en el eje X y eje Y del mapa considerado.

La parte de la partícula que representa el área de extracción de biomasa se realiza mediante la apotema, AP , de la región cuadrada. La apotema de un polígono regular se define como la distancia mínima entre el centro geométrico y uno de sus lados.

La representación y el número de parcelas seleccionado viene dado en función de la apotema como $(2 \cdot AP + 1)^2$. Para el valor de $AP = 0$ se supone una región que contiene una sola parcela, en donde se ubica la planta. Si $AP = 1$ se refiere a una región cuadrada de $3 \times 3 = 9$ parcelas. Sirva como ejemplo la representación existente en la figura 7.1 en la que la apotema es de 3 y el número de parcelas es de $7 \times 7 = 49$. Limitando la apotema a un valor máximo, AP_{\max} , la región más grande que le corresponde sería de $(2 \cdot AP_{\max} + 1)^2$ parcelas.

Así, el número total de bits que se usa para codificar la partícula, y por tanto, la solución que se busca sería:

$$N = \log_2 L_X + \log_2 L_Y + \log_2 AP_{\max} \quad (7.1)$$

El número de bits que conforman una partícula indican la dimensión del problema, N .

7.5 ALGORITMO DE RESOLUCIÓN PROPUESTO

La propuesta del algoritmo PSO para codificación binaria, que se presenta en esta Tesis, es una versión basada en el algoritmo de Afshinmanesh [2] visto en el capítulo 3. \mathbf{x}_i^t y \mathbf{v}_i^t son vectores binarios de dimensión N , en el tiempo o iteración t . La actualización de la posición de la partícula se calcula aplicando el operador binario XOR (OR exclusivo) representado por el símbolo \oplus . Así la ecuación en forma vectorial sería:

$$\mathbf{x}_i^t = \mathbf{x}_i^{t-1} \oplus \mathbf{v}_i^t \quad (7.2)$$

En esta versión, el vector velocidad funciona como vector de cambio. Es decir, si $v_{i,j}^t = '1'$, entonces $x_{i,j}^t = \bar{x}_{i,j}^{t-1}$ ($\bar{x}_{i,j}^t$ es la negación lógica de $x_{i,j}^t$), sin embargo, si $v_{i,j}^t = '0'$, entonces $x_{i,j}^t = x_{i,j}^{t-1}$ (el valor del bit no cambia).

El vector velocidad se actualiza en cada iteración aplicando la siguiente ecuación:

$$\mathbf{v}_i^t = \bar{\omega}_i^{t-1} + \omega_i^{t-1} \cdot (\mathbf{c}_{1i} \cdot (\mathbf{mejorpos}_i \oplus \mathbf{x}_i^{t-1}) + \mathbf{c}_{2i} \cdot (\mathbf{mejorposg} \oplus \mathbf{x}_i^{t-1})) \quad (7.3)$$

O para cada bit situado en la posición j :

$$v_{i,j}^t = \bar{\omega}_{i,j}^{t-1} + \omega_{i,j}^{t-1} \cdot \left(c_{1i,j} \cdot (\text{mejorpos}_{i,j} \oplus x_{i,j}^{t-1}) + c_{2i,j} \cdot (\text{mejorposg}_j \oplus x_{i,j}^{t-1}) \right) \quad (7.4)$$

Donde:

- $x_{i,j}^t \equiv$ Valor del bit en la posición j del vector de posición de la partícula i , $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,N})$ en el momento o iteración t .
- $v_{i,j}^t \equiv$ Valor del bit en la posición j del vector de velocidad i , $\mathbf{v}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,N})$ en la iteración t .
- $c_{1i,j} \equiv$ Valor del bit en la posición j perteneciente al vector de aprendizaje aleatorio $\mathbf{c}_{1i} = (c_{1i,1}, c_{1i,2}, \dots, c_{1i,N})$.
- $c_{2i,j} \equiv$ Valor del bit en la posición j perteneciente al vector de aprendizaje aleatorio $\mathbf{c}_{2i} = (c_{2i,1}, c_{2i,2}, \dots, c_{2i,N})$.
- $\text{mejorpos}_{i,j} \equiv$ Es el bit en la posición j del vector **mejorpos** _{i} = $(\text{mejorpos}_{i,1}, \text{mejorpos}_{i,2}, \dots, \text{mejorpos}_{i,N})$ que es la mejor posición encontrada hasta el momento para la partícula i .
- $\text{mejorposg}_j \equiv$ Es el bit en la posición j del vector **mejorposg** = $(\text{mejorposg}_1, \text{mejorposg}_2, \dots, \text{mejorposg}_N)$ que es la mejor posición global encontrada hasta el momento por toda la nube de partículas.
- $\omega_{i,j}^t \equiv$ Valor del bit en la posición j del vector de inercia para la partícula i , $\boldsymbol{\omega}_i = (\omega_{i,1}, \omega_{i,2}, \dots, \omega_{i,N})$, en la iteración t .
- $\bar{\omega}_{i,j}^t \equiv$ Valor del bit en la posición j del vector complementario al vector de inercia para la partícula i , $\bar{\boldsymbol{\omega}}_i = (\bar{\omega}_{i,1}, \bar{\omega}_{i,2}, \dots, \bar{\omega}_{i,N})$, en la iteración t .

En esta propuesta se define la probabilidad inercial, P_{ω}^t , como la probabilidad de cambio del vector $\boldsymbol{\omega}_i^t$ en la iteración t . P_{ω}^t decrece exponencialmente con el número de iteraciones.

Los bits del vector de inercia ω_i^t se obtienen de la siguiente manera: la probabilidad de que $\omega_{i,j}^t = '0'$ es P_ω^t , es decir, si P_ω^t es alta es más probable que $\omega_{i,j}$ sea '0' y si P_ω^t es baja probablemente $\omega_{i,j}$ sea '1'. Por tanto, en las iteraciones iniciales hay más probabilidad de que $\omega_{i,j}$ sea 0 y en las últimas iteraciones la posibilidad de que $\omega_{i,j}$ sea 1 es mayor.

El valor de P_ω^t en la iteración t se obtiene mediante la aplicación de las siguientes ecuaciones:

$$P_\omega^t = P_{\omega,\max} \cdot \exp\{-\lambda_p \cdot (t-1)\} \quad (7.5)$$

Siendo:

$$\lambda_p = \frac{\ln\left(\frac{P_{\omega,\max}}{P_{\omega,\min}}\right)}{(t_{\max} - 1)} \quad (7.6)$$

Donde:

- $P_{\omega,\max} \equiv$ Valor máximo de la probabilidad de cambio del vector $\omega_i = (\omega_{i,1}, \omega_{i,2}, \dots, \omega_{i,N})$.
- $P_{\omega,\min} \equiv$ Valor mínimo de la probabilidad de cambio del vector $\omega_i = (\omega_{i,1}, \omega_{i,2}, \dots, \omega_{i,N})$.
- $t_{\max} \equiv$ Número de iteraciones máximas establecidas.

En las iteraciones iniciales el valor de P_ω^t es elevado, el algoritmo explora el espacio de búsqueda de forma aleatoria. En las últimas iteraciones el valor de P_ω^t es pequeño y el algoritmo aprovecha el conocimiento y experiencia adquiridos por la nube de partículas en las anteriores iteraciones. En este caso, el movimiento de la partícula se calcula principalmente por los valores de **mejorpos_i** (la mejor posición encontrada por

la partícula i) y de **mejorposg** (la mejor posición encontrada por toda la nube de partículas).

Finalmente, el algoritmo para codificación binaria que se propone queda de la siguiente manera:

ALGORITMO PSO BINARIO PROPUESTO
<p>$t = 0$</p> <p>$Nube \leftarrow$ Inicializar Nube de Partículas</p> <p>Mientras no se alcance la condición de parada hacer</p> <p style="padding-left: 20px;">$t = t + 1$</p> <p style="padding-left: 20px;">Para $i = 1$ hasta tamaño ($Nube$) hacer</p> <p style="padding-left: 40px;">Evaluar cada partícula x_i de la $Nube$</p> <p style="padding-left: 40px;">Si $aptitud_x_i$ es mejor que $aptitud_mejorpos_i$ entonces</p> <p style="padding-left: 60px;">$mejorpos_i \leftarrow x_i$;</p> <p style="padding-left: 60px;">$aptitud_mejorpos_i \leftarrow aptitud_x_i$</p> <p style="padding-left: 40px;">Fin Si</p> <p style="padding-left: 40px;">Si $aptitud_mejorpos_i$ es mejor que $aptitud_mejorposg$ entonces</p> <p style="padding-left: 60px;">$mejorposg \leftarrow mejorpos_i$;</p> <p style="padding-left: 60px;">$aptitud_mejorposg \leftarrow aptitud_mejorpos_i$</p> <p style="padding-left: 40px;">Fin Si</p> <p style="padding-left: 20px;">Fin Para</p> <p style="padding-left: 20px;">Para $i = 1$ hasta tamaño ($Nube$) hacer</p> <p style="padding-left: 40px;">$\mathbf{v}_i^t = \bar{\omega}_i^{t-1} + \omega_i^{t-1} \cdot (\mathbf{c}_{1i} \cdot (\mathbf{mejorpos}_i \oplus \mathbf{x}_i^{t-1}) + \mathbf{c}_{2i} \cdot (\mathbf{mejorposg} \oplus \mathbf{x}_i^{t-1}))$</p> <p style="padding-left: 40px;">$\mathbf{x}_i^t = \mathbf{x}_i^{t-1} \oplus \mathbf{v}_i^t$</p> <p style="padding-left: 40px;">Fin Para</p> <p>Fin Mientras</p> <p>Salida: Devuelve la mejor solución encontrada.</p>

7.6 MODIFICACIONES REALIZADAS EN EL ALGORITMO PSO

Básicamente el algoritmo propuesto en el apartado anterior es muy parecido al de Afshinmanesh [2] con la salvedad de que se ha incorporado un término inercial típico de la versión clásica del algoritmo.

Tal y como se ha definido, la probabilidad inercial, P_{ω}^t , provoca que en las primeras iteraciones el algoritmo explore de forma prácticamente aleatoria en todo el espacio de búsqueda. Conforme avanza el número de iteraciones el algoritmo conoce mejor el entorno y busca de forma más intensa y ordenada en función del aprendizaje que ha adquirido en las primeras iteraciones. La aleatoriedad de la búsqueda se va perdiendo en las últimas iteraciones en las que domina la experiencia adquirida por la nube de partículas.

El movimiento de las partículas en el algoritmo de Afshinmanesh está conducido por los valores de *mejorpos_i* y *mejorposg* desde la primera iteración, por tanto, no explora con tanta intensidad. Este comportamiento provoca que la nube de partículas confluya pronto a máximos locales que no siempre son los óptimos, sobre todo en problemas complejos.

En este sentido se puede considerar el algoritmo propuesto como un híbrido que intenta conjuntar las ventajas de la versión clásica del PSO y de la versión de Afshinmanesh y obtener así un algoritmo con un mayor rendimiento.

En definitiva, con las mejoras introducidas se pretende que inicialmente el algoritmo explore más, a medida que avanza el número de iteraciones el nivel de exploración aleatoria se reduzca y aumente la intensificación del algoritmo en el conocimiento del entorno adquirido. Todo esto redundará en una mayor eficiencia, eficacia y robustez del algoritmo.

Las mejoras se demuestran en el siguiente capítulo donde en algunos casos se comparan los resultados obtenidos por las dos versiones del algoritmo para codificación binaria.

7.7 FUNCIONAMIENTO DEL ALGORITMO

Recordando lo expuesto en el capítulo 3, la partícula está asociada a tres vectores y dos valores de *aptitud*. De los tres vectores el primero, $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,N})$, se encarga de almacenar la posición actual de la partícula, el segundo, $\mathbf{mejorpos}_i = (mejorpos_{i,1}, mejorpos_{i,2}, \dots, mejorpos_{i,N})$, guarda la posición de la mejor solución encontrada por la partícula hasta el momento y el tercero, el vector velocidad, $\mathbf{v}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,N})$, indica la dirección según la cual se moverá la partícula.

Estos tres vectores están compuestos por cadenas de bits en un número igual a la dimensión del problema, N , definida en la ecuación (7.1) y que es dependiente de los tamaños máximos del eje X y del eje Y del mapa considerado (L_X y L_Y) y de la magnitud máxima del área de extracción de la biomasa, o lo que es lo mismo, de la apotema máxima definida.

Los valores de los componentes X e Y, y la apotema de una partícula no proporcionan únicamente la ubicación o posición de la planta y de las K parcelas sino que llevan ligados los valores de todas las variables asociadas a cada una de estas parcelas (densidad de biomasa, factor de utilización de la biomasa, coste de recogida, tipo de residuo, composición y propiedades energéticas del residuo, distancia a la línea eléctrica más próxima, etc.).

Los valores de *aptitud* son $aptitud_x_i$ que almacena el valor de adecuación de la posición actual correspondiente al vector \mathbf{x}_i y $aptitud_mejorpos_i$ que almacena el valor

de adecuación de la partícula con mejor solución encontrada hasta el momento correspondiente al vector **mejorpos_i**. Si se elige el índice de rentabilidad, *IR*, como la función objetivo, el valor de *aptitud_{x_i}* coincide con el valor que toma *IR* considerando que la ubicación de la planta queda definida por los componentes X e Y del vector **x_i** y la biomasa que se extrae se obtiene aplicando los valores que tiene las variables asociadas a cada una de las parcelas que componen las *K* parcelas seleccionadas definidas por el valor de la apotema. En cada iteración, y para cada partícula que forma parte de la nube, hay que realizar todo el análisis técnico-económico descrito en el capítulo 6 para obtener el *IR*, el vector **mejorpos_i** toma el valor de la partícula que posee mejor *IR*.

Si la nube está formada por *P* partículas, para cada iteración del algoritmo se deben de realizar *P* análisis o cálculos completos de *IR*, esto implica el cálculo de la biomasa utilizable de las *K* parcelas seleccionadas, la energía generada conforme al modelo SOFC-TG, los ingresos recibidos por la venta de energía, el cálculo de los diferentes costes de extracción, operación y mantenimiento, la inversión necesaria en función del tamaño de la planta, etc., y todas estas cantidades referidas a valores actuales.

Considerando lo anterior, el número de veces que se calcula el modelo para obtener un valor del *IR* es el que se obtiene de multiplicar el número de iteraciones por el número de partículas, (*P* · *N*). Si por ejemplo, se parte para la resolución de un problema de este tipo de 20 partículas y el número de iteraciones se fija en 50, el número de procesos de cálculo asciende a 1000, bastante inferior a los que se necesitan realizar con un método exacto.

El punto de partida de cualquier simulación es establecer los datos de entrada del algoritmo. Los parámetros relacionados con el funcionamiento del sistema SOFC-TG son los siguientes:

- Tensión de funcionamiento de la pila.

- Coeficiente de utilización del combustible.
- Coeficiente de mantenimiento de la planta en tanto por uno.
- Costes de operación fijos.
- Coste de operación variable unitario.
- Inversión necesaria por unidad de potencia.
- Inversión inicial fija.
- Tiempo de funcionamiento anual estimado de la planta.
- Vida útil de la planta estimada.

Los parámetros relacionados con las características dimensionales del mapa son:

- Tamaño del mapa y superficie total.
- Número de parcelas que lo componen.
- Superficie de la unidad parcelaria.

Los datos que poseen cada una de las parcelas que componen el mapa son los siguientes:

- Las coordenadas de la parcela.
- La densidad de residuos forestales.
- El coeficiente de utilización de la biomasa residual.
- La composición del residuo, dependiente del tipo de especie o vegetación predominante de la parcela, determina las cantidades de los gases combustibles que se generan en el proceso de gasificación. La cantidad de hidrógeno equivalente depende de $R_{gas,i}$ y de la biomasa residual utilizable, BRU_i .
- El poder calorífico inferior de la biomasa utilizable.
- El coste unitario de recogida de biomasa.
- El coste de transporte unitario de la biomasa en función de la accesibilidad a la parcela.
- La distancia real con respecto a todas las parcelas.

- La longitud existente entre la parcela y el punto de enganche más cercano de la red de distribución.

Otros parámetros de entrada que se necesitan para el análisis económico son:

- El interés anual del préstamo bancario.
- La duración del préstamo bancario.
- La tasa de descuento.
- Los índices estimados del incremento anual de los costes unitarios de recogida, de transporte, de mantenimiento y de operación.
- Los incentivos concedidos a la inversión.
- El precio de venta de la energía eléctrica generada a la empresa distribuidora.
- El coste unitario de la línea eléctrica necesaria para la conexión de la planta a la red eléctrica.

Antes de iniciarse, el algoritmo PSO propuesto necesita conocer los valores de:

- El número de partículas que conforman la nube.
- El número de iteraciones.
- Los valores máximos y mínimos de la probabilidad inercial.

Una vez ejecutado el algoritmo los resultados que se obtienen y de los cuáles se pueden extraer conclusiones son los siguientes:

- El valor de la función objetivo elegida IR y VAN .
- Las coordenadas de la parcela donde se ubica la planta.
- Las coordenadas de cada una de las parcelas que comprenden el área de suministro de la biomasa.
- El valor del área de suministro de la biomasa.
- La potencia eléctrica del sistema SOFC-TG.
- La evolución de la función objetivo en función del número de iteraciones del

algoritmo.

La descripción del proceso algorítmico adaptado al problema descrito es la siguiente:

- 1) El algoritmo comienza con la carga de las bases de datos que contienen los valores de todas las variables que están vinculadas con cada una de las parcelas del mapa o región a estudio. Estos datos se suelen obtener de herramientas SIG apropiadas.
- 2) Se inicializan de forma aleatoria los valores (componentes X e Y y la apotema en código binario) de los P vectores correspondientes a las partículas que componen la nube. Cada vector tiene N bits.
- 3) Con estos primeros valores se inicia el proceso iterativo y se establece como condición de parada un número de iteraciones, t .
- 4) Para cada iteración se evalúa la *aptitud* de cada una de las partículas, es decir, se calcula el índice de rentabilidad, IR . Para ello, se utilizan los valores que llevan asociados las parcelas definidas por cada partícula.
- 5) Se actualizan los valores de *aptitud* $_x_i$ y *aptitud* $_{mejorpos_i}$ en función de los resultados obtenidos en el punto anterior. Este es el mejor resultado obtenido hasta el momento por la partícula x_i .
- 6) Si el nuevo valor de *aptitud* $_{mejorpos_i}$ es el mejor encontrado hasta el momento, se actualizan el valor de la mejor posición global de la nube *mejorpos* y su *aptitud* $_{mejorpos}$. Este es el mejor resultado alcanzado hasta el momento por todas las partículas de la nube.
- 7) Se generan los vectores de velocidades de acuerdo a la ecuación (7.3).

- 8) Al vector de posición \mathbf{x}_i se le añade el vector velocidad \mathbf{v}_i para obtener un nuevo vector \mathbf{x}_i .
- 9) Con las nuevas posiciones de las partículas de la nube se reinicia el proceso iterativo del punto 4 hasta que se llegue a la iteración máxima establecida.

El tamaño de la nube de partículas es muy importante, ya que determina el equilibrio entre la calidad de las soluciones obtenidas y el número de iteraciones necesarias para alcanzar una buena solución en un tiempo computacional bajo. La determinación óptima del tamaño de la nube de partículas y el número de iteraciones del algoritmo se ajusta en función de los resultados de las primeras simulaciones de cada caso.

7.8 VALIDACIÓN, ROBUSTEZ E IMPLEMENTACIÓN

El algoritmo PSO desarrollado es una técnica metaheurística que no da la garantía total de encontrar la solución óptima pero sí una muy aproximada, sin embargo, el tiempo computacional se reduce ostensiblemente con respecto a la utilización de técnicas exactas. Para determinados problemas con un número elevado de parcelas la resolución con métodos exhaustivos podría llegar a ser inviable siendo la única herramienta posible la utilización de técnicas metaheurísticas apropiadas y adaptadas al problema.

El método propuesto no se puede validar a partir de resultados experimentales, ya que no se han encontrado publicaciones que compartan un planteamiento tan extenso y global, y menos con la especificación de un sistema de generación concreto. En las referencias consultadas [56, 65, 139, 162], los datos de partida proceden de bases de datos pertenecientes a Sistemas de Información Geográfica correspondientes a regiones reales. Estos datos no figuran en la publicación, únicamente, en algunos casos se

representan las regiones a estudio en mapas con poca resolución en los que se suele representar la biomasa residual disponible o el potencial energético de esa biomasa. Se intuye que en estos trabajos el cálculo de la mejor solución se realiza mediante técnicas exactas y la complejidad del problema se reduce aplicando algunas simplificaciones como la limitación de la distancia máxima de recogida de los residuos, la evaluación de tan sólo unas pocas ubicaciones de la planta, la fijación de la potencia de generación, etc.

No obstante, los resultados obtenidos con el algoritmo PSO propuesto se pueden comparar con los resultados alcanzados mediante el uso de otras técnicas metaheurísticas que posean un gran grado de madurez y sean suficientemente reconocidas, contrastadas y avaladas por la comunidad investigadora en diversos trabajos. La técnica elegida son los algoritmos genéticos que se describen brevemente en el apéndice II. Los algoritmos genéticos han demostrado en sus más de 30 años de vida una gran versatilidad, eficiencia, eficacia y robustez.

El planteamiento del método de resolución con algoritmos genéticos no sufre grandes cambios y la codificación de la partícula es la misma que la descrita y empleada para el algoritmo PSO.

No obstante, el primer modelo desarrollado en el capítulo anterior que considera que las características técnicas-energéticas del área de suministro de biomasa son constantes puede servir como una referencia inicial para comparar los valores que alcanzan las funciones económicas que, si bien, son aproximados, nos permite comprobar que los resultados son muy parecidos en condiciones y con datos de partida similares.

Uno de los criterios más importantes en la evaluación de cualquier nueva propuesta de algoritmo, es la calidad del algoritmo y su capacidad de generar soluciones similares cuando se ejecuta varias veces. Este factor puede ser determinante en el caso de las técnicas metaheurísticas.

La robustez de un algoritmo es la propiedad que mide el porcentaje de aciertos del mismo, por lo que un algoritmo válido en el 100 % de los casos, diremos que es un algoritmo totalmente robusto. No todos los algoritmos cumplen la condición de robustez total, sin embargo, si la robustez del algoritmo es relativamente alta puede resolver la mayor parte de los casos planteados. Un algoritmo eficaz, eficiente y con un alto grado de robustez garantiza que la mayoría de soluciones calculadas se consideren buenas.

El paquete de software que se ha utilizado para la implementación y programación del algoritmo PSO propuesto así como del resto de análisis y simulaciones previstas es MATLAB versión 7.1 de la Compañía The Mathworks, Inc.

En el apéndice I se presenta el programa principal del algoritmo PSO para codificación binaria utilizado.

7.9 CONSIDERACIONES PARA EL MODELO GLOBAL CON RESTRICCIONES

Por último, considerando el modelo global planteado en el apartado 6.5 del capítulo anterior en el que se incluyen restricciones técnicas de la red de distribución (se limita la potencia máxima de generación y el rango de tensiones en los nudos) y se considera el valor actual de las pérdidas económicas dependientes directamente de las pérdidas energéticas producidas en la red de distribución de la empresa distribuidora, el funcionamiento del algoritmo es prácticamente el mismo exceptuando que en el cálculo de la *aptitud* de cada partícula y para cada iteración t se realiza lo siguiente:

- 1) El cálculo correspondiente del flujo de cargas para obtener las pérdidas de energía en la red teniendo en cuenta que el sistema SOFC-TG se conecta al nudo más cercano de la red de distribución.

- 2) Se comprueba que la potencia del sistema SOFC-TG es inferior a la máxima potencia establecida.
- 3) Se comprueba que el rango de tensiones en todos los nudos está dentro de unos parámetros aceptables.

Si los valores que definen a una partícula x_i proporcionan una potencia superior a la máxima y/o alguna de las tensiones de los nudos de la red no entran en el rango de valores establecidos la solución de esta partícula se descarta.