



UNIVERSIDADE DA CORUÑA

DEPARTAMENTO DE ENXEÑERÍA INDUSTRIAL II

TESIS DOCTORAL

**Modelización de procesos degradativos a partir de datos  
termogravimétricos**

Autora: Matilde Ríos Fachal

Directores:

Ramón Pedro Artiaga Díaz

Jorge José López Beceiro

Ferrol 2013



UNIVERSIDADE DA CORUÑA

DEPARTAMENTO DE ENXEÑERÍA INDUSTRIAL II

TESIS DOCTORAL

**Modelización de procesos degradativos a partir de datos  
termogravimétricos**

Autora: Matilde Ríos Fachal

(Rios-Fachal, M.)

Directores:

Ramón Pedro Artiaga Díaz

Jorge José López Beceiro

Ferrol 2013

*A Lucía y Salvador*

# Índice general

<b>Agradecimientos</b>	<b>I</b>
<b>Resumen</b>	<b>III</b>
0.1. Resumen . . . . .	III
0.2. Resumen . . . . .	V
0.3. Abstract . . . . .	VII
<b>Prólogo</b>	<b>IX</b>
0.4. Prólogo . . . . .	IX
0.5. Objetivos . . . . .	X
<b>1. Introducción</b>	<b>1</b>
1.1. Introducción al Análisis Térmico . . . . .	1
1.1.1. Técnicas de Análisis Térmico . . . . .	1
1.1.2. Técnica TG (Análisis Termogravimétrico) . . . . .	2
1.1.3. Instrumentos para medir la degradación de un material . . . . .	2
1.2. Aplicación a diferentes materiales . . . . .	3
1.2.1. Un ejemplo clásico: el oxalato de calcio . . . . .	5
1.2.2. Aplicación a maderas industriales . . . . .	5
1.2.3. Aplicación a nanocompuestos epoxi-nanosílice . . . . .	6
<b>2. Modelización</b>	<b>11</b>
2.1. Introducción a la modelización . . . . .	11
2.1.1. Etapas del ciclo de vida de un modelo . . . . .	11
2.1.2. Modelos de regresión . . . . .	13
2.1.3. Modelos no paramétricos . . . . .	15
2.2. Modelos cinéticos . . . . .	16
2.2.1. Modelos cinéticos tipo Arrhenius . . . . .	18
2.2.2. Cálculo de los parámetros de la ecuación de Arrhenius . . . . .	19
2.2.3. Método de Friedman . . . . .	20
2.2.4. Otros modelos para la cinética . . . . .	21
2.3. Modelos de mezcla de logísticas . . . . .	22
2.3.1. Método de funciones logísticas generalizadas . . . . .	23
2.3.2. Aplicaciones del método de ajuste logístico . . . . .	25

2.3.3.	Ventajas del modelo de mezcla de logísticas . . . . .	25
<b>3.</b>	<b>Optimización</b> . . . . .	<b>29</b>
3.1.	Introducción . . . . .	29
3.2.	Métodos clásicos de optimización . . . . .	31
3.2.1.	Algoritmo de Brute Force . . . . .	31
3.2.2.	Algoritmo de Newton-Raphson o de Newton-Fourier . . . . .	31
3.2.3.	Algoritmo de Gauss-Newton . . . . .	32
3.3.	Optimización en análisis cinético . . . . .	32
3.3.1.	Inconvenientes de los métodos clásicos . . . . .	34
3.4.	Algoritmos Evolutivos . . . . .	34
3.4.1.	Breve reseña histórica . . . . .	34
3.4.2.	Principios de un Algoritmo Evolutivo . . . . .	36
3.4.3.	Desarrollo de un algoritmo en codificación binaria . . . . .	40
3.4.4.	Convergencia del Algoritmo . . . . .	42
3.4.5.	Descripción del método . . . . .	43
3.4.6.	Fundamentos matemáticos de los AG. Teorema de los esquemas . . . . .	46
3.5.	Selección de Algoritmos Evolutivos para AT . . . . .	50
3.5.1.	Algoritmo Evolución Diferencial (Differential Evolution) . . . . .	50
3.5.2.	Algoritmo CMA-ES . . . . .	51
3.5.3.	Algoritmo Simulated Annealing . . . . .	52
3.5.4.	Otros algoritmos metaheurísticos . . . . .	53
3.6.	Aplicaciones de Algoritmos Evolutivos . . . . .	54
3.6.1.	El algoritmo genético estadístico (AGE) . . . . .	54
3.6.2.	El Algoritmo Genético de Selección de Multirrecombinación, (MRS) . . . . .	57
3.6.3.	Combinaciones con otros algoritmos evolutivos . . . . .	58
3.6.4.	Aplicación de Algoritmos evolutivos en el ajuste de controladores del tipo PID . . . . .	58
3.6.5.	Optimización para Descubrir Motifs en Secuencias de ADN . . . . .	59
<b>4.</b>	<b>Algoritmo memético para optimización de curvas TG</b> . . . . .	<b>61</b>
4.1.	Introducción . . . . .	61
4.2.	Aplicación de Algoritmos Evolutivos para la estimación de parámetros del modelo logístico . . . . .	62
4.3.	Propuesta de un método para la optimización . . . . .	63
4.4.	Estudio de simulación . . . . .	65
4.4.1.	Algoritmos de optimización globales . . . . .	66
4.4.2.	Propuesta de un método para optimizar los parámetros de ajuste . . . . .	67
4.4.3.	Generación de datos . . . . .	68
4.4.4.	Resultados de las simulaciones . . . . .	70
4.5.	Conclusiones sobre el método propuesto . . . . .	75

<b>5. Modelización de la degradación de un nanocompuesto de PS con nanotubos de carbono</b>	<b>77</b>
5.1. Introducción . . . . .	77
5.2. Nanocompuesto de matriz polimérica y nanotubos de carbono . .	78
5.3. Análisis termogravimétrico del nanocompuesto . . . . .	78
5.3.1. Introducción al estudio termogravimétrico del PS . . . . .	78
5.3.2. Datos experimentales . . . . .	79
5.3.3. Modelización de la curva TG del PS-carbono . . . . .	81
5.3.4. Resultados de la degradación del PS con nanotubos . . . .	88
<b>6. Conclusiones y líneas futuras</b>	<b>93</b>
6.1. Conclusiones y objetivos alcanzados . . . . .	93
6.2. Líneas futuras . . . . .	94
<b>7. Anexos</b>	<b>97</b>
7.1. Anexo I: Cálculos para el modelo mezcla de logísticas . . . . .	97
7.1.1. Propiedades de las derivadas . . . . .	97
7.2. Anexo II: Propiedades relativas a la Epistasis . . . . .	102
7.2.1. Definición y cálculo de la <i>epistasis</i> . . . . .	102
7.2.2. Cálculo de la epistasis para funciones de segundo orden .	103
7.2.3. Cálculo de valores extremos de la <i>epistasis</i> . . . . .	103
7.3. Anexo III: Implementación de sentencias en R . . . . .	104
7.3.1. Introducción al programa R . . . . .	104
7.3.2. Simulación de un modelo logístico generalizado con R . .	105
7.3.3. Cálculo de máximos y mínimos con algoritmos genéticos en R . . . . .	106
7.4. Anexo IV: Implementación en R de algoritmos evolutivos . . . .	108
7.4.1. Algoritmo memético: DEoptim y Simulated Annealing . .	109
7.4.2. Aplicación a nanotubos . . . . .	118
<b>Bibliografía</b>	<b>121</b>



# Agradecimientos

Deseo hacer mención de mi agradecimiento hacia algunas personas e instituciones que con su ayuda me han permitido llevar a término este proyecto.

En primer lugar, a mis directores de tesis Ramón Artiaga y Jorge López Becero por su apoyo y aliento en todo momento.

También quiero mostrar mi agradecimiento a los profesores del Departamento de Matemáticas Javier Tarrío-Saavedra por su inestimable apoyo y ayuda en el desarrollo de las aplicaciones en R, a la profesora María Teresa Iglesias por haberme iniciado en los estudios sobre algoritmos genéticos y al profesor Salvador Naya por sus correcciones y ánimo en todo este camino.

Asimismo, quiero dar las gracias a los compañeros del grupo de investigación Propiedades Térmicas y Reológicas de Materiales (PROTERM) por la ayuda y accesibilidad prestada en todo momento.



# Resumen

## 0.1. Resumen

La cinética de reacciones y en especial la búsqueda de algoritmos de optimización adecuados es un tema de gran interés en Análisis Térmico. El objetivo principal que se ha planteado en esta tesis es desarrollar nuevos métodos capaces de resolver estos problemas con el menor tiempo computacional posible, mejorando la calidad de los resultados obtenidos por los métodos ya existentes. Para ello combinamos varios algoritmos de optimización de tipo evolutivo.

En esta memoria se presenta una alternativa a la estimación del ajuste de modelos paramétricos tradicionalmente utilizados en los estudios cinéticos de datos procedentes del análisis térmico. El problema planteado consiste en ejecutar un método de optimización alternativa para evaluar y ajustar curvas de análisis térmico, más específicamente curvas termogravimétricas (TG) y sus primeras derivadas. La propuesta es la estimación de los parámetros óptimos correspondientes a los modelos cinéticos de ajuste aplicado a las curvas de termogravimetría (TG), utilizando algoritmos evolutivos: la evolución diferencial (DE), el recocido simulado (SA) y la estrategia evolutiva basada en la matriz de covarianzas (CMAES). Este procedimiento no necesita incluir un vector con los valores iniciales de los parámetros, como se requiere en otros métodos. A pesar de sus beneficios potenciales, la aplicación de estos métodos no suele ser habitual en el contexto de la estimación de las curvas de análisis térmico.

La metodología que se sugiere en este trabajo se ha aplicado a diferentes datos procedentes de análisis térmico. Para verificar los métodos se simuló varias curvas TG y se ajustaron mediante un modelo de mezcla de logísticas generalizadas, en las que cada componente logística representa un proceso individual de degradación térmica. La simulación de las curvas de TG se hizo en cuatro escenarios diferentes reflejando la superposición de procesos, lo que nos permite la evaluación de los resultados finales y por lo tanto validar el procedimiento propuesto en diferentes contextos: dos procesos de degradación no solapada usando dos logísticas generalizadas, dos procesos superpuestos, cuatro procesos no solapados y cuatro procesos solapados dos a dos. La medida del ajuste se hizo con el estudio del error cuadrático medio que se escogió como función objetivo y los algoritmos anteriores fueron aplicados por separado y en cadena, es decir, tomando la solución final de un algoritmo como solución

inicial del siguiente. Los resultados muestran que los algoritmos evolutivos proporcionan una buena solución para el ajuste de las curvas de TG simuladas, mejor que la proporcionada por los métodos tradicionales.

Finalmente, se aplican estos algoritmos recomendados al estudio de un material compuesto de poliestireno (PS) con nanotubos de carbono (CNT). Se ha demostrado que la introducción de nanotubos de carbono en una matriz de polímero afecta a la estabilidad térmica de los materiales compuestos. Aunque los efectos observados son más complejos de lo que en principio cabía esperar, la tendencia general es un aumento en la temperatura de degradación y la reducción en la velocidad de degradación, cuantificada como la disminución de la masa con el tiempo. Se analizan los efectos de los nanotubos en la degradación del poliestireno.

El propósito de este estudio es evaluar cómo influyen los CNT en el proceso de degradación principal de dichos nanocompuestos con matriz de poliestireno y conteniendo 2, 3 y 5 % los nanotubos de carbono. Para este fin se llevaron a cabo experimentos termogravimétricos, en atmósfera de nitrógeno, en múltiples rampas de calentamiento lineales.

Se han empleado diferentes técnicas para la optimización de los parámetros cinéticos haciendo uso de un modelo recientemente desarrollado, el de mezcla de funciones logísticas generalizadas, que se adapta al contexto termogravimétrico. El modelo permite separar el proceso principal de otros subprocesos, lo que facilita el análisis cinético de cada proceso individual, y permite también restar la masa residual, que podría producir un efecto estabilizador aparente.

El principal proceso de degradación está claramente estabilizado por la presencia de nanotubos, aunque la estabilización es más pronunciada en el más bajo de los contenidos de carga considerados. Se observaron claros efectos de los nanotubos sobre los parámetros cinéticos.

## 0.2. Resumo

A cinética de reaccións e especialmente o problema da búsqueda de algoritmos de optimización axeitados é un dos temas de máis interese na Análise Térmica. O principal obxectivo desta tese é o desenvolvemento de novos métodos ou algoritmos que permitan resolver estes problemas co mínimo tempo computacional, mellorando a calidade dos resultados obtidos por outros métodos existentes. Para facelo, combinaráanse varios algoritmos de optimización, concretamente de tipo evolutivo.

Nesta memoria preséntase unha alternativa á estimación do axuste de modelos paramétricos tradicionalmente utilizados nos estudos cinéticos a partir de datos procedentes da análise térmica. O problema é propoñer un método de optimización para avaliar e axustar as curvas do análise térmico máis especificamente curvas termogravimétricas (TG), e as súas primeiras derivadas. Esta proposta implica a estimación dos parámetros correspondentes aos modelos cinéticos aplicados para axustar as curvas de termogravimetria, utilizando algoritmos evolutivos: evolución diferencial (DE), recocido simulado (SA) ou estratexia evolutiva con adaptación da matriz de covarianzas (CMAES). Neste procedemento non é necesario incluír un vector que conteña os valores iniciais dos parámetros, como se require en outros métodos. A pesar dos beneficios potenciais da aplicación destes métodos non é usual a súa aplicación para estimar curvas de análise térmica.

A metodoloxía proposta neste traballo foi aplicada en diferentes escenarios de análise térmica. Para comprobar os métodos simuláronse distintas curvas TG e axustáronse mediante un modelo de mestura de loxísticas xeneralizadas, onde cada compoñente supón un proceso de degradación térmica. Simuláronse curvas TG en catro escenarios diferentes reflexando procesos superpostos, o que nos permite valorar os resultados finais e, polo tanto, validar o método proposto en diferentes contextos: dous procesos superpostos de degradación usando dous modelos de loxísticas xeneralizadas con dous procesos de superposición e sen superposición e outros escenarios con catro procesos de superposición, con superposición dous a dous e sen superposición. A medida do axuste fíxose para o erro cadrado medio que foi escollida como función obxectivo a minimizar, e os algoritmos foron comprobados por separado e encadeados, ou sexa, a solución final de un deles úsase como solución inicial do seguinte. Os resultados mostran que os algoritmos evolutivos proporcionan unha boa solución para axuste de curvas TG simulados, mellores ca os proporcionados polos métodos tradicionais.

Finalmente aplicaranse estes algoritmos recomendados ao estudo de un material composto de poliestereno (PS) con nanotubos de carbono (CNT). Demostrouse que a introdución de nanotubos de carbono (CNT) nunha matriz polimérica afecta á estabilidade térmica dos materiais compostos. Aínda que os efectos observados son máis complexos do que en principio se esperaba, a tendencia xeral é un aumento na temperatura de degradación e na redución da velocidade de degradación, cuantificada como a diminución da masa co tempo.

O propósito deste estudo é avaliar cómo inflúen os CNT no proceso de degradación principal de ditos nanocompostos con matriz de poliestireno e que

contéñen un 2, 3 e 5% dos nanotubos de carbono. Para este fin leváronse a cabo experimentos termogravimétricos, con atmósfera de nitróxeno, en múltiples rampas de quentamento lineares. Avalíanse os efectos dos nanotubos na degradación do poliestireno.

Empregáronse diferentes técnicas para a optimización dos parámetros cinéticos facendo uso dun modelo recentemente desenrolado, o de mestura de funcións loxísticas xeneralizadas, que se adapta ao contexto termogravimétrico. O modelo permite separar o proceso principal doutros subprocesos, o que facilita o análise cinético de cada proceso individual, e permite tamén restar a masa residual, que podería producir un efecto estabilizador aparente.

O principal proceso de degradación está claramente estabilizado pola presenza de nanotubos, aínda que a estabilización é máis pronunciada no máis baixo dos contidos de carga considerados. Observáronse claros efectos dos nanotubos sobre os parámetros cinéticos.

### 0.3. Abstract

The kinetics of reactions and in particular the finding of optimization algorithms is one of the most investigated in Thermal Analysis. The main objective which has been raised in this thesis is to develop new methods and algorithms that can solve these problems with the least possible computational time, improving the quality of the results obtained by existing methods. To do this we combine several evolutionary algorithms.

This thesis presents an alternative to simple estimation of parametric fitting models used in thermal analysis. The addressed problem consists of performing an alternative optimization method to evaluate and fit thermal analysis curves, specifically TG curves and their first derivatives. This proposal consists of estimating the optimal parameters corresponding to fitting kinetic models applied to thermogravimetric (TG) curves, using Evolutionary algorithms: differential evolution (DE), simulated annealing (SA) and covariance matrix adapting evolutionary strategy (CMAES).

This procedure need not include a vector with the initial values of the parameters, as it is currently required. In spite of their potential benefits, the application of these methods are by no means usual in the context of thermal analysis curves estimation. So, this is a novel proposal in this thesis.

The methodology proposed in this work has been implemented in thermal analysis problems. Simulated TG curves are obtained and fitted using a generalized logistic mixture model, where each logistic component represents a thermal degradation process.

The simulation of TG curves in four different scenarios reflecting the extent of overlapping processes, allow us to evaluate the final results and thus to validate the proposed procedure: two non overlapped degradation processes were simulated using two generalized logistics, two overlapped processes, four non overlapped processes and four overlapped processes two by two. The mean square error function is chosen as the objective function and the above algorithms have been applied separately and together, i.e. taking the final solution of the DE algorithm is the initial solution of the remaining methods. The results show that the evolutionary algorithms provide a good solution for adjusting simulated TG curves, better than that provided by traditional methods.

Finally, we apply these algorithms to the study recommended a polystyrene composite with carbon nanotubes. It has been shown that introducing carbon nanotubes (CNT) into a polymer matrix has a beneficial effect on thermal stability of composites. While the specific effects noted differ depending on many parameters, the general trend is an increase in the degradation temperature and reduction in the degradation rate, quantified as the decline in mass over time.

The purpose of this study is to evaluate how CNTs influence the main degradation process of composites made with polystyrene (PS) containing 2, 3 and 5 % of CNTs. Thermogravimetric experiments are performed, with nitrogen purge, at multiple linear heating ramps.

The effects of the nanotubes on the degradation of polystyrene are evaluated. Insightful kinetic parameters were obtained for the main process making use of

a recently developed model, which is adapted to the thermogravimetric context. The model allows the means to separate the main process from other processes, which could interfere with the kinetic analysis, and also subtract the residual mass, which could produce an apparent stabilizing effect.

The main degradation process is clearly stabilized by the presence of nanotubes, although the stabilization is more pronounced at the lowest of the filler contents considered. Clear effects of nanotubes on kinetic parameters were observed.

# Prólogo y objetivos

## 0.4. Prólogo

El objetivo de esta tesis es proponer soluciones para la modelización y algoritmos de optimización para la resolución de problemas en análisis térmico de materiales, entre las que están el estudio de algunos modelos que formalizan matemáticamente datos de estudios en laboratorio, concretamente el comportamiento de variables usadas en análisis termogravimétrico de materiales.

Con este fin, se utilizarán diferentes modelos cuyos parámetros será preciso estimar, para lo que habrá que diseñar o adaptar distintos métodos del Optimización. Concretamente, se abordará el problema de modelización mediante diferentes propuestas paramétricas de mezcla de funciones no lineales y se pondrán los mejores métodos para la optimización de los parámetros, con la combinación de métodos basados en algoritmos evolutivos.

La memoria consta de siete capítulos. En el primero de ellos se presenta una introducción a los diferentes problemas de Análisis Térmico de materiales y se introducen algunas de las técnicas empleadas en este campo y, finalmente, se enfatiza la importancia que tiene la modelización de las curvas mostrando ejemplos de experimentos de procesos degradativos en los que ha trabajado el grupo de Propiedades Térmicas y Reológicas (PROTERM) analizados en la Escuela Politécnica Superior

En el segundo capítulo se analiza el tema de la modelización, para lo que se presentarán diferentes propuestas de modelos matemáticos, en concreto los modelos cinéticos basados en el de Arrhenius y los modelos de mezcla de funciones logísticas.

En el capítulo III se abordará el problema de la optimización de parámetros analizando las diferentes técnicas de optimización paramétrica, centrándose en el caso de modelos de tipo no lineal y en la optimización basada en el empleo de algoritmos evolutivos, con especial mención a aquellas técnicas que resultan más apropiadas para el ajuste de los datos procedentes de estudios termogravimétricos de materiales.

En el capítulo IV se propondrá un algoritmo memético secuencial basado en la aplicación de diferentes algoritmos de optimización. La idea de esta nueva propuesta se basa en buscar algoritmos que optimicen los parámetros de los modelos propuestos. El método propuesto será analizado con detalle mediante

la propuesta de simulaciones en diferentes escenarios que describen situaciones complejas en análisis térmico, como es el caso de varios procesos solapados.

En el capítulo V se realiza el estudio de un nuevo material, un nanocompuesto a base de poliestireno reforzado con nanotubos de carbono. Se aplican los algoritmos propuestos y se analizan las propiedades térmicas de este compuesto.

El capítulo VI recoge las conclusiones de mayor relevancia a las que se ha llegado con este estudio y se proponen nuevas líneas de investigación a partir de las conclusiones.

Finalmente, en el capítulo VII se incluyen cuatro anexos. En el primero se presentan los desarrollos matemáticos que permiten analizar las propiedades del modelo de mezcla de logísticas, en el segundo se exponen algunos cálculos sobre la epistasis, que mide la idoneidad de los algoritmos evolutivos. Los anexos III y IV incluyen el código desarrollado para la implementación de los modelos y algoritmos en el programa R; las primeras librerías que se presentan están orientadas a la simulación y las últimas a la modelización y optimización.

## 0.5. Objetivos

El objetivo principal de esta tesis consiste en combinar tres aspectos clave dentro de la modelización de datos de la degradación de materiales por termogravimetría (TG), que son la modelización, la optimización y su aplicación a problemas concretos de análisis térmico.

En los últimos años se han presentado diferentes modelos que resuelven el ajuste de datos del análisis térmico, como los modelos cinéticos basados en modificaciones del modelo de Arrhenius o los modelos de mezcla de funciones logísticas. Sin embargo, el problema de encontrar los parámetros óptimos de estos modelos no está todavía resuelto. Un objetivo de esta investigación es la propuesta de nuevas metodologías para la optimización de estos parámetros basados en algoritmos evolutivos.

Otro importante objetivo está relacionado con la aplicación de estas nuevas propuestas metodológicas a problemas concretos de modelización de las curvas TG obtenidas experimentalmente.

Estos objetivos generales de la investigación podrían dividirse en objetivos específicos, que marcaron también las fases de la investigación, y son los siguientes:

- **1.** Definir, formular y resolver el problema de la optimización de los parámetros de curvas TG.
- **2.** Diseñar y analizar nuevos algoritmos metaheurísticos específicos, debido a que éstos son una buena forma de abordar problemas de optimización en los que no es posible la aplicación de técnicas de optimización exactas. Se pretende analizar algunos de los algoritmos conocidos como los de la evolución diferencial (DE), el llamado de recocido simulado (SA) y el basado en la estrategia evolutiva de matriz de covarianza con adaptación (CMAES).

- **3.** Crear un algoritmo memético que permita hibridar distintos algoritmos. El potencial de las técnicas híbridas junto con el conocimiento de los requerimientos del problema tratado, es un objetivo que puede permitir crear nuevos algoritmos más eficientes.
- **4.** Realizar estudios de simulación para verificar la eficiencia de los métodos propuestos, planteando distintos escenarios.
- **5.** Aplicar estas técnicas a problemas concretos del campo del análisis térmico de materiales.
- **6.** Transferir y difundir el conocimiento adquirido. Con el objetivo de difundir el contenido del trabajo realizado en esta investigación se proponen diferentes métodos de transferencia. Además de las publicaciones aparecidas en revistas y congresos tanto nacionales como internacionales, se propone la inclusión de estos algoritmos en librerías de Software libre como el R en la web de dominio público del grupo de Propiedades Térmicas y Reológicas de la Universidade da Coruña (PROTERM), de modo que permitan aplicar las aportaciones de este trabajo a otros conjuntos de datos.



# Capítulo 1

## Introducción

### 1.1. Introducción al Análisis Térmico

Puede decirse que el Análisis Térmico es el conjunto de técnicas mediante las cuales, una propiedad física o química de un material es medida en función de la temperatura o del tiempo (Turi, 1997). Por tanto, será esta herramienta la que se usará para estudiar propiedades térmicas de los materiales a través de los procesos degradativos al ser sometidos a un programa térmico.

Para este objetivo se suelen utilizar termobalanzas que proporcionan medidas de pérdida de masa del material en función de la temperatura o del tiempo. Estos instrumentos permiten salidas tanto gráficas como numéricas que deben de ser convenientemente interpretadas y modelizadas.

Los estudios actuales en el campo del Análisis Térmico se apoyan en el manejo de potentes aparatos de laboratorio. Éstos permiten obtener series de puntos que relacionan algún tipo de variable dependiente (flujo de calor, pérdida de masa, etc...) con la temperatura o el tiempo. Los resultados obtenidos de estos experimentos, proporcionan una gran cantidad de datos que requieren de un adecuado modelado a fin de interpretar adecuadamente las distintas gráficas y hacer una lectura correcta de las mismas, que es uno de los objetivos de este trabajo.

#### 1.1.1. Técnicas de Análisis Térmico

Las técnicas de Análisis Térmico suelen clasificarse en función del instrumento que se utilice. Para ello se acostumbra a nombrarlas mediante sus siglas:

-TA (Thermal Analysis): grupo de técnicas con las que se mide una propiedad física de un sustancia como función de la temperatura o el tiempo, mientras es sometida a un programa térmico controlado.

- DTA (Differential Thermal Analysis): técnica que mide la diferencia de temperatura entre la muestra y un material de referencia como función de la temperatura, mientras ambas son sometidas a un programa térmico controlado.

-DSC (Differential Scanning Calorimetry): técnica que mide la diferencia de flujo de calor entre una muestra y un material de referencia como función de la temperatura mientras ambos son sometidos a un programa térmico controlado.

-TGA (Thermal Gravimetric Analysis): técnica en que la masa de una muestra se mide en función de la temperatura o el tiempo mientras es sometida a un programa térmico controlado en una atmósfera específica.

### 1.1.2. Técnica TG (Análisis Termogravimétrico)

Serán las técnicas TG las más utilizadas en esta memoria al permitir estudiar la degradación de la masa de las muestras del material mientras está siendo sometida a un programa térmico en atmósfera controlada. A esta técnica también se le denomina termogravimetría, llamándose termogramas a las gráficas resultantes, que denominaremos con el mismo nombre de la técnica, es decir, TG.

Estas curvas TG pueden obtenerse utilizando un método dinámico, si se estudia el comportamiento de la muestra mientras es sometida a un programa de temperaturas con una determinada velocidad de calentamiento; mientras que si se mantiene fija la temperatura y se estudia la variación de la masa con el tiempo, se hablará de método isoterma. Siempre es posible reflejar la pérdida de masa de la muestra, lo que se llama también camino de degradación del material.

Además de las curvas TG, también puede resultar de interés para el análisis de la degradación del material, ver la velocidad con que ésta se produce, para lo que es interesante usar la curva que representa la velocidad de variación de la masa, que no es más que la derivada de la curva TG, y que se denomina DTG (Derivative Termogravimetry) y que además de la cinética del proceso, también permite determinar puntos críticos, como los instantes de velocidad máxima de pérdida de masa. En ocasiones, la modelización de la primera derivada puede venir acompañada del estudio de derivadas sucesivas; la segunda, que indica puntos de inflexión en la curva original y otras posteriores. Este problema será también abordado desde la perspectiva de búsqueda del mejor método para la optimización de las derivadas sucesivas.

### 1.1.3. Instrumentos para medir la degradación de un material

Uno de los instrumentos clásicos de un laboratorio que se usa para la medida de la degradación de la masa de un material con la temperatura es el calorímetro. Hoy en día, estos aparatos están compuestos, básicamente, por una termobalanza, un horno y un procesador de temperaturas y disponen de un circuito de gas de purga (normalmente aire o  $NO_2$ ) y conectado se usa un ordenador, con su correspondiente programa de procesado de datos, que permite el estudio directo de éstos.

Además de la técnica TG es frecuente el uso simultáneo con aparatos que permiten medir también el flujo de calor, conocidos como técnicas DSC. Así,

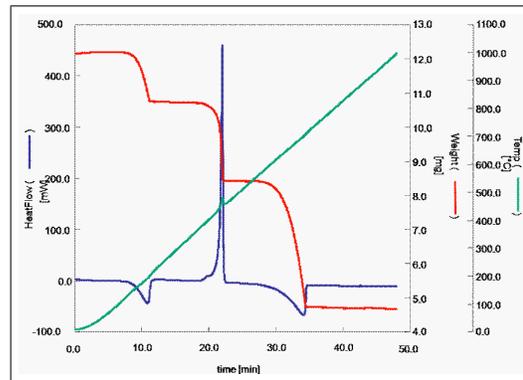


Figura 1.1: Oxalato de calcio analizado mediante una técnica STA, en que se aprecia la curva TGA (rojo) y la DSC (azul). En verde se presenta la rampa lineal de temperaturas aplicada.

existen instrumentos que combinan ambas técnicas, como el STA, que presenta la ventaja de visualizar las dos gráficas en la misma salida. Puede verse un ejemplo del empleo de esta técnica en la Figura 1.3, en la que se presentan las curvas DSC y TG (camino de degradación del material), obtenidas a partir de una muestra de oxalato de calcio en un aparato STA. Se observa cómo cada uno de los escalones que presenta la gráfica TG, se corresponde con un cambio experimentado por la muestra al degradarse y también se ve cómo la curva DSC tiene ciertos valores extremos que están correlacionados con cambios en la degradación de la curva TG.

En la figura 1.2 se recoge la foto de un calorímetro de tipo TG, de la marca TA, con el detalle de la termobalanza (abierta en el centro). En la foto de la izquierda de dicha figura se muestra el detalle del calorímetro sin su tapa de protección en el que se aprecia el horno en el que va implementada la balanza.

## 1.2. Aplicación a diferentes materiales

Debe mencionarse que este trabajo de investigación se enmarca dentro del que realiza el grupo de Propiedades Térmicas y Reológicas de Materiales (PRO-TERM) de la Universidad de A Coruña, por tanto, hace uso de muchos de los estudios previos desarrollados por miembros de este grupo, especialmente por aquellos en los que participan los directores de esta tesis y que han dado lugar a que se plantee esta investigación, ya que el tema concreto de la optimización de parámetros no estaba resuelto.

Este grupo cuenta con una amplia experiencia en la modelización de procesos en Análisis Térmico de Materiales y algunos de los casos estudiados por miembros de este laboratorio son los ejemplos que se utilizan en esta memoria. Por ello se ha considerado oportuno mostrar algunas de estas casuísticas como

Figura 1.2: En la imagen se muestra un aparato TG en el que se aprecia el detalle de la termobalanza.



ejemplos en este capítulo introductorio.

Entre las investigaciones de este grupo en las que se apoya este trabajo hay que destacar la propuesta de alternativas a los modelos clásicos tipo Arrhenius basada en un modelo paramétrico de mezcla de logísticas (Naya et al. 2003 y Naya, 2011). Estos modelos y algunas modificaciones posteriores, como las logísticas generalizadas, han sido aplicados con éxito a diferentes materiales, como el las resinas epoxi (Cao et al. 2004 y López-Beceiro, 2011). Este método de mezcla de logísticas se utilizó para problemas más concretos, como separar componentes y otras características de interés dentro del análisis térmico de materiales (Artiaga et al. 2005).

La propuesta de este tipo de modelos ha permitido determinar características de interés en el estudio de nuevos materiales, entre las que están las siguientes: estudio de las propiedades de las resinas epoxi con nanoclays empleados en el recubrimiento de los depósitos de combustible de hidrógeno (Naya, Martínez Vilariño y Artiaga, 2009), aplicaciones al estudio de degradación del poliéster-poliuretano (Barbadillo et al. 2007), análisis de propiedades del nanosílice incorporado a las resinas epoxi (Tarrío-Saavedra et al. 2008, 2011), estudio de las propiedades de las maderas comerciales y su clasificación mediante los modelos funcionales propuestos (Tarrío-Saavedra et al. 2011a y Sebio-Puñal et al. 2012), propiedades específicas de la alumnita (López-Beceiro et al. 2011a), estudio térmico de la degradación de diferentes aceites comerciales (López-Beceiro

et al. 2011b), propiedades térmicas de biocombustibles (Artiaga et al. 2011) o nuevas estimaciones de características térmicas de nuevos nano-compuestos epoxi-nanosílice (Tarrío-Saavedra et al. 2012).

En la mayoría de estas modelizaciones ha surgido el problema de encontrar el modelo matemático que permita el mejor ajuste de los datos, para lo que ha sido preciso recurrir a diferentes métodos de optimización, habitualmente con funciones no lineales, que no siempre encuentran de forma fácil la solución óptima. Siendo éste un problema recurrente en la mayoría de estos estudios, se propuso como objetivo de esta investigación analizar con más detalle el problema concreto de la optimización numérica y encontrar un método que permita la búsqueda de soluciones óptimas en este contexto. Con este objetivo se abordarán en el siguiente capítulo los distintos tipos de optimización matemática, haciendo una especial mención al caso de los últimos algoritmos evolutivos propuestos, que han resultado ser los métodos que permiten un mejor modelizado de los datos analizados.

### 1.2.1. Un ejemplo clásico: el oxalato de calcio

En la Figura 1.3, se presentan las curvas DSC y TG frente al tiempo (camino de degradación del material), obtenidas a partir de una muestra de oxalato de calcio monohidrato en un instrumento STA. Se observa cómo cada uno de los escalones que presenta la gráfica TG, se corresponde con un cambio experimentado por la muestra al degradarse. Concretamente, en dicha Figura 1.3, puede verse cómo la muestra de oxalato de calcio se descompone en distintos procesos de degradación al ser sometida a un calentamiento lineal. Se aprecia que cada uno de los escalones de la curva TG está relacionado con algún cambio en el material; por ejemplo, el primer escalón corresponde a la pérdida de agua, el segundo a una pérdida de  $CO$ , en el tercero lo que desaparece es el  $CO_2$  y, al final del proceso, lo que ha quedado es óxido de calcio ( $CaO$ ). Este ejemplo permite ilustrar la importancia que tiene encontrar cada uno de los valores en que ocurren los cambios en el proceso, y sus puntos críticos, para lo que el adecuado ajuste de estas funciones será primordial.

En la Figura 1.4 se puede observar la modelización de una muestra de poliuretano, ajustada con un modelo de mezcla de funciones logísticas (en rojo el modelo sobre los datos originales en negro). En este caso la curva TG (en rojo) fue modelizada mediante la suma de cuatro componentes logísticas que pueden verse en diferentes colores en la gráfica (Barbadillo et al. 2007).

### 1.2.2. Aplicación a maderas industriales

En López-Becceiro et al. (2011a) se empleó un nuevo modelo de regresión compuesto por una suma de componentes logísticas generalizadas para ajustar curvas TG a diferentes maderas comerciales, de forma que se pudieran estudiar separadamente los procesos de degradación solapados correspondientes a una muestra de madera.

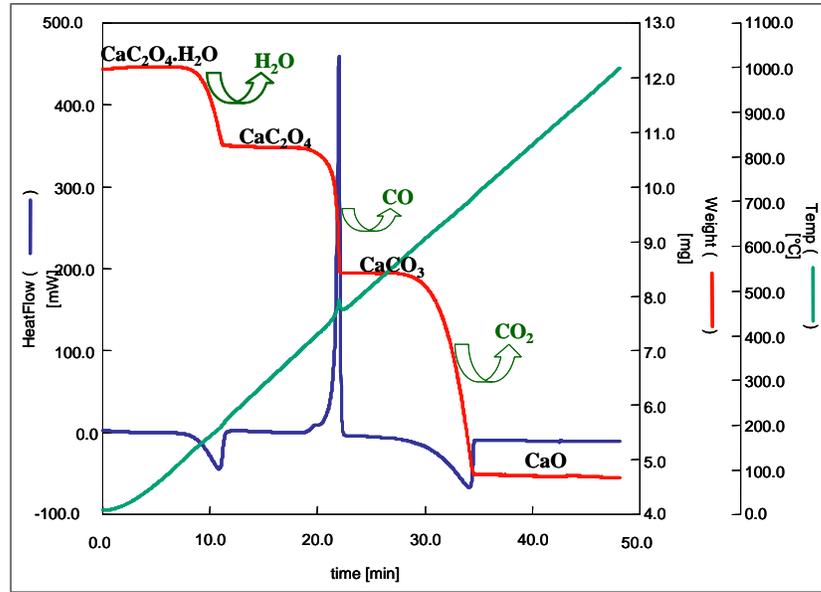


Figura 1.3: Curva TG (en rojo) y DSC (en azul) del oxalato de calcio.

Concretamente, se empleó un nuevo modelo de regresión compuesto por una suma de componentes logísticas generalizadas para ajustar curvas TG y sus derivadas, de manera que se pudieran estudiar separadamente los procesos de degradación de muestras de madera comercial. Como puede verse en la Figura 1.5 cada uno de los procesos se corresponde con la degradación de distintas componentes de la madera, como el agua, la lignina, la celulosa y la hemicelulosa. Es importante observar que estos materiales pueden relacionarse mediante una determinada curva logística, de ahí la importancia de la óptima modelización y optimización, pues los parámetros estimados mediante procesos numéricos, como los descritos en el capítulo IV de esta memoria permiten interpretar dicha degradación.

De este tipo de modelos de mezcla de logísticas generalizadas se sirve el artículo Tarrío-Saavedra et al. (2011), para seleccionar las características representativas de las curvas TG obtenidas a partir de muestras de madera. Además, la adecuada modelización de las curvas TG también se utilizó en Sebio-Puñal et al. (2012) para seleccionar las características representativas las curvas TG obtenidas a partir de muestras de madera y su posterior clasificación supervisada (Figura 1.6).

### 1.2.3. Aplicación a nanocompuestos epoxi-nanosílice

En la Figura 1.7 puede observarse la degradación de un material compuesto al que se añadieron diferentes porcentajes de humo de sílice. El material estaba

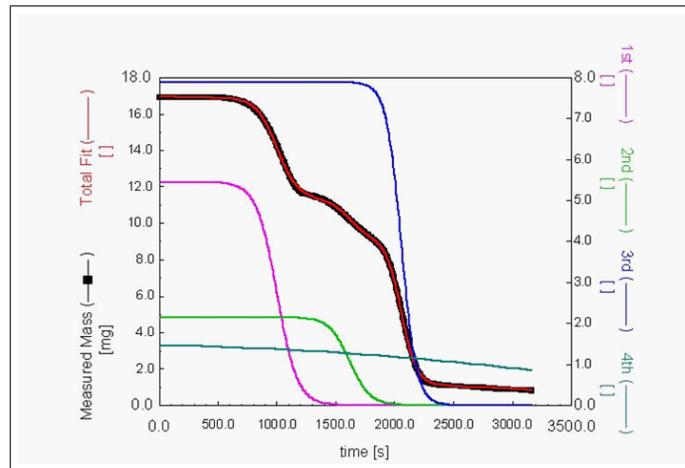


Figura 1.4: Modelización de una curva TG de poliuretano mediante un modelo de mezcla de cuatro funciones logísticas.

formado por una matriz polimérica epoxídica y una carga inorgánica de humo de sílice, que es un subproducto de la industria del silicio. Concretamente, se estudió su degradación térmica mediante análisis termogravimétrico en atmósfera oxidante (Tarrío-Saavedra et al. 2008, 2011).

Gracias a este estudio se estimó su estabilidad térmica calculando números índice clásicos, obtenidos a partir de las curvas TG, como la temperatura de descomposición inicial (IDT), la temperatura correspondiente a la velocidad máxima de pérdida de masa (Tmax) y la temperatura de descomposición por procedimiento integral (IPDT). El adecuado ajuste de los datos mediante mezcla de logísticas ha permitido sustraer matemáticamente, a cada curva TG, la masa del humo de sílice añadido (carga); dicha masa se correspondía con la masa inicial de sílice de los experimentos, pues el humo de sílice no se degrada a la temperatura programada. Por tanto, después de la sustracción de esta masa, se obtuvo el camino de degradación de, únicamente, la resina epoxi que forma parte del compuesto; si no se resta esta masa, la estabilidad térmica del compuesto aumenta aparentemente con el contenido en sílice. Sin embargo, después de la sustracción de la masa inorgánica, el camino de degradación térmica de la resina epoxi (curvas TG) se vio sólo muy ligeramente afectado por el contenido de sílice (Tarrío-Saavedra et al., 2008).

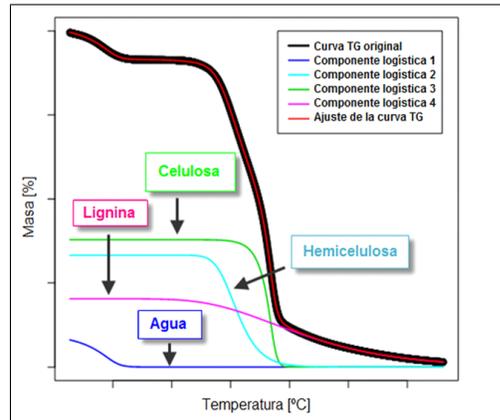


Figura 1.5: Descomposición de la degradación de una muestra de madera en sus componentes básicos, representadas como logísticas.

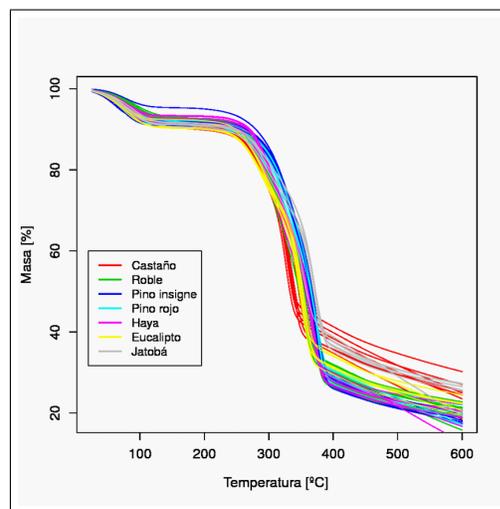


Figura 1.6: Curvas TG de muestras de diferentes maderas comerciales.

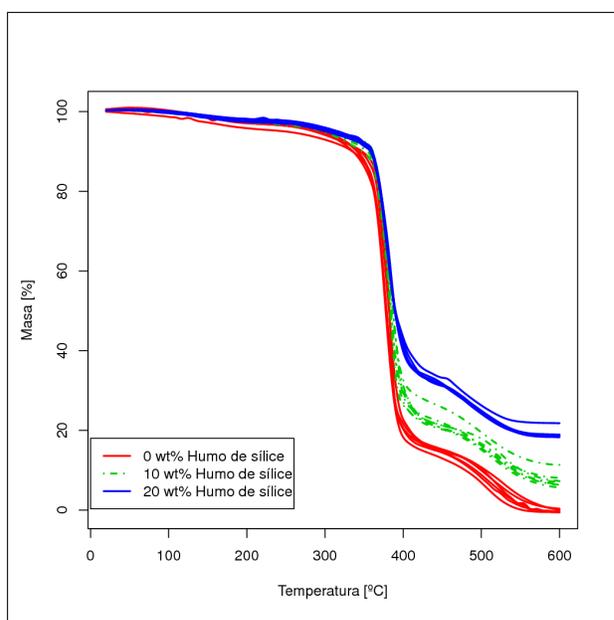


Figura 1.7: Curvas TG de nanocompuestos con diferentes porcentajes de humo de sílice.



# Capítulo 2

## Modelización

### 2.1. Introducción a la modelización

Un modelo es un esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja, que se elabora para facilitar su comprensión y el estudio de su comportamiento. (RAE).

Un modelo es, pues, una representación matemática simplificada de una realidad compleja, y modelar es la acción de construir un modelo. Esto implica la relación entre dos figuras (no necesariamente personas únicas, sino más bien equipos): el modelador (encargado de la especificación y desarrollo del modelo) y el experto sobre la realidad (conocedor del problema real). La mayoría de las veces, el desarrollo de un modelo puede involucrar a un equipo multidisciplinar compuesto por matemáticos, químicos, ingenieros, etc. que aportan diferentes perspectivas y conocimiento en la representación de la realidad. Un modelo debe equilibrar la necesidad de contemplar todos los detalles con la factibilidad de encontrar técnicas de solución adecuadas, y sus resultados han de ser inteligibles y útiles.

Modelar se puede entender simultáneamente como ciencia y como arte. Es una ciencia porque se basa en un conjunto de procesos estructurados: análisis y detección de las relaciones entre los datos, establecimiento de hipótesis y aproximaciones en la representación de los problemas y desarrollo o uso de algoritmos de optimización específicos de solución. Y es un arte porque materializa una visión o interpretación de la realidad, no siempre de manera unívoca, y cada persona imprime su estilo en el modelo mismo y en la especificación, desarrollo y documentación.

#### 2.1.1. Etapas del ciclo de vida de un modelo

En cualquier proceso de modelización se suelen distinguir las siguientes etapas:

- **Identificación del problema.** Consiste en la recolección y análisis de la información relevante para el problema, el intercambio de información entre

modelador y experto en una relación simbiótica y una estrecha coordinación entre ambos.

Los problemas reales suelen estar definidos en términos vagos e imprecisos y debe hacerse la traducción o interpretación en frases precisas, convertibles en ecuaciones matemáticas. En esta etapa se establecen y documentan los supuestos realizados que en etapas posteriores deberán ser validados. Es fundamental para que las conclusiones sean útiles y en consecuencia se tomen las decisiones correctas.

- **Especificación matemática y formulación** en “escritura matemática” del problema de optimización, definiendo sus variables, sus ecuaciones, su función objetivo y sus parámetros. En esta fase se analiza el tamaño del problema, su estructura, las características de sus restricciones y la clase de modelo. Es una etapa de creación y debe prestarse especial atención a la precisión en la formulación y a la escritura de las ecuaciones que describen el problema.

- **Resolución.** Es la implantación de un algoritmo de obtención de la solución numérica (muy próxima a la matemática) óptima o cuasióptima. Este algoritmo puede ser de propósito general o específico y puede haber diferentes métodos de solución de un problema o diferentes implementaciones de un mismo método. El tiempo de resolución puede depender drásticamente de la formulación.

- **Verificación, validación y refinamiento.** Consiste en la eliminación de errores de codificación, es decir, en verificar que el modelo haga lo que se le ha especificado matemáticamente en la etapa anterior mediante la escritura en lenguaje informático. Debe comprobarse la validez a través de los resultados obtenidos, contrastando con situaciones reales ya transcurridas (validar) o comprobando que son coherentes con la realidad. Esto da lugar a nuevas necesidades de refinamiento en el modelo para mejorar su capacidad de representación.

- **Interpretación y análisis de los resultados.** Es la propuesta de soluciones. Permite conocer en detalle el comportamiento del modelo al analizar los parámetros de entrada, estudiar diferentes escenarios plausibles de los parámetros, detectar soluciones alternativas cuasióptimas pero suficientemente aceptables y comprobar la robustez de la solución óptima.

La parte más difícil al usar las Matemáticas para estudiar una aplicación es la conversión de los fenómenos de la vida real al formalismo matemático. Por lo general esto es muy complicado porque implica la conversión de hipótesis imprecisas en fórmulas muy precisas.

Es importante recordar que los modelos matemáticos no dejan de ser modelos, es decir, el objetivo no es producir una copia exacta del objeto “real”, sino más bien representar algunas características de la realidad.

Una vez elaborado el modelo, debemos comparar las predicciones de éste con los datos del sistema. Si el modelo y el sistema concuerdan, tendremos confianza en que las hipótesis hechas al crear el modelo son razonables y que podemos usarlo para hacer predicciones.

Los tipos de predicciones que son razonables dependen de nuestras hipótesis. Si nuestro modelo se basa en reglas precisas como las leyes de Newton sobre el movimiento o las reglas del interés compuesto, entonces podemos usarlo para

hacer predicciones cuantitativas muy exactas; si las hipótesis son menos precisas o si el modelo es una versión simplificada del sistema, entonces sería absurdo tratar de obtener predicciones cuantitativas exactas.

Algunos autores han estudiado la relación entre la carga de trabajo de las distintas tareas y su posible relación con el grado de éxito final de un problema de modelización. Así, Shannon (1998) propuso la regla 40-20-40. Según ésta un 40 % del trabajo se debe dedicar a las tareas de formulación del problema y adquisición de datos; la implementación del modelo debería representar otro 20 % del trabajo y el 40 % restante las fases de verificación, validación, experimentación y análisis de resultados.

Para la construcción de un modelo han de tenerse en cuenta tres preceptos básicos:

- (1) Establecimiento claro de las hipótesis en las que se basará el modelo. Éstas deben describir las relaciones entre las cantidades a estudiar.
- (2) Definición completa de las variables y parámetros que se usarán en el modelo.
- (3) Usar las hipótesis formuladas en (1) para obtener ecuaciones que relacionen las cantidades de (2).

En (1), precepto científico, se describe cómo se cree que funciona el sistema físico/químico o, al menos, cuáles son sus aspectos más importantes, aunque a veces estas hipótesis son bastante especulativas. En (2) se nombran las cantidades que se van a estudiar y, en caso necesario, se describen las unidades y escalas implicadas. En nuestros modelos, las cantidades se agrupan en tres categorías básicas: la variable independiente (puede ser la temperatura o el tiempo en análisis térmico), las variables dependientes (son funciones de la variable independiente, como la masa de un material en función del tiempo) y los parámetros, que son cantidades que no cambian con el tiempo o con la variable dependiente pero que pueden ajustarse por causas naturales o por la ejecución de un experimento (por ejemplo la energía de activación en un modelo tipo Arrhenius).

### 2.1.2. Modelos de regresión

Desde la perspectiva del ajuste estadístico de datos, se denominan modelos de regresión a los que tratan de encontrar un algoritmo para estimar matemáticamente la relación entre una variable independiente (en nuestro caso el tiempo o la temperatura), denotada por la variable  $X$ , y una variable respuesta (en nuestro caso el peso o la diferencia de calor), denotada por la variable  $Y$ . Este problema se engloba en el contexto de un modelo general de regresión mediante la expresión  $Y = m(X) + \varepsilon$ , donde  $m$  es la función de regresión de  $Y$  dado  $X$ , y  $\varepsilon$  es una variable aleatoria que representa el error de observación. Así, podría decirse que el análisis clásico de regresión estudia las asociaciones cuantitativas entre una o varias variables explicativas  $X$  (aleatorias o no), y una variable

aleatoria respuesta o dependiente  $Y$ , de modo que, a partir de un conjunto de observaciones  $\{(X_i, Y_i)\}_{i=1}^n$ , la relación de regresión puede modelizarse como:

$$Y_i = m(X_i) + \varepsilon_i, \quad i = 1, 2, \dots, n, \quad (2.1)$$

donde  $m(\cdot)$  representa la función de regresión desconocida y  $\varepsilon$  el error de observación que se supone de media cero. Uno de los objetivos del análisis de regresión es obtener una aproximación razonable de la función desconocida  $m$ , reduciendo el error de las observaciones y centrándose en la respuesta media de  $Y$ , condicionada a  $X$ , lo que se puede expresar como una esperanza condicionada:

$$m(x) = E(Y|X = x). \quad (2.2)$$

En nuestro caso, al modelizar curvas procedentes de experimentos de análisis térmico, va a ser de interés aproximar ciertas características de la función de regresión como, por ejemplo, sus derivadas.

El proceso de estimar la función desconocida  $m(\cdot)$  se puede llevar a cabo de diferentes formas, en función del tipo de datos disponibles y del enfoque elegido, dando nombre a los modelos: diseño fijo o aleatorio, modelos paramétricos y no paramétricos, regresión de variables reales, vectoriales o funcionales.

### Modelos paramétricos

Un modelo de regresión paramétrico es aquel en que la curva de regresión teórica  $m$  pertenece a una familia indicada por un parámetro finito dimensional (por ejemplo, un polinomio). Una vez realizadas este tipo de suposiciones, las técnicas paramétricas se basan en utilizar la muestra para estimar los parámetros desconocidos (como los coeficientes del polinomio) usando, por ejemplo, el método de mínimos cuadrados.

El método de mínimos cuadrados consiste en encontrar los valores de aquellos parámetros que mejor ajustan la nube de puntos de los datos muestrales, minimizando la suma de las distancias al cuadrado entre los valores esperados ( $m(t_i)$ ) y los datos observados ( $y_i$ ), según la siguiente expresión:

$$\sum_{i=1}^n (y_i - m(t_i))^2. \quad (2.3)$$

La elección de  $m(t)$  estará marcada por el problema particular que se analice. Para cualquier especificación funcional:  $m(t) = m(a_0, a_1, a_2, \dots, a_p; t)$ , siendo los coeficientes  $a_0, a_1, a_2, \dots, a_p$ , los parámetros del modelo.

Una posible clasificación de los modelos paramétricos los divide en modelos lineales y no lineales, entendiéndose que la linealidad se observará en la expresión de la variable explicativa  $X$ .

### Modelos paramétricos lineales

Los modelos lineales son aquellos en que la media de la distribución de la variable  $Y$ , para cada valor  $x$  fijo, crece linealmente con  $x$ . Esta hipótesis

debe ser contrastada siempre, ya que condiciona la construcción del modelo. La utilidad de este modelo radica en que muchas relaciones no lineales se pueden convertir en lineales transformando las variables adecuadamente.

La solución al problema de estimación paramétrica de minimización de (2.3), está ampliamente estudiada, pudiendo expresarse el vector de parámetros estimados en forma matricial como:

$$\widehat{\mathbf{a}} = (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \mathbf{Y} \quad (2.4)$$

Un estudio detallado de este tipo de modelos puede verse en textos de estadística clásica, como en el libro de Peña (Peña, 1987).

### Modelos paramétricos no lineales

La mayoría de los modelos empleados en esta memoria son de tipo no lineal ya que la expresión de la relación de la variable  $Y$  no se adapta a la hipótesis de linealidad. Los fundamentos teóricos de estos modelos pueden verse en Gay (Gay, 1984).

La dificultad para muchos de estos modelos no lineales consiste en el cálculo de los parámetros por mínimos cuadrados, debido a que en ocasiones es necesaria la utilización de métodos numéricos que pueden no converger a la solución buscada, siendo necesario partir de unos valores iniciales para el algoritmo de aproximación. Uno de los métodos más usados, en estos casos, para el cálculo de los valores de los parámetros es el de Levenberg-Marquart, que trata de generar una secuencia de aproximación al punto mínimo basada en el algoritmo de la "trust region", siendo tratado en textos como el de Gay.

La suposición, a priori, de una cierta forma funcional de la curva desconocida es muy útil si la función realmente se ajusta a esa forma supuesta pero, en muchos casos, esto puede ser muy distinto de la realidad, con lo que se estarán cometiendo grandes errores; así, se podría estar ajustando una recta de regresión a una nube de puntos que no tengan relación lineal, sino parabólica.

### 2.1.3. Modelos no paramétricos

Los métodos de estimación no paramétrica de la regresión intentan estimar directamente la forma de la función de ajuste  $m$ , sin necesidad de especificar, a priori, su estructura. La filosofía de estos métodos se basa en la idea de dejar que los datos hablen por sí mismos.

Estas técnicas estiman la curva punto a punto, con lo que, en principio, presentan la desventaja del gran coste computacional que tienen, si bien esto es, hoy en día, un asunto menor debido al avance informático.

Las técnicas de estimación no paramétricas empiezan a usarse a partir de 1940, pero es a partir de 1980 cuando su expansión y uso se hacen extensivos. Publicaciones como las de Fan y Gijbels (1996), entre otras muchas, no hacen más que confirmar el desarrollo de esta metodología.

Los estimadores no paramétricos de la función de regresión suelen tener una estructura lineal en la variable respuesta. La idea de estos estimadores

no paramétricos lineales se basa en realizar promedios locales: si la función de regresión,  $m$ , se considera *suave*, entonces las observaciones con variable respuesta asociada con los  $X_i$  cercanos a un punto  $x$  contendrán información acerca del valor de  $m(x)$ , y su promedio, debidamente ponderado, podrá utilizarse para construir un estimador de  $m(x)$ :

$$\hat{m}_n(x) = \frac{1}{n} \sum_{i=1}^n W_{ni}(x) Y_i, \quad (2.5)$$

donde  $W_{ni}(x)$ ,  $i = 1, 2, \dots, n$ , es una sucesión de pesos que depende de las observaciones  $X_1, X_2, \dots, X_n$ .

Imponemos que los pesos,  $W_{ni}(x)$ ,  $i = 1, 2, \dots, n$ , sean positivos y estén normalizados para todo  $x$ , es decir,

$$\frac{1}{n} \sum_{i=1}^n W_{ni}(x) = 1,$$

así el estimador  $\hat{m}_n(x)$  se puede obtener como solución de un problema de mínimos cuadrados ponderado localmente, es decir, se puede expresar  $\hat{m}_n(x)$  como una solución del siguiente problema de minimización:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^n W_{ni}(x) (Y_i - \theta)^2. \quad (2.6)$$

Esta expresión, con promedios locales, es equivalente al procedimiento de calcular un estimador de mínimos cuadrados ponderados localmente.

El estimador  $\hat{m}_n(x)$  propuesto en (2.5), es un promedio de las observaciones en un entorno del punto de interés. El modo de promediar las observaciones está controlado por los pesos  $W_{ni}(x)$ ,  $i = 1, 2, \dots, n$ , los cuales, a su vez, dependen de un parámetro de suavizado que regula el tamaño del entorno de  $x$  en que se considerarán las observaciones  $X_i$  utilizadas para estimar  $m(x)$ . Si dicho parámetro de suavizado es grande, el promedio local se realizará con un número grande de observaciones y la curva obtenida estará *sobresuavizada*, dando lugar a un estimador,  $\hat{m}_n(x)$ , sesgado. Por el contrario, si el parámetro de suavizado es pequeño, podría ocurrir que el número de observaciones utilizadas para la construcción de  $\hat{m}_n$  fuera también muy pequeño, lo cual aumentará la varianza de  $\hat{m}_n$  y producirá una curva demasiado *rugosa*.

## 2.2. Modelos cinéticos

El estudio cinético de los materiales por termogravimetría dinámica se basa en llevar a cabo reacciones de descomposición térmica a temperatura variable, con el fin de analizar la velocidad a la que el material pierde peso. La pérdida de peso de la muestra puede considerarse como la suma de diferentes

procesos de descomposición isotérmicos, como queda de manifiesto en textos clásicos como el de Turi (Turi, 1997).

Uno de los objetivos del estudio cinético es conocer si existen uno o varios procesos y cuáles son las constantes de reacción características de éstos. La utilidad industrial de estos estudios es el aprovechamiento de los productos químicos que se puedan generar, el aprovechamiento energético, o simplemente, la eliminación de determinados materiales.

Al aumentar la velocidad de calefacción existe un desplazamiento de las curvas TG que ha sido ampliamente estudiado en la literatura (véase, por ejemplo, Prime 2007). Hay diversos argumentos para explicar estos desplazamientos. Algunos autores consideran que este desplazamiento puede ser explicado por las expresiones matemáticas de las leyes de la cinética. Obviamente, un modelo matemático que represente un conjunto de experimentos debe ser capaz de explicar tales desplazamientos a varias velocidades de calefacción (Conesa, 2000). Sin embargo, es posible ajustar las mismas curvas TG a diferentes expresiones, dependiendo del modelo cinético elegido, lo cual representa un serio problema.

En principio, los métodos cinéticos suelen clasificarse en endotermos e isotermos, con una o con varias velocidades de calentamiento. También cabe la clasificación en métodos derivativos o integrales, dependiendo del procedimiento que se emplee para el cálculo de los parámetros.

La utilización de la derivada de la curva, DTG, en lugar de la propia curva, tiene la ventaja de que un pequeño cambio en la TG se ve amplificado en su derivada. Otras ventajas de la termogravimetría dinámica, con respecto al estudio en procesos isotermos, según puede verse en trabajos como el de Carrasco y Costa (Carrasco y Costa, 1989), son las siguientes:

- Tiempos de manipulación inferiores permitiendo, con un solo experimento, determinar todos los parámetros cinéticos.
- Sólo se utiliza una muestra, con lo que se evitan los problemas inherentes al cambio de muestra necesarios en los procesos isotérmicos.
- La cinética global aparente puede determinarse en una gran gama de temperaturas de forma continua.
- En los procesos isotermos se requiere un precalentamiento previo, antes de alcanzar la temperatura de reacción. Durante este periodo transitorio pueden obtenerse conversiones no nulas, de forma que se falsearía el análisis cinético.
- Es posible determinar con precisión la temperatura umbral de descomposición.

Uno de los problemas de los métodos utilizados para el estudio cinético es la dificultad de ajustar los datos al modelo usado, normalmente de tipo Arrhenius.

### 2.2.1. Modelos cinéticos tipo Arrhenius

La forma clásica de estudiar la descomposición térmica de una muestra se basa en suponer que la velocidad de pérdida de peso de las reacciones de descomposición térmica depende de la masa y de la temperatura, y su expresión general sería:

$$m' = dm/dt = -f(m)k(T)q(m, T). \quad (2.7)$$

En la expresión (2.7) la función  $f(m)$  se toma como  $m_t^n$ , lo que comúnmente se interpreta diciendo que la pérdida de peso de la muestra obedece a una *cinética de orden  $n$* , donde  $m_t$  es la masa de la muestra en el tiempo  $t$ .

Esta suposición del orden de reacción es aplicable a reacciones homogéneas y a algunas reacciones heterogéneas que se llevan a cabo en fase condensada. Sin embargo, algunos polímeros que se degradan siguiendo un mecanismo de escisión aleatoria de enlaces, no poseen un orden de reacción constante, de forma que sería erróneo el uso de esta expresión (2.7), dado que los parámetros estarían sobreestimados (véase Conesa, 2000).

En todo caso, la hipótesis fundamental de la relación (2.7) se basa en suponer que la pérdida de masa sigue la Ley de Arrhenius representada por la ecuación siguiente:

$$k(T) = A \exp\left(-\frac{E_a}{RT}\right). \quad (2.8)$$

La ecuación de Arrhenius (2.8) expresa la variación de la velocidad de reacción ( $k$ ) con la temperatura ( $T$ ), y parte de la suposición de que la velocidad de descomposición del proceso aumenta exponencialmente con la temperatura. La constante  $A$ , llamada constante preexponencial, es independiente de la temperatura;  $E_a$  es la denominada energía de activación y  $R$  es la constante de los gases y proviene del papel que juega en la ley de los gases perfectos ( $pV = nRT$ ). El nombre se debe a que, en principio, esta ecuación fue propuesta por el químico sueco Svante August Arrhenius para la velocidad de reacción química en los gases.

Finalmente, el término  $q(m, T)$  de la expresión (2.7) indica la interacción entre la masa y la temperatura, y, aunque en muchos procesos se considera igual a uno, en ocasiones no se puede hacer esta suposición y su estimación puede resultar complicada. Estas interacciones entre masa y temperatura pueden estar originadas por los siguientes fenómenos (Carrasco y Costa, 1989):

- Modificación de las propiedades físicas de la muestra durante la descomposición térmica.
- Existencia de reacciones competitivas debidas a que la energía de activación y el factor preexponencial pueden variar con la conversión.

Existe algún otro método alternativo a la ley de Arrhenius, como el que se basa en suponer que la energía de activación  $E_a$  no es constante, sino que puede representarse por una distribución de probabilidad normal, como el presentado por Carrasco y Costa (1989).

### 2.2.2. Cálculo de los parámetros de la ecuación de Arrhenius

Con objeto de exponer los problemas que presenta el cálculo de los parámetros cinéticos, haremos a continuación una revisión de los modelos más utilizados para este estudio.

Para la determinación de los distintos parámetros existentes en la ecuación de Arrhenius (2.8), es necesario llevar a cabo algunas transformaciones, que consisten en expresar dicha ecuación en forma logarítmica:

$$\ln(k(T)) = \ln A - \frac{E_a}{R} \frac{1}{T}. \quad (2.9)$$

Esta ecuación (2.9) permite, representando  $\ln(k(T))$  frente al inverso de la temperatura absoluta ( $\frac{1}{T}$ ), calcular los parámetros deseados. En la práctica, basta con calcular la recta de regresión de  $\ln(k(T))$  frente al inverso de la temperatura absoluta ( $\frac{1}{T}$ ) para, al menos, dos experimentos (hay que determinar dos constantes de la recta). La pendiente de dicha recta es la estimación del cociente  $-\frac{E_a}{R}$ , mientras que la ordenada en el origen es una estimación de  $\ln A$ .

Habitualmente se usa este método para la determinación de la energía de activación a partir de la pendiente de la recta de regresión. Este hecho tiene el fundamento teórico en que la energía de activación, que tiene unidades de energía por mol, es posible expresarla mediante la propia ecuación de Arrhenius, sin más que dividir  $E_a$  y  $R$  entre el número de Avogrado ( $N_{AV}$ ), obteniéndose:

$$k(T) = C \exp(-q/kT), \quad (2.10)$$

donde  $q = E_a/N_{AV}$  es la energía de activación por unidad de escala atómica y  $k = R/N_{AV}$  es la constante de Boltzmann ( $13,8 \times 10^{24} J/K$ ).

Además, la expresión (2.10) permite efectuar una comparación con el extremo de alta energía de la distribución de Maxwell-Boltzmann de energías moleculares de los gases, dada por:

$$p \approx \exp\left(-\frac{\Delta E}{kT}\right). \quad (2.11)$$

La ecuación de Maxwell-Boltzmann (2.11) expresa la probabilidad  $p$  de encontrar una molécula con una energía  $\Delta E$  mayor que la energía media característica correspondiente a una temperatura,  $T$ . Este hecho es el que permite interpretar la energía de activación  $E_a$  como la barrera energética que debe ser superada con la activación térmica. Aunque la ecuación (2.11) se desarrolló inicialmente para gases, también es aplicable a sólidos.

#### Cálculo del orden de la reacción

En la expresión (2.7), una de las suposiciones más simples para la función  $f$  es suponer que  $f(m) = m_t^n$ , denominada cinética de orden  $n$ . Para el cálculo del orden de reacción,  $n$ , se supone que la masa en el instante  $t$ ,  $m_t$ , está relacionada con la conversión mediante la ecuación:

$$m_t = m_0(1 - \alpha). \quad (2.12)$$

Existen diferentes formas lineales para la expresión de la función  $f$ , destacando los métodos de Freeman y Carol, Newkirk, Horowitz y Metzger, Coats y Redfern, Vachuska y Voboril y Friedman. En las siguientes secciones se presentarán las ideas en las que se basan los más representativos.

### 2.2.3. Método de Friedman

A partir de experimentos realizados a varias velocidades de calentamiento, el método de Friedman se basa en la representación del valor  $\log \frac{d\alpha}{dt}$  frente a  $1/T$  para diferentes valores de  $\alpha$ ; se buscará el mejor ajuste lineal, ya que la pendiente de la recta es una estimación de  $E_a/R$ .

Cuando el análisis termogravimétrico se lleva a cabo en condiciones de programación lineal con la temperatura, ésta última y el tiempo de reacción están vinculados por la velocidad de calentamiento:

$$\beta = \frac{dT}{dt}. \quad (2.13)$$

Puesto que la masa en el instante  $t$ ,  $m_t$ , está relacionada unívocamente con la conversión, se sigue que:

$$f(m) = h(\alpha)m_0^n. \quad (2.14)$$

Teniendo en cuenta las ecuaciones (2.7, 2.8, 2.12 y 2.13), y previa separación de variables, se obtiene la expresión siguiente:

$$\int_{\alpha=0}^{\alpha} \frac{d\alpha}{h(\alpha)} = \frac{Am_0^{n-1}}{\beta} \int_{T(\alpha=0)}^T \exp\left(\frac{-E_a}{RT}\right) dT. \quad (2.15)$$

En la ecuación (2.15) la integral no posee primitiva analítica, pero utilizando el método de aproximación numérico de Coats-Redfern, basado en la expresión truncada del desarrollo en serie de Taylor, se obtiene como resultado:

$$\int_{T(\alpha=0)}^T \exp\left(\frac{-E_a}{RT}\right) dT = \frac{RT^2}{E_a} \left(1 - \frac{2RT}{E_a}\right) \exp\left(\frac{-E_a}{RT}\right). \quad (2.16)$$

Además, la integral que depende de la conversión en la ecuación (2.15), tiene una primitiva que puede expresarse como:

$$\int_{\alpha=0}^{\alpha} \frac{d\alpha}{h(\alpha)} = g(\alpha). \quad (2.17)$$

Introduciendo estos resultados (2.16 y 2.17) en la expresión (2.15), y linealizando la ecuación resultante, se llega a la expresión final siguiente:

$$\log \left[ \frac{g(\alpha)}{T^2(1 - 2RT/E_a)} \right] = \log \left[ \frac{A^*R}{\beta E_a} \right] - \frac{E_a}{R} \frac{1}{T},$$

donde  $A^* = Am_0^{n-1}$ .

Finalmente, mediante regresión lineal, como ya se dijo, puede calcularse la energía de activación y el factor preexponencial a partir de la pendiente y de la ordenada en el origen de dicha recta, respectivamente. Cabe destacar que para conocer la ordenada se necesita conocer la energía de activación, lo que precisa de un método iterativo.

Además, la linealización de la ecuación cinética engendra errores propios de la estimación de la recta de regresión. Una forma de atenuar dichos errores es usar un método de regresión lineal ponderada, en el que se aplica un factor o peso multiplicativo en los puntos experimentales.

Autores como Várhegyi (1987) afirman que los parámetros cinéticos son muy sensibles a los datos del principio y del final de la curva TG, mientras que su sensibilidad es mínima en las proximidades de la zona de máxima velocidad de reacción lo que justificaría utilizar pesos en el ajuste lineal ponderado del tipo de:

$$w = \left( \frac{\partial y}{\partial \alpha} \right)_T^{-2} = \left( \frac{g(\alpha)}{g'(\alpha)} \right)^2. \quad (2.18)$$

En cualquier caso, siempre puede ocurrir que dentro del material puedan tener lugar varios mecanismos simultáneamente, y cada mecanismo podría tener una energía de activación  $E_a$  característica, por lo que resulta difícil encontrar los parámetros para este tipo de modelos basados en la ley de Arrhenius.

Los diferentes métodos para el cálculo de dichos parámetros necesitan de aproximaciones numéricas, para integrales o derivadas, lo que permite clasificar estos métodos en integrativos y derivativos.

#### 2.2.4. Otros modelos para la cinética

Existen otros modelos para la cinética que, partiendo de la ecuación de Arrhenius, llegan a distintas expresiones. A continuación exponemos, someramente, algunos de los más destacables.

La mayoría de estas modificaciones de la ecuación general (2.7) pueden obtenerse a partir del modelo propuesto por Sesták y Berggren (1971), cuya forma general sigue la ecuación:

$$\frac{d\alpha}{dt} = k(\alpha)^m (1 - \alpha)^n [-\ln(1 - \alpha)]^p, \quad (2.19)$$

donde  $n$ ,  $m$  y  $p$  son constantes. Obsérvese que para  $m = p = 0$  se reduce a la expresión vista anteriormente para el cálculo del orden de reacción (2.12).

Burnham y Braun (1999) introducen un nuevo parámetro  $q$ , en la expresión (2.19) obteniendo una nueva variante, dada por:

$$\frac{d\alpha}{dt} = k((1-q)(1-\alpha))^m (1-\alpha)^n \quad (2.20)$$

Para el cálculo de los parámetros de las ecuaciones de los distintos modelos es necesario estimar valores mediante una regresión lineal. Así, hay autores que estiman estos números mediante el ajuste de  $\log(\alpha)$  frente a  $1/T$ , como el caso de Ozawa (1965), y en otros casos se opta por la estimación de estos parámetros mediante el método Kissinger, basado en obtener el ajuste mediante regresión lineal de  $\log(\beta/T_{\text{máx}}^2)$  frente a  $1/T_{\text{máx}}$ .

Otros modelos, de tipo derivativo, como los basados en la ecuación de Freeman y Carroll (1958), asumen la expresión  $f(\alpha) = (1-\alpha)^n$ , y buscan relacionar  $1-\alpha$  con  $1/T$  mediante la ecuación:

$$\Delta \ln \left( \frac{d\alpha}{dT} \right) = n \Delta \ln(1-\alpha) - \left( \frac{E_a}{RT} \right) \Delta \left( \frac{1}{T} \right). \quad (2.21)$$

Algunos autores critican este tipo de métodos basados en la suposición del modelo de Arrhenius y enumeran los siguientes inconvenientes de éstos (Prime, 2007):

- Se manipulan muchos datos obtenidos en el equipo, por lo que el reiterado abuso de logaritmos de logaritmos puede esconder los verdaderos datos de la señal TG.
- Estos métodos son válidos para descomposiciones que tengan lugar en un proceso simple. Si se observa más de un proceso, es necesario realizar una deconvolución “visual” antes de aplicar el método.
- En lugar de la relación experimental  $T = f(t)$ , se utiliza la velocidad de calefacción programada  $T = T_0 + \beta t$ , donde  $\beta$  es la velocidad real de calefacción: este hecho produce situaciones alejadas de la realidad, por cuanto que la temperatura experimental (leída por el termopar) puede desviarse puntualmente de la programada.
- La mayoría de estos métodos (todos los citados anteriormente excepto el de Friedman) utilizan sólo los datos obtenidos a una velocidad de calentamiento.

### 2.3. Modelos de mezcla de logísticas

Entre los modelos de tipo paramétrico que permiten un mejor ajuste de los datos TG están los que se basan en descomponer toda la curva TG en suma de diferentes funciones logísticas (modelo con mezcla de logísticas). La idea de esta modelización se basa en suponer que cada una de estas logísticas representa la cinética de descomposición de los distintos materiales simples de los que está compuesta la muestra inicial. La expresión matemática de este modelo es la siguiente:

$$\begin{aligned}
m(t) &= \sum_{i=1}^k w_i f(a_i + b_i t) \\
f(t) &= \frac{\exp t}{1 + \exp t},
\end{aligned} \tag{2.22}$$

donde  $w_i$  son parámetros que representan pérdidas de peso en cada proceso;  $a_i$  son parámetros de localización y  $b_i$  representa la velocidad de pérdida de peso, para los  $k$  escalones que presente la función  $m(t)$  en función del tiempo o la temperatura  $t$ .

La idea inicial parte de buscar una función que se ajuste lo mejor posible a los datos reales obtenidos para las curvas termogravimétricas. Las funciones candidatas a estimar estas curvas  $(t, (m(t)))$ , deben verificar que para valores grandes de  $t$  ( $t \rightarrow \infty$ ) las respuestas  $(m(t))$  deben tender a 0, lo que obliga a que los parámetros  $b_i$  tienen que ser negativos, mientras que al inicio del proceso ( $t = 0$ ) la función debe tender al peso de la muestra en cada momento; por tanto, los valores  $w_i$  representan, aproximadamente, las pérdidas de masa en el  $i$ -ésimo proceso. Los distintos procesos estarían marcados por los “escalones” de cada experimento (Naya et al., 2003; Cao et al., 2004 y Naya et al., 2006).

### 2.3.1. Método de funciones logísticas generalizadas

Una ampliación del método de logísticas consiste en considerar, en lugar de las logísticas simples, la versión de logísticas generalizadas que permiten incluir un parámetro adicional, lo que posibilita el ajuste en casos donde los procesos degradativos no sean simétricos.

Este modelo puede expresarse por la siguiente ecuación:

$$m(t) = \sum_{i=1}^k w_i g_i(t), \tag{2.23}$$

donde  $g_i(t)$  son las funciones logísticas generalizadas de la forma

$$g_i(t) = \frac{c_i}{(1 + \tau_i \exp(-b_i(t - m_i)))^{\frac{1}{\tau_i}}}, \tag{2.24}$$

en donde  $c_i$  representa el porcentaje de muestra involucrada en cada etapa de degradación,  $m_i$  la temperatura en la máxima razón de cambio,  $b_i$  está relacionado con la velocidad de cambio,  $\tau_i$  es la medida para la asimetría y  $t$  es la temperatura o el tiempo. Entonces, aunque  $m_i$ ,  $b_i$  y  $\tau_i$  son parámetros de ajuste, en ausencia de procesos con gran solapamiento,  $m_i$  puede ser fácilmente identificado como la temperatura en el pico de la derivada (DTG). Por otra parte,  $\tau_i$  está relacionado con un orden de reacción aparente,  $n$ , por  $n = 1 + \tau$ . (Véase López-Beceiro et al. 2011).

Como puede observarse en la Figura 2.1, dependiendo de los distintos parámetros del modelo se pueden conseguir efectos de forma, escala y simetrías para el adecuado modelizado de las curvas TG.

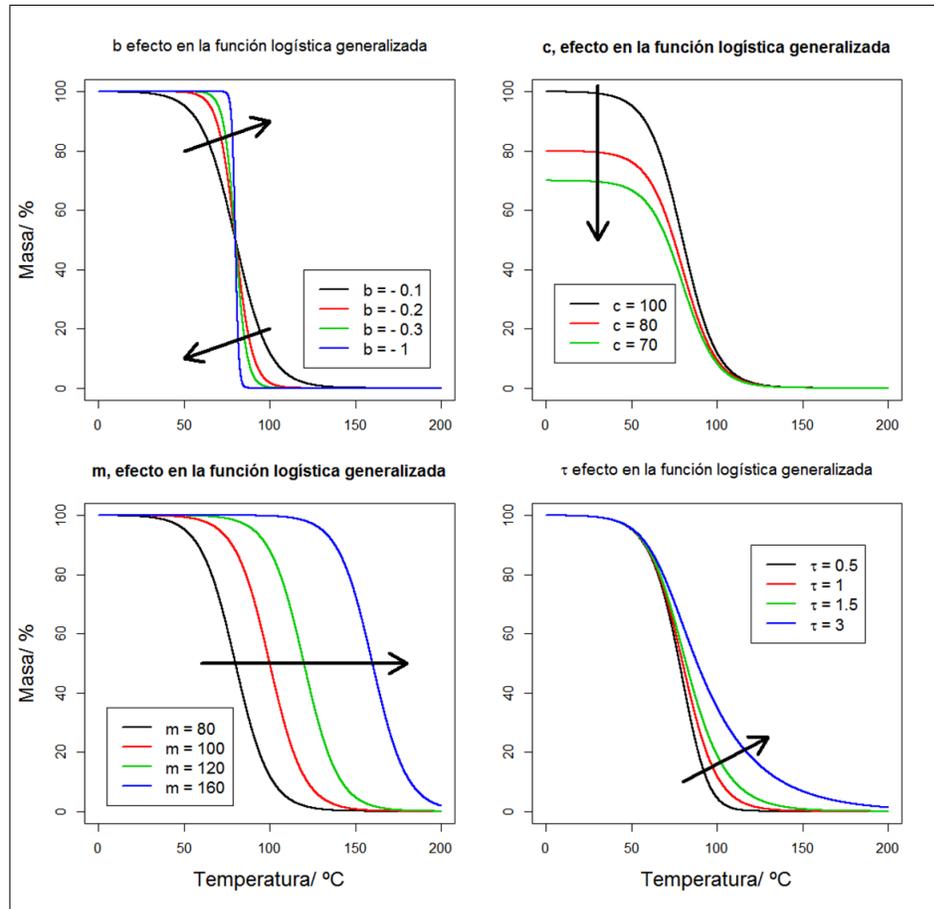


Figura 2.1: Efecto de los parámetros en el modelo de logísticas generalizadas.

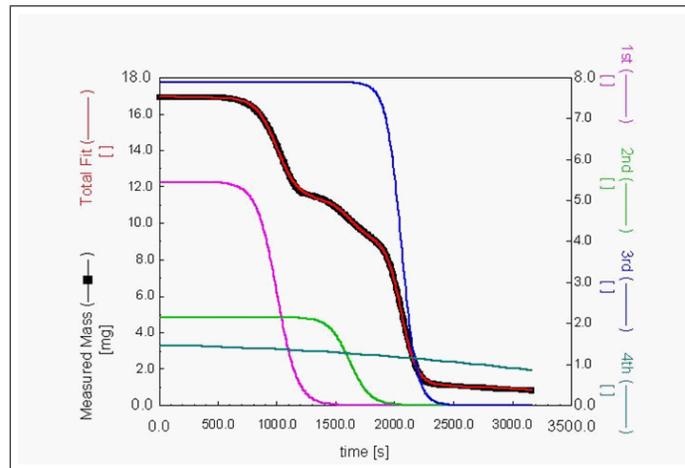


Figura 2.2: Modelización de una curva TGA mediante mezcla de funciones logísticas.

### 2.3.2. Aplicaciones del método de ajuste logístico

Vamos a analizar diferentes materiales para validar el método propuesto en la sección anterior en distintos casos. En la figura (2.2) se presenta el ejemplo del ajuste de un poliéster-poliuretano con 4 escalones distintos, así que el modelo en este caso está formado por la suma de cuatro funciones logísticas.

En el caso del ajuste de la degradación de las maderas industriales, se puede ver cómo las distintas funciones logísticas se corresponden con las diferentes componentes de la madera (ver figura 2.3).

### 2.3.3. Ventajas del modelo de mezcla de logísticas

Podríamos enumerar las siguientes ventajas para el modelo propuesto:

1. Este método abarca toda la gama de conversiones. El modelo (2.22) puede ser usado para el estudio de cualquier tipo de material, tanto si está regido por una sola ley cinética como si lo está por varias.
2. Permite representar tanto la velocidad de pérdida de masa como el ajuste de ésta con el tiempo mediante una ecuación única para cada proceso.
3. No se basa en una ley preestablecida tipo Arrhenius, lo que permite compararlo con otros modelos al uso.
4. Proporciona medios estadísticos que permitirán valorar la bondad del ajuste del modelo propuesto mediante contrastes de significación.
5. Permite la aplicación de modelos cinéticos clásicos del tipo Arrhenius a cada una de las funciones de degradación individuales, para su comparación con materiales ya estudiados.

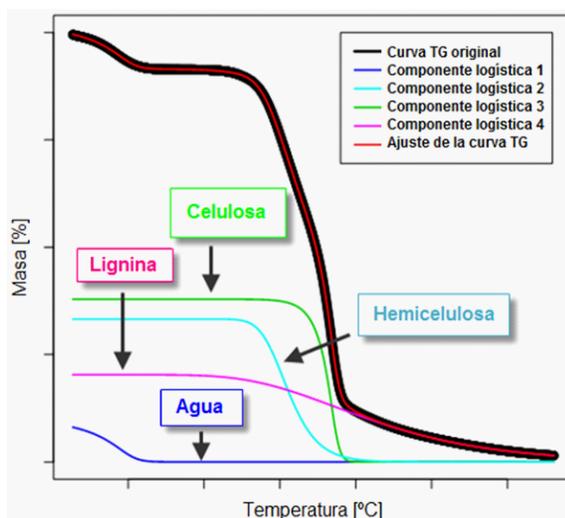


Figura 2.3: Modelización de una madera industrial mediante la suma de 4 funciones logísticas.

6. Reproduce perfectamente el carácter asintótico al comienzo y al final de cada proceso degradativo.

Resumiendo, el método de mezcla de logísticas o su extensión a las logísticas generalizadas, permite el ajuste de las curvas TG, basado en la composición de tantas funciones logísticas como procesos distintos sufra la muestra, supone un modelo alternativo a los existentes para la explicación de la cinética. Este método nos parece más simple, en cuanto a la fácil interpretación, que los anteriores basados en la ley de Arrhenius.

Entre las aplicaciones más destacadas de este modelo se podrían citar la aplicación al hexahidroftálico (Naya, Cao y Artiaga, 2003a). El modelo con mezcla de logísticas, con aplicaciones a distintos casos, se ha publicado en la revista *Polymer Degradation and Stability* (Cao et al. 2004) y una comparativa con los métodos anteriores fue publicada en la revista *Journal of Chemometrics* (Naya et al. 2008). Además, este método paramétrico se utilizó para separar componentes en distintos tipos de materiales (Artiaga et al. 2005), también se aplicó con éxito en la determinación de la degradación de las resinas epoxi con nanoclays (Naya, Martínez-Vilariño, Artiaga, 2009) o a la determinación de la degradación de la madera, ajustando una función logística a cada una de las fases de los distintos componentes de la madera (agua, lignina, celulosa y hemicelulosa) lo que permite por una parte analizar la descomposición térmica de este material (Tarrío et al. 2011) y por otra, su clasificación en distintas especies (Sebio-Puñal et al. 2012).

En el capítulo IV de esta tesis se estudiará cómo modelizar la degradación de nautubos de carbono mediante la aplicación de este tipo de modelos (Ríos-Fachal

et al. 2013) usando algoritmos evolutivos para la estimación de los parámetros que se presentarán en el siguiente capítulo.



# Capítulo 3

## Optimización

### 3.1. Introducción

Una de los aspectos con mayor dificultad al usar las Matemáticas para la modelización es encontrar los parámetros óptimos del modelo propuesto, para lo que se precisa emplear los métodos de optimización adecuados en cada caso.

Los métodos de optimización permiten la mejora de las operaciones y decisiones, mediante técnicas científicas adecuadas, seleccionadas de la matemática. Se basan en recoger y analizar datos, desarrollar y probar modelos matemáticos, proponer soluciones o recomendaciones, interpretar la información y, en definitiva, ayudar a implantar acciones de mejora. Como resultado desarrollan e implementan programas informáticos que permiten la ejecución rápida de los algoritmos de optimización.

Estos problemas tienen sus orígenes en la II Guerra Mundial, debido a la necesidad urgente de asignación de los escasos recursos en las operaciones militares y en problemas tácticos y estratégicos. Estas mismas técnicas se han extendido con posterioridad a las empresas.

La optimización puede ser: lineal, no lineal, entera, estocástica, multiobjetivo y además también involucra otras disciplinas como los algoritmos metaheurísticos, la lógica borrosa, las redes neuronales, el reconocimiento de patrones, etc. En otros contextos también se estudian estos mismos problemas en disciplinas como la inteligencia artificial.

La optimización consiste en la selección de una alternativa mejor, en algún sentido, que el resto, entre las alternativas posibles. Los problemas de optimización generalmente tienen tres componentes:

- **Función objetivo:** es la medida cuantitativa del funcionamiento del sistema que se desea optimizar (maximizar o minimizar). Por ejemplo, son funciones objetivo: la minimización de los costes variables de operación de un sistema eléctrico, la maximización de los beneficios netos de venta de ciertos productos, la minimización del cuadrado de las desviaciones con

respecto a unos valores observados o la minimización del material utilizado en la fabricación de un producto.

- **Variables:** representan las decisiones que se pueden tomar para afectar el valor de la función objetivo. Desde un punto de vista funcional se pueden clasificar en variables independientes (principales o de control) y variables dependientes (auxiliares o de estado), aunque matemáticamente todas son iguales. En el caso de un sistema eléctrico serán los valores de producción de los grupos de generación o los flujos por las líneas; en el caso de la venta, la cantidad de cada producto fabricado y vendido; en la fabricación de un producto, sus dimensiones físicas,...
- **Restricciones:** representan el conjunto de relaciones (expresado mediante ecuaciones e inecuaciones) que ciertas variables están obligadas a satisfacer. Por ejemplo, las potencias máxima y mínima de un grupo de generación, la capacidad de producción de una fábrica para los diferentes productos o las dimensiones del material bruto del producto.

Resolver un problema de optimización consiste en encontrar los valores que deben tomar las variables para hacer óptima la función objetivo satisfaciendo el conjunto de restricciones. Normalmente, en el caso de buscar un valor mínimo, la idea sería encontrar un valor  $x^*$  de forma que para cualquier otro valor  $x \in R$ ,  $f(x^*) \leq f(x)$ , donde  $f$  es la función de dominio en el espacio de los valores que modelizan el problema.

Además, los métodos de optimización también pueden clasificarse en: métodos clásicos (algoritmos habitualmente descritos en los libros de optimización) y métodos metaheurísticos (surgidos ligados a la inteligencia artificial y que imitan fenómenos sencillos observados en la naturaleza). Entre los primeros están la optimización lineal, lineal entera mixta, estocástica, dinámica, etc.; y en el segundo grupo se incluyen los algoritmos evolutivos (genéticos entre otros), el método de recocido simulado (simulated annealing), las búsquedas heurísticas o los sistemas multiagente. De forma muy general y aproximada, puede considerarse que los métodos clásicos buscan y garantizan un óptimo local, mientras que los metaheurísticos tienen mecanismos específicos para alcanzar un óptimo global, aunque no garantizan su alcance.

Los problemas se distinguen por el carácter de las funciones  $f$  que intervienen (lineales o no lineales) y el de las variables (reales, continuas, discretas o enteras). También suelen diferenciarse en métodos de tipo heurístico (ad hoc) o metaheurísticos. Son éstos últimos los de mayor interés en esta memoria y entre ellos se distinguen: los métodos de búsqueda de vecinos más próximos (Variable Neighborhood Search o VNS), de búsqueda tabú, de enjambre de partículas, de recocido simulado, basados en algoritmos genéticos, etc. A éstos últimos se le dedicarán las últimas secciones de este capítulo por sus interesantes prestaciones para los problemas a resolver en esta memoria.

La palabra *heurística* tiene su origen en la antigua palabra griega *heuriskein*, el arte del descubrimiento de nuevas estrategias (reglas) para resolver problemas. El sufijo *meta*, también griego, significa “metodología de alto nivel”. El término

*metaheurística* fue introducido por Glover en su artículo sobre optimización del año 1986 (Glover, 1986).

Estos métodos de búsqueda metaheurísticos pueden definirse como metodologías (plantillas) generales de alto nivel que pueden ser utilizadas como estrategias guía en el diseño de heurísticas subyacentes que resuelvan problemas de optimización específicos. Al contrario que con los métodos exactos, las metaheurísticas permiten abordar instancias de problemas complicados obteniéndose soluciones satisfactorias en un tiempo razonable. Estas técnicas metaheurísticas han tenido mucha popularidad en las últimas décadas dada la evolución de los ordenadores que las hacen rápidas y eficaces en la resolución de problemas complejos.

La resolución de problemas complejos mediante algoritmos y técnicas evolutivas es uno de los aspectos más investigados. El objetivo principal que se ha planteado en esta tesis es desarrollar nuevos métodos o algoritmos capaces de resolver estos problemas con el menor tiempo computacional posible, mejorando la calidad de los resultados obtenidos por los métodos ya existentes en el contexto del análisis térmico.

## 3.2. Métodos clásicos de optimización

### 3.2.1. Algoritmo de Brute Force

Este algoritmo, que podría traducirse como algoritmo de fuerza bruta, búsqueda combinatoria o búsqueda exhaustiva, es una técnica trivial, que consiste en enumerar sistemáticamente todos los posibles candidatos para la solución de un problema con el fin de contrastar si dicho candidato satisface la solución del mismo. Es sencillo de implementar y, siempre que exista, encuentra la solución. Su coste de ejecución es proporcional al número de soluciones candidatas, y éste es exponencialmente proporcional al tamaño del problema.

Este método se usa si el número de soluciones candidatas no es muy elevado o si éste puede reducirse previamente usando algún otro método heurístico y también si es más importante una implementación sencilla que una mayor rapidez.

### 3.2.2. Algoritmo de Newton-Raphson o de Newton-Fourier

El algoritmo de Newton-Raphson o de Newton-Fourier es uno de las más empleados por ser eficiente para encontrar aproximaciones de los ceros o raíces de un polinomio y puede ser utilizado para encontrar máximos o mínimos de una función encontrando los ceros de su derivada.

Es un método abierto en el sentido de que no garantiza la convergencia global y la única forma de asegurar la convergencia es seleccionar un valor inicial lo suficientemente cercano a la raíz:

$$x_{n+1} = x_n - f(x_n)/f'(x_n)$$

Se sustituye la función por la recta tangente, se iguala a 0 y se despeja: este 0 será una aproximación mejor de la raíz de la función.

### 3.2.3. Algoritmo de Gauss-Newton

El algoritmo de Gauss-Newton se utiliza para resolver problemas no lineales de mínimos cuadrados, es iterativo y se necesita proporcionar una estimación inicial del parámetro:

$$p_{k+1} = p_k - (Jf(p_k)^t \cdot Jf(p_k))^{-1} \cdot Jf(p_k)^t \cdot f(p_k)$$

Es una modificación del método de optimización de Newton que no usa segundas derivadas y se debe a Gauss, donde  $f = (f_1, \dots, f_m)$  y  $Jf(p)$  denota el Jacobiano de  $f$  en  $p$ .

El problema que presentan la mayoría de estos métodos clásicos de optimización es que requieren ciertas propiedades de la función objetivo a minimizar, que en el caso de funciones de mezcla de logísticas, no siempre se verifican, por lo que su convergencia al óptimo no siempre está garantizada. Por este motivo se propone la utilización de algoritmos evolutivos que no precisan de las mismas hipótesis, y que se presentarán en las siguientes secciones.

## 3.3. Optimización en análisis cinético

A partir de 1985 se extiende el uso de los denominados métodos numéricos de análisis cinético, auspiciados sobre todo por la aparición de programas informáticos acoplados a calorímetros. Estos métodos emplean algoritmos numéricos como los de Euler o el de Runge-Kutta (Turi (2000)).

La idea de estas aproximaciones numéricas parte de una expresión para  $f(m)$ , se integran las ecuaciones para unos determinados valores de los parámetros cinéticos, se compara la curva calculada con la experimental mediante una función objetivo y se calculan aquellos valores que optimizan dicha función objetivo mediante la utilización de métodos como el *simplex flexible*.

El criterio más utilizado es el método de mínimos cuadrados, cuyas expresiones más habituales son:

$$\text{mín} \sum_i \sum_j (m_{cal(i,j)} - m_{exp(i,j)})^2 \quad (3.1)$$

$$\text{mín} \sum_i f_i \sum_j \left( \frac{dm_{cal(i,j)}}{dt} - \frac{m_{exp(i,j)}}{dt} \right)^2. \quad (3.2)$$

Al primer método (3.1) se le denomina integral y al segundo método (3.2) diferencial. En ambas expresiones  $m_{cal(i,j)}$  y  $m_{exp(i,j)}$  representan respectivamente, los valores calculados con los diferentes modelos cinéticos y los valores experimentales de la masa de la muestra. El factor  $f_i$  es un parámetro de escala,

incluido para asegurar que todos los experimentos tienen el mismo peso en la función objetivo.

Para obtener el mínimo de las expresiones anteriores (3.1) y (3.2) suele emplearse el método del simplex flexible. Se trata de una técnica de optimización para funciones no lineales desarrollada por Nelder y Mead (Nelder y Mead, 1965), que no necesita utilizar las derivadas para el cálculo del mínimo, sino que evalúa la función en cada uno de los  $n + 1$  vértices de un hiperpoliedro, y va eliminando aquellos vértices que dan valores más altos; la iteración de este proceso permite obtener el mínimo buscado.

El método Nelder-Mead utiliza el concepto de un simplex, que es un politopo de  $N + 1$  vértices en  $N$  dimensiones: un segmento de línea en una línea, un triángulo en un plano, un tetraedro en un espacio tridimensional y así sucesivamente. Este método busca de modo aproximado una solución óptima local a un problema con  $N$  variables cuando la función a minimizar varía suavemente.

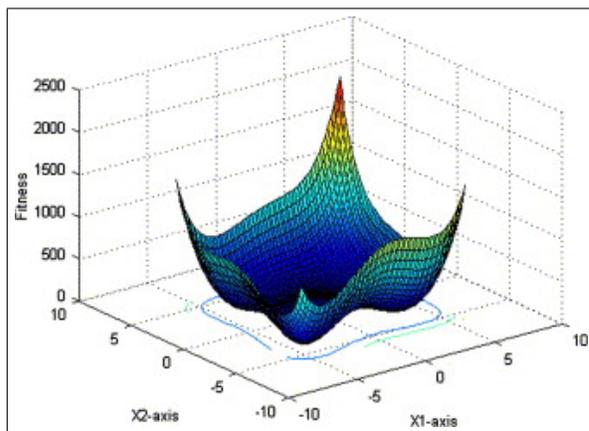


Figura 3.1: Superficie con varios mínimos locales.

Como otros algoritmos de optimización, Nelder-Mead a veces se queda atrapado en un mínimo local (una zona que es un mínimo de la función comparado con los puntos de alrededor pero hay motivos para pensar que existe un mínimo mejor en otra parte). El algoritmo se da cuenta y se reinicia con un nuevo simplex que empiece en el mejor valor encontrado. Esto puede resolverse con propuestas de algoritmos como el simulated annealing para tratar de escapar de los mínimos locales y que será tratado con detalle en esta memoria.

Existen también propuestas que no se ajustan al modelo inicial propuesto (2.7). En este sentido cabe mencionar el *método cinético de tipo no paramétrico* (NPK) propuesto por Nomen y Sampere (1998), que se basa en suponer que la reacción se puede expresar mediante el producto de dos funciones independientes,  $f(\alpha)$  y  $f'(T)$ ; la función  $f(\alpha)$  representará la cinética y  $f'(T)$  la dependencia de la temperatura. El modelo se expresaría mediante una matriz  $A$ , de rango  $n \times m$  con  $A_{i,j} = f(\alpha_i)f'(T_j)$ , según la expresión  $A = ab^t$ , siendo

los vectores  $a = \{f(\alpha_1), f(\alpha_2), \dots, f(\alpha_n)\}^t$  y  $b = \{f'(T_1), f'(T_2), \dots, f'(T_m)\}^t$ . Este método tiene dos inconvenientes: por un lado necesita un número muy elevado de cálculos para la estimación de los parámetros mediante algoritmos de aproximación y, por otra parte, exige realizar varios experimentos a distintas temperaturas  $T_1, T_2, \dots, T_n$  para la determinación del modelo.

### 3.3.1. Inconvenientes de los métodos clásicos

Los métodos mencionados anteriormente tienen como gran inconveniente proporcionar resultados diferentes para los parámetros de un mismo modelo. Este hecho fue puesto de manifiesto por varios científicos, como Vyazovkin (1996), como el mayor inconveniente en la aplicación de métodos tipo Arrhenius.

El problema se ha evidenciado en diversos artículos, pero fue sin duda en el congreso internacional de la ICTAC (Confederación Internacional de Análisis Térmico y Calorimetría) celebrado en Filadelfia, USA, en agosto de 1996, donde se ha dado a conocer a la comunidad científica. En este foro, se puso en evidencia la diferencia en el ajuste de modelos cinéticos a una misma muestra dependiendo del método usado, al proponer la aplicación a un experimento con carbonato de calcio (CC) a diferentes participantes invitados, y los resultados obtenidos se recogen en una serie de artículos de Brown et al. (2000) en la revista *Thermochimica Acta*. En ellos se constatan grandes diferencias entre los valores obtenidos, tanto para la energía de activación ( $E_a$ ) como para la constante preexponencial  $A$ . Cabe resaltar que en estos trabajos participaron también los investigadores Nomen y Sempere, quienes aplicaron a los datos su método no paramétrico, resultando estimaciones con grandes diferencias con respecto a métodos más clásicos tipo Freedman, Ozawa, Kissinger y otros.

Las diferencias entre los parámetros encontrados por cada autor para un mismo experimento corroboran la poca eficacia de los modelos existentes para este tipo de ajustes, lo que nos llevó a la búsqueda de algoritmos alternativos que serán expuestos en la siguiente sección.

## 3.4. Algoritmos Evolutivos

### 3.4.1. Breve reseña histórica

Darwin concibe por primera vez la idea de evolución por selección natural leyendo el argumento de Thomas Malthus (“Ensayo del principio de la población”) sobre la eficacia de la ayuda al pobre: el crecimiento de la población siempre será mayor que el de la comida suministrada. Esta discrepancia, notó Darwin, entre población y recursos disponibles conducirá a la competición y adaptación.

A principios de los años 60 un grupo de biólogos comenzó a usar programas de ordenador (muy elementales) que imitaban el comportamiento de la Naturaleza. No tuvieron demasiado éxito porque consideraban como operador generador de variación en las especies la mutación. Fue a John Holland (científi-

co, programador y psicólogo) al que –leyendo la “Teoría genética de la selección natural” de R.A. Fisher– se le ocurre la idea de que los algoritmos de ordenador basados en principios de genética y evolución podrían ser usados para desarrollar soluciones de problemas: “los organismos vivos son extraordinarios resolutores de problemas”, dijo. Holland se dio cuenta de que es el cruce, y no la mutación, el operador fundamental en el diseño y evolución de un algoritmo evolutivo, en este caso un algoritmo de tipo genético. Se preguntaba cómo logra la naturaleza crear seres cada vez más adaptados al medio en que se desarrollan. Aunque no conocía la respuesta, tenía una cierta idea de cómo hallarla: observó que esa adaptación se lleva a cabo a base de interacciones locales entre individuos, y entre éstos y lo que les rodea; así que trató de hacer pequeños modelos de la naturaleza, que tuvieran alguna de sus características, y ver cómo funcionaban, para luego extrapolar sus conclusiones a la totalidad. Este es el origen de los Algoritmos Genéticos (en adelante AG).

Este hecho ilustra cómo la combinación de ideas desde diferentes disciplinas, ayuda a resolver problemas, siendo ésta también una propiedad de los AG que contribuye a hacerlos muy poderosos: la combinación de información desde fuentes dispares. John Holland es considerado el padre de los AG, sin embargo la idea del uso de conceptos de biología en programación de ordenadores ya había sido planteada por Alan Turing en 1948 en su obra “Máquinas inteligentes”.

A principios de la década de 1960, en la universidad de Michigan, dentro del grupo Logic of Computers, las ideas de Holland comenzaron a desarrollarse a raíz de que impartiese un curso titulado “Teoría de sistemas adaptativos”. Dentro de este curso, y con una participación activa por parte de sus estudiantes, fue donde sus ideas comenzaron a formalizar lo que serían los algoritmos genéticos.

Sus objetivos eran dos:

- imitar los procesos adaptativos de los sistemas naturales, y
- diseñar sistemas artificiales (normalmente programas) que imitasen los mecanismos importantes de los sistemas naturales.

Unos 15 años más tarde, David Goldberg conoció a Holland y comenzó a trabajar con él. Goldberg era un ingeniero industrial que trabajaba en diseño de tuberías, y fue uno de los primeros que trató de aplicar los algoritmos genéticos a problemas industriales. Aunque Holland trató de disuadirle, porque pensaba que el problema era excesivamente complicado como para aplicarle algoritmos genéticos, Goldberg consiguió su propósito, escribiendo un algoritmo genético en un ordenador personal Apple II. Éstas y otras aplicaciones creadas por estudiantes de Holland convirtieron los algoritmos genéticos en un campo con base suficientemente aceptada para impulsar la primera conferencia internacional sobre Algoritmos Genéticos en 1985 (ICGA '85), que sigue celebrándose bianualmente.

Una generalización de los algoritmos genéticos son los algoritmos evolutivos. Estos algoritmos son métodos de optimización y búsqueda de soluciones basados en los postulados de la evolución biológica, aunque no sólo se basan en la formulación binaria como los genéticos. En ellos se mantiene un conjunto de

entidades que representan posibles soluciones, las cuales se mezclan, y compiten entre sí, de tal manera que las más aptas son capaces de prevalecer a lo largo del tiempo, evolucionando cada vez hacia mejores soluciones.

Los algoritmos evolutivos, y la computación evolutiva, son una rama de la inteligencia artificial. Son utilizados principalmente en problemas con espacios de búsqueda extensos y no lineales, en donde otros métodos no son capaces de encontrar soluciones en un tiempo razonable. Este será el principal motivo de proponerlos para la modelización de los datos de degradación.

### 3.4.2. Principios de un Algoritmo Evolutivo

Los biólogos han estudiado en profundidad los mecanismos de la evolución, y aunque quedan parcelas por entender, muchos aspectos están bastante explicados. De manera muy general, podemos decir que en la evolución de los seres vivos el problema al que cada individuo se enfrenta día a día es la supervivencia. Para ello cuenta con las habilidades innatas implantadas en su material genético. A nivel de los genes, el problema es el de buscar aquellas adaptaciones beneficiosas en un medio hostil y cambiante. Debido en parte a la selección natural, cada especie gana en cada generación una cierta cantidad de conocimiento, que es incorporado a la información de sus cromosomas.

Así pues, la evolución tiene lugar en los cromosomas, en donde está codificada la información del ser vivo. La información almacenada en el cromosoma varía de unas generaciones a otras y en el proceso de formación de un nuevo individuo, se combina la información cromosómica de los progenitores, aunque la forma exacta en la que se realiza no es del todo conocida todavía.

Aunque muchos aspectos están todavía por discernir, existen unos principios generales ampliamente aceptados por la comunidad científica. Algunos de éstos son:

1. La evolución opera en los cromosomas en lugar de operar en los individuos que los representan.
2. La selección natural es el proceso por el que los cromosomas con buenas estructuras se reproducen más a menudo que los demás.
3. En el proceso de reproducción tiene lugar la evolución mediante la combinación de los cromosomas de los progenitores. Llamamos *recombinación* a este proceso en el que se forma el cromosoma del descendiente. También son a tener en cuenta las *mutaciones* que pueden alterar dichos códigos.
4. La evolución biológica no tiene memoria, en el sentido de que en la formación de los cromosomas únicamente se considera la información del período anterior.

Los AG son efectivos en problemas de búsqueda de soluciones complejas, de gran dimensión, no lineales o espacios multidimensionales, y junto con la programación evolutiva, las estrategias de evolución y la programación genética, forman las técnicas llamadas *algoritmos evolutivos*, que convierten a los ordenadores en un tipo de máquinas inteligentes. De hecho, las “máquinas que aprenden” son una de las más prometedoras aplicaciones de los Algoritmos Evolutivos.

El esquema general o pseudo-código de un algoritmo evolutivo es el siguiente:

```

Inicio
   $t \leftarrow 0$ 
  inicializar  $P(t)$ 
  evaluar  $P(t)$ 
  mientras (no se dé la condición) hacer
     $t \leftarrow t + 1$ 
    seleccionar  $P(t)$  de  $P(t + 1)$ 
    recombinar  $P(t)$ 
    evaluar  $P(t)$ 
  fin
fin

```

Para llevar a la práctica el esquema anterior y concretarlo en un algoritmo, hay que especificar los siguientes elementos:

- Una representación cromosómica.
- Una población inicial.
- Una medida de evaluación.
- Un criterio de selección/eliminación de cromosomas.
- Una o varias operaciones de recombinación.
- Una o varias operaciones de mutación.
- Un test de parada.

A continuación se detallan estos elementos.

Un algoritmo es la descripción general de un procedimiento, y un programa es la ejecución de un algoritmo como secuencia de instrucciones dadas al ordenador. El primer paso en programación de AG es elegir la población inicial (aleatoria o heurísticamente), que estará formada por candidatos a solución del problema en cuestión. En esta población cada individuo será caracterizado por una cadena binaria de bits (cada bit corresponde a la presencia -1- o ausencia -0- de una determinada característica) de longitud fija que se llama *cromosoma*. En el segundo paso todos los cromosomas son evaluados por una *función de idoneidad*, y se crea una nueva población de cromosomas, por combinación de los individuos más aptos, es decir, con valores de idoneidad más altos. Esta combinación se hace a través de *operadores evolutivos*: selección, cruce y mutación. En algunas ocasiones podría suceder que el mejor individuo no produzca hijos, así que para evitarlo se introduce el concepto de *elitismo* (el mejor individuo de la población vieja se compara con todos los de la nueva, y, si es mejor, se transfiere directamente a la nueva población). Este proceso se continúa hasta obtener un individuo que logra un nivel específico de idoneidad, esto es, que resuelve el problema con suficiente precisión.

Los algoritmos evolutivos poseen varias ventajas computacionales:

1. Usan operadores estocásticos y no reglas deterministas.
2. Son conceptualmente simples, libres de condiciones sobre la función; por ejemplo, no precisan condiciones sobre las derivadas.
3. Pueden aplicarse a problemas de tiempo continuo y discreto.
4. Destacan por sus resultados en problemas difíciles.
5. Puede combinarse su uso con métodos tradicionales pudiendo ser extremadamente efectivos, por ejemplo para una conjetura inicial.
6. Permiten pasos ilógicos (similares a los de la mente humana).

Entre sus desventajas destacan su intensidad numérica y su elevado coste de tiempo computacional, aunque con los modernos procesadores, hoy esto resulta un problema menor. En ocasiones los límites son difíciles de implementar, y, aunque por su naturaleza resultan más adecuados para ser ejecutados en procesadores en paralelo, que imitan la evolución natural, no es seguro que una condición satisfecha por los padres la satisfagan también los hijos.

En los trabajos tradicionales es habitual caracterizar a los individuos mediante el uso de cadenas binarias de longitud fija, pero recientemente se han desarrollado representaciones más flexibles, y además se aborda la selección probabilísticamente, lo que conduce a una mayor probabilidad de desplazamiento dirigida a alcanzar el óptimo global.

La evolución como proceso no siempre conduce a la optimización, sólo a adaptación. Otra deficiencia intrínseca es que no hay un mejor algoritmo evolutivo, ni un mejor conjunto de parámetros y operadores para todos los problemas (“no free lunch theorem”).

A pesar de estas limitaciones existen muchos ejemplos en los que los AG son usados con éxito:

- Diseño de cuchilla de turbinas para motores de propulsión a chorro, problema que implica el uso de más de 100 variables: en un par de días con un diseño estándar con AG se incrementó la eficiencia 3 veces, lo que a un experimentado ingeniero le hubiese costado dos meses de cálculos.
- Planificación y uso de recursos: no sólo problemas clásicos como el del viajante (resuelto para miles de ciudades empleando técnicas combinadas con redes neuronales artificiales), sino también de transporte de mercancías, programación de entrega para mensajeros internacionales, trazado físico de plantas de manufactura o asignación de tareas y máquinas. En general, se trata de problemas en los que se ha de tener en cuenta productividad, evaluación de material, costes, tiempos, beneficios, satisfacción del cliente, recursos evaluables, etc., todo ello en entornos muy competitivos.
- Control de mecanismos seleccionando valores óptimos, afinando parámetros de redes de neuronas artificiales, redes de conectividad, número de capas ocultas, etc., con intención de perfeccionar operaciones al máximo nivel.
- En medicina han sido aplicados en el diseño de instrumentos para la clasificación automática de células (por ejemplo, en tests de manchas con tejidos cancerígenos) para el desarrollo de planes de tratamiento por radiología.

- En programación genética, como una técnica evolucionista que automáticamente crea programas de ordenador desde lenguajes de alto nivel. El objetivo final sería conseguir un ordenador capaz de resolver un problema sin programarlo explícitamente.

Como queda dicho, en general, los algoritmos evolutivos operan en una población de candidatos a solución (que pueden ser programas de ordenador) usando el principio de supervivencia del más apto (búsqueda del programa que mejor desempeña una especificación). A veces estos programas no conducen a una solución comprensible para un humano, lo que hace que su uso en control de sistemas de seguridad crítica se considere, en ocasiones, imprudente, pero en otra multitud de aplicaciones es obvia su utilidad.

La filosofía de actuación de los AG se puede resumir diciendo que establecen una analogía entre el conjunto de soluciones de un problema y el conjunto de individuos de una población natural. En palabras del propio Holland: se pueden encontrar soluciones aproximadas a problemas de gran complejidad computacional mediante un proceso de *evolución simulada*.

A tal efecto se introduce una función de evaluación de los cromosomas, que llamaremos función de aptitud o idoneidad (*fitness*) y que, en el caso particular de la optimización funcional, es la función objetivo del problema. Igualmente, se introduce un mecanismo de *selección*, de manera que los cromosomas con mejor evaluación sean escogidos para *reproducirse* más a menudo que los que la tienen peor.

Los algoritmos desarrollados por Holland inicialmente eran sencillos pero dieron buenos resultados en problemas considerados difíciles. Esto no debe sorprendernos ya que los AG se basan en integrar e implementar eficientemente dos ideas fundamentales: las representaciones simples como vectores binarios de las soluciones del problema y la realización de transformaciones simples para modificar y mejorar estas representaciones.

En los primeros trabajos originales las soluciones se representaban por cadenas binarias, es decir, listas de 1s y 0s. Este tipo de representaciones es ampliamente utilizado, incluso en problemas en donde no es muy natural. En 1985, De Jong introduce la siguiente cuestión: ¿qué se debe hacer cuando los elementos del espacio de búsqueda se representan de modo natural por estructuras complejas como vectores, árboles o grafos?, ¿se debe intentar linealizar en una cadena o trabajar directamente con estas estructuras?

En la actualidad podemos distinguir dos escuelas:

- . Limitarse a cadenas binarias como el caso de los algoritmos genéticos
- o
- . Utilizar otro tipo de configuraciones

Hay que destacar que las operaciones genéticas dependen del tipo de representación, por lo que la elección de una condiciona a la otra.

### 3.4.3. Desarrollo de un algoritmo en codificación binaria

Las ventajas de la codificación binaria son que permite definir fácilmente operaciones de recombinación y los resultados sobre convergencia están probados para el caso de cadenas binarias. Sin embargo, en algunos problemas puede ser poco natural y eficiente el utilizarlas. Por ejemplo, en el problema del viajante sobre 5 ciudades (A, B, C, D y E) y una codificación de 4 bits por cada ciudad, A = 0001, B = 0010, C = 0100, D = 1000, E = 1010 la cadena siguiente 0100/1010/1000/0001/0010 representa la ruta que se inicia en C, pasa por E, luego por D, a continuación por A y finaliza en B. Sin embargo, dicha representación no es muy natural y además, no todas las cadenas con 20 bits representan soluciones, por ejemplo 0100/1100/0010/0101/1010, lo cual complica substancialmente la definición de una operación de cruce. En este caso sería más natural la ruta de ciudades: (2,3,1,5,4) –donde cada número se asocia a una ciudad–, porque permite definir naturalmente diferentes operaciones estables.

El código Gray es otra forma de representación, que tiene la propiedad de que un incremento de una unidad en un parámetro corresponde al cambio de un bit en el código (números sucesivos sólo se diferencian en un dígito). La fórmula de conversión de código Gray a código binario es:

$$\gamma_k = \begin{cases} \beta_1 & \text{si } k = 1 \\ \beta_k \oplus \beta_{k-1} & \text{si } k > 1 \end{cases}$$

donde  $\gamma_k$  es el  $k$ -ésimo bit del código Gray, y  $\beta_k$  es el  $k$ -ésimo bit del código binario, numerando de izquierda a derecha, y  $\oplus$  denota la suma módulo 2. La conversión de código Gray a binario es:

$$\beta_k = \sum_{i=1}^k \gamma_i, \text{ con la suma módulo 2.}$$

Por ejemplo, el código binario 1101011 corresponde a 1011110 en código Gray.

La gran ventaja de este código es su facilidad de conversión al código binario natural y viceversa. En la siguiente tabla se reflejan los códigos Gray de 2, 3 y 4 bits.

Dígito decimal	2 bits	3bits	4bits
0	00	000	0000
1	01	001	0001
2	11	011	0011
3	10	010	0010
4		110	0110
5		111	0111
6		101	0101
7		100	0100
8			1100
9			1101
10			1111
11			1110
12			1010
13			1011
14			1001
15			1000

Una vez determinado el tipo de representación a utilizar, el siguiente paso es *inicializar una población de individuos*. La población inicial suele ser generada aleatoriamente. Sin embargo, últimamente se están utilizando métodos heurísticos para generar soluciones iniciales de buena calidad. En este caso, es importante garantizar la diversidad estructural de estas soluciones para tener una representación de la mayor parte posible de población o al menos evitar la convergencia prematura.

Como ya se ha dicho, la *selección* consiste en decidir cuáles de los individuos que constituyen una población serán los idóneos para generar la siguiente población, proceso para el que existen varias posibilidades. Este proceso de selección de los *padres* está determinado, habitualmente, por probabilidades según su idoneidad (*fitness*). En cuanto al método de selección, el más sencillo consiste en ordenar la población de acuerdo a su mérito (*idoneidad*), hacer que sobreviva la mitad de mejor calidad y generar la nueva mitad a partir de la anterior. Otra técnica posible es la de la *ruleta* que consiste en realizar una escala o normalización de la idoneidad, atendiendo a lo cual se le asigna una probabilidad de reproducción a cada cromosoma. Para realizar una selección se efectúa una tirada de bola en la ruleta, tomando el individuo asociado a la casilla en la que caiga la bola.

Otra posibilidad es la selección por *torneo* que consiste en seleccionar dos cromosomas de forma aleatoria y enfrentarlos de forma que sobreviva el de mejor calidad en la mayoría de los casos (por ejemplo en el 95 % de las luchas).

Los operadores de cruce más utilizados son:

- De un punto: se elige aleatoriamente un punto de ruptura en los padres y

se intercambian sus bits. Ejemplo:

$$\begin{aligned} 1100\mathbf{1011}+1101\mathbf{1111} &= 11001111 \text{ (hijo 1)} \\ 1100\mathbf{1011}+1101\mathbf{1111} &= 11011011 \text{ (hijo 2)} \end{aligned}$$

- De dos puntos: se eligen dos puntos de ruptura al azar para intercambiar los bits fuera de esos dos puntos. Ejemplo:

$$\begin{aligned} 1100\mathbf{1010} + 1101\mathbf{1111} &= 11011110 \text{ (hijo 1)} \\ 1100\mathbf{1010} + 1101\mathbf{1111} &= 11001011 \text{ (hijo 2)} \end{aligned}$$

- Uniforme: en cada bit se elige al azar un padre para que contribuya con su bit al del hijo, mientras que el segundo hijo recibe el bit del otro padre.
- PMX (Partially Matched Crossover) y SEX: son operadores más sofisticados, fruto de mezclar y aleatorizar los anteriores.

La operación de *mutación* más sencilla, y una de las más utilizadas, consiste en reemplazar con cierta probabilidad el valor de un bit. Por ejemplo:

$$0100\mathbf{1100} \rightarrow 0100\mathbf{0100}$$

Obsérvese que el papel que juega la mutación es el de introducir un factor de diversificación ya que, en ocasiones, la convergencia del procedimiento a buenas soluciones puede ser prematura y quedarse atrapado en óptimos locales. Otra forma obvia de introducir nuevos elementos en una población es recombinar elementos tomados al azar sin considerar su calidad.

Un tercer operador genético es la *inversión*. La inversión sencilla es un operador bit a bit (como la mutación): se seleccionan dos puntos dentro de una cadena y se invierte el orden de los bits entre dichos puntos, pero recordando el significado de los bits; lo que significa que tenemos que identificar bits en la cadena.

En la siguiente sección se profundiza sobre la actuación de los algoritmos genéticos.

#### 3.4.4. Convergencia del Algoritmo

Dado que el algoritmo genético opera con una población en cada iteración, se espera que el método converja, de modo que al final del proceso la población sea muy similar y, en el infinito, se reduzca a un sólo individuo.

Se ha desarrollado toda una teoría para estudiar la convergencia de estos algoritmos en el caso de cadenas binarias. Esta teoría se basa principalmente en considerar que una cadena es un representante de una clase de equivalencia o *esquema*, reinterpretando la búsqueda en lugar de entre cadenas, entre esquemas. De este modo se concluye lo que se conoce como *paralelismo intrínseco*: “En una población de  $m$  cadenas se están procesando implícitamente  $o(m^3)$  esquemas.”

A partir de este resultado el teorema de los esquemas prueba que la población converge a unos esquemas que cada vez son más parecidos, y en el límite a una única cadena.

Como consecuencia del *Teorema de los esquemas* se han probado varias generalizaciones, la mayoría para poblaciones discretas, pero también para espacios de búsqueda y funciones medibles ( $\Omega = \mathbb{R}^n$ ). En este último caso, la función de idoneidad será una función integrable y el operador de cruce es un operador estocástico en dos variables; de este modo, si el espacio de búsqueda tiene una métrica, se puede tener en cuenta la distancia entre dos puntos para obtener nuevos puntos.

En la práctica, en ocasiones, los algoritmos genéticos resuelven satisfactoriamente un problema de optimización dado y en otras se quedan muy alejados del óptimo. Los estudiosos del tema han tratado de caracterizar lo que han denominado problemas AG-fáciles (aquellos en los que los AG proporcionan buenos resultados) y AG-difíciles con el objetivo de saber de antemano, al estudiar un nuevo problema, si los AG son una buena elección para su resolución.

Se han tratado de caracterizar estas clases mediante el concepto de *engaño* considerando que si el algoritmo converge al mejor esquema (aquel con mejor promedio de aptitud de sus cadenas) y en éste se encuentra el óptimo, entonces es fácil que se resuelva satisfactoriamente. En caso de que el óptimo esté en un esquema con bajo promedio se habla del fenómeno de la *decepción*. Durante cierto tiempo se creyó que equivalían dificultad y decepción, pero resultados experimentales han demostrado que la ausencia (o presencia) de este fenómeno no garantiza la facilidad (o dificultad) de una función.

Es importante citar que, a diferencia de otros metaheurísticos, los Algoritmos Genéticos han crecido de forma espectacular, hasta el punto de poder encontrar referencias sobre ellos en revistas de informática de carácter general. Además, muchos de los investigadores de este campo están trabajando en desarrollar los aspectos teóricos de la materia incorporando algunas otras técnicas de búsqueda local en el esquema genético.

Especial mención merecen los Algoritmos Meméticos, híbridos o lamarkianos, que son una clase amplia de Metaheurísticos que intentan aunar ideas y conceptos de diferentes prácticas de resolución. Estos métodos incorporan técnicas variadas, desde “búsqueda local” hasta operadores de cruce con el fin de introducir información y conocimiento en la búsqueda. Uno de los métodos que podemos encontrar bajo esta denominación es *Scatter Search* que comparte también elementos de *Tabu Search* (técnica que restringe la selección de nuevos hijos y requiere memoria en la que guardar el conjunto histórico).

### 3.4.5. Descripción del método

En esta sección se expone cómo funciona un AG en el caso de un problema clásico de optimización.

Se supone, sin pérdida de generalidad, que se maximizan funciones; el caso de minimización estaría contemplado sin más que considerar la maximización de la función opuesta ( $\min(f) = \max(-f)$ ).

Además, se puede suponer que la función objetivo,  $f$ , toma valores positivos en su dominio pues en caso contrario basta hacer una traslación sumándole una constante.

Se tendrá, pues, el problema de maximizar una función  $f$ , real de variable real, de  $k$  variables,  $x_1, x_2, \dots, x_k$ , donde  $x_i \in [a_i, b_i]$ . Se desea la optimización de esta función con una cierta precisión fija, por ejemplo de seis cifras decimales para cada una de las variables  $x_i$ , por lo que cada uno de los dominios deberá dividirse en  $(b_i - a_i)/10^6$  partes iguales.

Llamando  $m_i$  al menor entero tal que  $(b_i - a_i)/10^6 \leq 2^{m_i} - 1$ , una representación binaria en la que cada variable  $x_i$  está codificada por una cadena de  $m_i$  bits de longitud, claramente satisface la precisión requerida.

Se denota por decimal (cadena) el valor decimal de la cadena de bits en base 2 de la variable, por lo que podremos expresar:

$$x_i = a_i + \text{decimal}(1001\dots 001) \frac{b_i - a_i}{2^{m_i} - 1}.$$

Ahora, cada cromosoma  $\nu$  (posible solución) está representado por una cadena de longitud:

$$\begin{aligned} m &= \sum_{i=1}^k m_i, \quad \nu = (c_1, c_2, \dots, c_k), \quad \text{long}(\nu) = m, \\ \nu &= (10^{m_1}01/0101^{m_2}01/\dots/111^{m_k}000). \end{aligned}$$

Los  $m_1$  primeros bits están en  $[a_1, b_1]$ , los siguientes  $m_2$  bits están en  $[a_2, b_2]$ ,... etc.

Para inicializar la población se elige un número  $N$ , tamaño de la población de cromosomas. Incluso, si se tiene algún conocimiento del posible valor óptimo se puede usar esta información. El resto del proceso del algoritmo es el siguiente: en cada generación se evalúa cada cromosoma (usando la función  $f$  sobre las secuencias decodificadas de las variables), se selecciona una nueva población respecto a la distribución de probabilidad basada en valores convenientes y se recombinan los cromosomas para formar una nueva población (por medio de operadores de cruce y mutación). Después de un número de generaciones, cuando ya no se observen grandes mejoras, el mejor cromosoma representa una solución óptima (posiblemente global).

A menudo se para el algoritmo después de un número fijo de iteraciones, dependiendo de criterios de rapidez y recursos.

El procedimiento de iteración es el siguiente:

*i)* Se calcula la función objetivo (idoneidad o aptitud) para cada cromosoma

$$\text{eval}(\nu_j) = \text{eval}(c_1^j, \dots, c_k^j) = f(x_1^j, \dots, x_k^j).$$

*ii)* Se halla la idoneidad global de la población, dada por

$$F = \sum_{i=1}^N \text{eval}(\nu_i).$$

*iii)* Se calcula la probabilidad  $p_i$  de seleccionar cada cromosoma

$$p_i = \frac{\text{eval}(\nu_i)}{F} = \frac{\text{eval}(\nu_i)}{\sum_{i=1}^N \text{eval}(\nu_i)}.$$

*iv)* Se calcula la probabilidad acumulada  $q_i$  de cada cromosoma

$$q_i = \sum_{l=1}^i p_l.$$

Para el proceso de selección se utiliza la ruleta tantas veces como el tamaño de la población (número total de cromosomas). Cada vez se selecciona un único cromosoma, para la nueva población, de la siguiente forma:

- 1) Se selecciona un número aleatorio  $r$  entre 0 y 1.
- 2) Si  $r < q_1 = p_1$  se selecciona  $\nu_1$ , en otro caso se selecciona el  $i$ -ésimo cromosoma  $\nu_i$  ( $2 \leq i \leq N$ ) tal que  $q_{i-1} < r \leq q_i$ . Obviamente, algunos cromosomas pueden resultar seleccionados más de una vez. (Los mejores cromosomas generan más copias, los peores desaparecen).
- 3) A continuación, se hace la primera recombinación, utilizando un operador de cruce, de los individuos de la nueva población.

Como se ha mencionado antes, uno de los parámetros de un sistema genético es la probabilidad de cruce  $p_c$ . Esta probabilidad proporciona el número esperado ( $p_c \times N$ ) de cromosomas que se verán afectados por la operación de cruce. Se procederá de la siguiente forma, para cada cromosoma de la nueva población:

- a) Se selecciona un número aleatorio  $r$  entre 0 y 1.
- b) Si  $r < p_c$  se selecciona el cromosoma para ser cruzado.

Ahora se emparejan los cromosomas seleccionados arbitrariamente: para cada par de cromosomas se genera un número entero aleatorio  $pos$  de  $[1, m - 1]$  (con  $m$  la longitud total de cromosomas), que indicará la posición del punto de cruce. Dos cromosomas seleccionados

$$\begin{aligned} &(b_1, b_2, \dots, b_{pos}, b_{pos+1}, \dots, b_m) \\ &(c_1, c_2, \dots, c_{pos}, c_{pos+1}, \dots, c_m) \end{aligned}$$

se reemplazan por sus descendientes

$$\begin{aligned} &(b_1, b_2, \dots, b_{pos}, c_{pos+1}, \dots, c_m) \\ &(c_1, c_2, \dots, c_{pos}, b_{pos+1}, \dots, b_m). \end{aligned}$$

El siguiente operador de recombinación es la mutación, que actúa bit a bit, para lo que se necesita otro parámetro del sistema genético  $p_m$ , que proporcionará el número esperado de bits mutables ( $p_m \times N$ ). Cada bit tiene la misma

probabilidad de ser afectado por la mutación. Se procede de la siguiente forma: para cada cromosoma actual (después de los cruces) y para cada bit de cada cromosoma:

- a) Se selecciona un número aleatorio  $r$  entre 0 y 1.
- b) Si  $r < p_c$  mutamos el bit.

La nueva población está lista para una nueva evaluación. Esta evaluación se usa para construir la distribución de probabilidad para el siguiente proceso de selección. El resto del proceso es una repetición de los pasos anteriores.

### 3.4.6. Fundamentos matemáticos de los AG. Teorema de los esquemas

Los algoritmos genéticos actúan, cualitativamente, procesando esquemas. Un esquema, que se construye introduciendo el símbolo asterisco (\*) en el alfabeto de los genes, representa todas las cadenas que coinciden en todas las posiciones que no son asteriscos, es decir, un esquema es un hiperplano en el espacio métrico  $(E, d)$  donde  $E$  es el espacio de búsqueda y  $d$  es la distancia Hamming entre dos cromosomas; esto es, el número de bits distintos en su representación binaria. Por ejemplo si  $E = \{000, 001, 010, 011, 100, 101, 110, 111\}$  entonces  $d(000, 111) = 3$ , mientras que  $d(101, 111) = 1$ .

Un esquema de orden  $r$  es un conjunto de  $2^r$  cromosomas y cada cadena de longitud  $m$  estará en  $2^m$  esquemas. Considerando cadenas de longitud  $m$ , hay en total  $3^m$  posibles esquemas, y una población de tamaño  $N$  puede ser representada por entre  $2^m$  y  $N2^m$  esquemas diferentes.

Hay dos parámetros importantes en un esquema: *orden* y *longitud de definición*. El orden es el número de 0 y de 1 fijos en el esquema,  $o(S)$ , y define la especificidad del esquema (a mayor orden mayor “especificidad”, menor número de cromosomas). La noción de orden se usa para calcular las probabilidades de supervivencia de un esquema frente a mutaciones.

La longitud de definición de un esquema  $S$ ,  $\delta(S)$ , es la distancia entre la primera y la última posiciones fijas. Define la “compacidad” de la información contenida en un esquema, usándose para el cálculo de las probabilidades de supervivencia de un esquema a los cruces.

Como se ha comentado ya, el proceso de evolución simulado consiste en 4 pasos repetidos consecutivamente: el primer paso simplemente se encarga de mover el reloj de la evolución una unidad más, y durante el último paso se evalúa la población actual. El fenómeno principal del proceso de evolución sucede en los 2 pasos intermedios del ciclo evolutivo: selección y recombinación. Discutiremos el efecto de cada uno de esos pasos en el número esperado de esquemas representados en la población.

Sea  $n(S, t)$  el número de cromosomas de la población en  $t$  que pertenecen al esquema  $S$ . Otra propiedad de un esquema es su idoneidad en la iteración  $t$  (su ajuste),  $eval(S, t)$ , que es el valor medio de  $S$  en la iteración  $t$ . Así, suponiendo que hay  $q$  cromosomas  $\{\nu_1, \nu_2, \dots, \nu_q\}$  en la población que pertenecen al esquema  $S$  en  $t$ , entonces:

$$eval(S, t) = \frac{\sum_{i=1}^q eval(v_i)}{n(S, t)}.$$

Durante el paso de la selección se genera una población intermedia: se crean tantas cromosomas como el tamaño de la población,  $N$ . Cada cadena se copia cero, una o más veces dependiendo de su idoneidad. Como se vio anteriormente, en una selección de un cromosoma, el cromosoma  $\nu_i$  tiene una probabilidad  $p_i = \frac{eval(\nu_i)}{F(t)}$  de ser seleccionado, donde  $F(t)$  es el ajuste de la población completa en el tiempo  $t$ ,  $F(t) = \sum_{i=1}^N eval(\nu_i)$ .

Si en la iteración  $t$ -ésima tenemos  $n(S, t)$  elementos en  $S$ , dado que:

- i) para una cadena “media” del esquema  $S$  su probabilidad de ser seleccionada es  $\frac{eval S(t)}{F(t)}$  (en una selección individual).
- ii) el número de cadenas del esquema  $S$  es  $n(S, t)$ .
- iii) el número de selecciones individuales que se realizan es igual a  $N$ , entonces está claro que:

$$n(S, t + 1) = n(S, t) \cdot N \cdot \frac{eval S(t)}{F(t)}.$$

Teniendo en cuenta que el valor medio de la población es

$$\bar{f} = \frac{F(t)}{N},$$

entonces:

$$n(S, t + 1) = n(S, t) \cdot \frac{eval S(t)}{\bar{f}}, \quad (\text{Ecuación del crecimiento reproductivo}).$$

En definitiva, el número esperado de cadenas en  $t + 1$  del esquema  $S$  es el producto de la probabilidad de escoger una cadena del esquema  $S$  por el número de veces que se seleccionan las cadenas, por lo que se podrá escribir

$$\begin{aligned} p(S) &= n(S, t) \cdot \frac{eval S(t)}{F(t)} \Rightarrow \\ n(S, t + 1) &= p(S) \cdot N = n(S, t) \cdot N \cdot \frac{eval S(t)}{F(t)}. \end{aligned}$$

En otras palabras, el número de cromosomas en la población crece en proporción a la idoneidad del esquema. Esto significa que en esquemas “por encima de la media” se incrementa el número de representantes en la siguiente generación, y en esquemas “por debajo de la media” decrece el número de ellos; en esquemas con ajuste medio igual a la media se mantiene el mismo nivel.

El efecto, a largo plazo, de la regla anterior es claro. Si suponemos que un esquema  $S$  está sobre la media en un  $\varepsilon\%$ :

$$\begin{aligned} \text{eval } S(t) &= \bar{f}(t) + \varepsilon \cdot \bar{f}(t) \Rightarrow \\ n(S, t) &= n(S, 0) (1 + \varepsilon)^t. \end{aligned}$$

Aunque la selección no introduce nuevos puntos (potenciales soluciones) para considerar en el espacio de búsqueda, copia algunos vectores para formar una población intermedia. El segundo paso, la recombinación, es el responsable de introducir nuevos individuos en la población, mediante los operadores genéticos cruce y mutación.

Parece claro que la longitud de definición de un esquema juega un papel importante en la probabilidad de supervivencia o destrucción. En general, el lugar de cruce se selecciona uniformemente entre  $m-1$  posibles. Esto implica que la probabilidad de supervivencia de un esquema  $S$  es:

$$P_{Superv}(S) = 1 - \frac{\delta(S)}{m-1}.$$

Es importante destacar que sólo algunos cromosomas se cruzan, y la probabilidad de cruce es  $P_C$ . Esto significa que la probabilidad de que un esquema sobreviva es, de hecho:

$$P_{Superv}(S) = 1 - P_C \frac{\delta(S)}{m-1}.$$

Nótese también que si el lugar de cruce se selecciona entre posiciones fijas en un esquema, hay todavía una posibilidad de que el esquema sobreviva, pero la probabilidad es muy pequeña. No obstante, debemos modificar la fórmula y escribir:

$$P_{Superv}(S) \geq 1 - P_C \frac{\delta(S)}{m-1}.$$

El efecto combinado de la selección y el cruce nos da una nueva forma de la ecuación de crecimiento reproductivo:

$$n(S, t+1) \geq n(S, t) \cdot N \cdot \frac{\text{eval } S(t)}{F(t)} \left[ 1 - P_C \frac{\delta(S)}{m-1} \right]$$

que nos da el número esperado de individuos siguiendo el esquema  $S$  en la próxima generación, en función del número actual de cadenas del esquema, de su ajuste relativo, y de su longitud de definición. Claramente, esquemas por encima de la media con longitud de definición pequeña podrían crecer en proporción exponencial.

El siguiente operador a considerar es la mutación. El operador mutación modifica aleatoriamente una única posición en un cromosoma con cierta probabilidad. Cada una de las posiciones fijas de un esquema debe permanecer inalterada para que el esquema sobreviva y sólo son importantes para efectos de mutación los bits fijos, es decir, tantos como el orden del esquema. Dado que la probabilidad de alteración de un bit es  $P_M$ , la probabilidad de que un bit

sobreviva es  $1 - P_M$ . Cada mutación es independiente de las demás, de manera que:

$$P_{Superv}(S) \geq (1 - P_M)^{o(S)} = 1 + \binom{o(S)}{1} (-P_M)^1 1^{o(S)-1} + \dots$$

Como  $P_M \ll 1$ , se puede aproximar la probabilidad por:

$$P_{Superv}(S) \approx 1 - o(S)P_M.$$

El efecto combinado de la selección, cruce y la mutación proporciona una nueva forma para la ecuación de crecimiento reproductivo:

$$n(S, t + 1) \geq n(S, t) \frac{\text{eval } S(t)}{\bar{f}} \left[ 1 - P_C \frac{\delta(S)}{m - 1} - o(S)P_M \right].$$

Como en las formas anteriores más simples, la última expresión nos da el número esperado de representantes del esquema  $S$  en la próxima generación en función del número actual, la idoneidad relativa del esquema, su orden y su longitud de definición. De nuevo es claro que esquemas por encima de la media, con longitud de definición corta y orden bajo todavía incrementan exponencialmente el número de representantes en generaciones sucesivas.

Resumiendo, la ecuación anterior muestra que la selección incrementa la proporción de cadenas por encima de la media en un esquema. El muestreo no introduce nuevos esquemas (no representados en el estado inicial  $t = 0$ ), por lo que se introduce el operador de cruce, que intercambia aleatoriamente la información. Además, el operador de mutación introduce mayor grado de variabilidad en la población. El efecto combinado de estos dos operadores sobre un esquema es importante si el esquema es de orden bajo y longitud pequeña. Esta es la conclusión del conocido:

*Teorema de los esquemas:* Los esquemas cortos, de bajo orden, y por encima de la media incrementan exponencialmente su número de representantes en las generaciones sucesivas de un algoritmo genético.

Una consecuencia inmediata de este teorema es que los AG exploran el espacio de búsqueda por esquemas cortos y de orden bajo, ignorando otros subconjuntos. Este tipo de esquemas se conocen como bloques constructivos (“building blocks”) y dan nombre a la

*Hipótesis de los building blocks:* un algoritmo genético busca el comportamiento óptimo por medio de la yuxtaposición de “building blocks”. Una forma de calcular la eficacia del algoritmo es mediante el cálculo de su *epistasis*, que está relacionado con estos resultados, tanto con el teorema de los esquemas como con el concepto de los building blocks. Por ejemplo, los esquemas cortos se corresponden con una epistasis baja, y el aumento del tamaño de los esquemas genera un aumento de la epistasis. Dado que el cálculo resulta prolijo, se expone en un anexo al final de esta memoria su desarrollo para algunos casos concretos.

### 3.5. Selección de Algoritmos Evolutivos para AT

En esta sección se expondrán con más detalle cada uno de los algoritmos de tipo evolutivo que se han seleccionado por sus prestaciones para aplicarlos a la optimización en Análisis Térmico que es uno de los objetivos básicos de esta tesis.

#### 3.5.1. Algoritmo Evolución Diferencial (Differential Evolution)

El algoritmo denominado Evolución Diferencial, o con su denominación inglesa: *Differential Evolution* (DE), es un método de optimización perteneciente a la categoría de computación evolutiva, aplicado en la resolución de problemas complejos. Al igual que otros algoritmos de esta categoría, el DE mantiene una población de soluciones candidatas, que se recombinan y mutan para producir nuevos individuos, los cuales serán elegidos de acuerdo al valor de su función de idoneidad. Lo que caracteriza al método DE es el uso de vectores de prueba, los cuales compiten con los individuos de la población actual a fin de sobrevivir.

El primer artículo sobre DE fue publicado por Kenneth Price y Rainer Storn en octubre de 1995 bajo el nombre de "Differential Evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces". Originalmente, el método estaba enfocado a la resolución del problema de ajuste del polinomio de Tchebychev utilizando una variante del método llamado Genetic Annealing, el cual había sido desarrollado por Price el año anterior, en 1996. Este nuevo método de DE fue presentado en el First International Contest on Evolutionary Optimization, que buscaba comparar el potencial de distintos métodos de optimización en computación evolutiva, finalizando DE en tercer lugar.

Este algoritmo asume que las variables del problema a optimizar están codificadas como un vector de números reales. La longitud de estos vectores ( $n$ ) es igual al número de variables del problema, y la población está compuesta de  $NP$  (número de padres) vectores. Se define un vector  $x_p^g$ , en el que  $p$  es el índice del individuo en la población ( $p = 1, \dots, NP$ ) y  $g$  es la generación correspondiente. Cada vector está a su vez compuesto por las variables del problema  $x_{p,m}^g$ , en que  $m$  es el índice de la variable en el individuo ( $m = 1, \dots, n$ ).

Una de las ventajas de este método es que no precisa de un vector de valores iniciales, sino que puede iniciarse mediante un intervalo de valores. Este algoritmo se encuentra implementado en el paquete de R, DEoptim y en dicho programa han participado Mullen et al. (2011) y Ardia et al. (2011a, 2011b).

Storn y Price (1997) emplean tres operadores: mutación (ecuación 3.3), recombinación (ecuación 3.4) y selección (ecuación 3.5):

$$v^i|_k = x^{r1}|_k + F(x^{r2}|_k - x^{r3}|_k) \quad (3.3)$$

$$u_{k,j}^i = \begin{cases} v_{k,j}^i & \text{si } rand(0,1) \leq Cr \\ x_{k,j}^i & \text{si } rand(0,1) > Cr \end{cases} \quad (3.4)$$

$$u^i|_{k+1} = \begin{cases} u^i|_k & \text{si } f(u^i|_k) \leq f(x^i|_k) \\ x^i|_k & \text{si } f(u^i|_k) > f(x^i|_k) \end{cases} \quad (3.5)$$

donde  $r_1 \neq r_2 \neq r_3 \neq i$ , y  $F$  se conoce como el factor de escalado, siendo  $Cr$  la probabilidad para llevar a cabo la recombinación. El pseudo-código para este algoritmo es el siguiente:

- 1 : Generar población inicial  $P|0$  con  $Np$  individuos.
- 2 : Evaluar  $P|0$ .
- 3 : MIENTRAS criterio de terminación no sea alcanzado
- 4 : Leer contador de generaciones  $G$ .
- 5 : HACER para cada  $x_i \in P|G$
- 6 : Generar un vector mutante (ecuación logística)
- 7 : Generar un vector hijo
- 8 : Efectuar una comparación por pares Padre-Hijo
- 9 : FIN-HACER
- 10:  $G = G + 1$ .
- 11: FIN-MIENTRAS

### 3.5.2. Algoritmo CMA-ES

El algoritmo CMA-ES (estrategia evolutiva mediante adaptación de la matriz de covarianza), cuyo nombre corresponde con las siglas inglesas que lo definen, Covariance Matrix Adaptation Evolution Strategy. Evolution strategies (ES), es un algoritmo de tipo estocástico, específico para la optimización numérica de problemas no lineales y pertenece también a la clase de algoritmos evolutivos. En este caso el nombre del algoritmo hace referencia a la estrategia usada para la evolución, ya que las nuevas soluciones candidatas se toman de acuerdo a una distribución normal multivariante que está representada por su matriz de covarianzas. La adaptación de la matriz de covarianzas es lo que permite aprender al modelo, de forma similar a la aproximación de la matriz inversa de Hesse en el método Quasi-Newton en la optimización clásica. En contraste con otros métodos evolutivos, en éste se hacen menos suposiciones sobre la naturaleza de la función objetivo subyacente, siendo aplicable a la mayoría de las situaciones y presentando resultados muy competitivos. Además, en contraste con la mayoría de los algoritmos evolutivos, el método CMA-ES es de los que se denominan del tipo “cuasi paramétrico libre”, en los que el número de muestras candidatos (tamaño de la población) es ajustable por el usuario con el fin de cambiar el

comportamiento de búsqueda característica. CMA-ES ha sido aplicado con éxito en cientos de problemas.

Este algoritmo está implementado en el Software R mediante la librería *cmaes* (Covariance Matrix Adapting Evolutionary Strategy), creada por Heike Trautmann, Olaf Mersmann y David Arnu (Hieke et al. 2012).

### 3.5.3. Algoritmo Simulated Annealing

El algoritmo Simulated Annealing (SA) (que podría traducirse por recocido simulado o enfriamiento simulado) es un algoritmo de búsqueda meta-heurística para problemas de optimización global; el objetivo general de este tipo de algoritmos es encontrar una buena aproximación al valor óptimo de una función en un espacio de búsqueda grande. A este valor óptimo se lo denomina "óptimo global".

El nombre e inspiración viene del proceso de recocido del acero, una técnica que consiste en calentar y luego enfriar lentamente el material para variar sus propiedades físicas. El calor provoca que los átomos aumenten su energía y que puedan así desplazarse de sus posiciones iniciales (un mínimo local de energía); el enfriamiento lento les da mayores probabilidades de recrystalizar en configuraciones con menor energía que la inicial (mínimo global).

El método fue descrito independientemente por Scott Kirkpatrick, C. Daniel Gelatt y Mario P. Vecchi en 1983 y por Vlado Černý en 1985 y es una adaptación del algoritmo Metropolis-Hastings, un método de Montecarlo utilizado para generar muestras de estados de un sistema termodinámico.

En cada iteración, este método evalúa algunos vecinos del estado actual  $s$  y probabilísticamente decide entre efectuar una transición a un nuevo estado  $s'$  o quedarse en el estado  $s$ . En el ejemplo de recocido de metales descrito arriba, el estado  $s$  se podría definir en función de la posición de todos los átomos del material en el momento actual; el desplazamiento de un átomo se consideraría como un estado vecino del primero en este ejemplo. Típicamente la comparación entre estados vecinos se repite hasta que se encuentre un estado óptimo que minimice la energía del sistema, hasta que se cumpla cierto tiempo computacional u otras condiciones.

El vecindario de un estado  $s$  está compuesto por todos los estados a los que se pueda llegar a partir de  $s$  mediante un cambio en la conformación del sistema. Los estados vecinos son generados mediante métodos de Montecarlo. El método de evaluación de estados vecinos es fundamental para encontrar una solución óptima global al problema dado. Los algoritmos heurísticos, basados en buscar siempre un estado vecino mejor (con energía más baja) que el actual se detienen en el momento que encuentran un mínimo local de energía. El problema con este método es que no puede garantizar que la solución encontrada sea un óptimo global, pues el espacio de búsqueda explorado no abarca todas las posibles variaciones del sistema.

La probabilidad de hacer la transición al nuevo estado  $s'$  es una función  $P(\delta E, T)$  de la diferencia de energía  $\delta E = E(s') - E(s)$  entre los dos estados,

y de la variable  $T$ , llamada temperatura por analogía con el concepto físico de temperatura.

Si  $\delta E$  es negativo, es decir, la transición disminuye la energía, el movimiento es aceptado con probabilidad  $P = 1$ . Cuando  $\delta E$  es positivo la probabilidad de transición  $P$  es siempre distinta de cero, es decir, el sistema puede pasar a un estado de mayor energía (peor solución) que el estado actual. Esta propiedad impide que el sistema se quede atrapado en un óptimo local, lo cual hace que este tipo de algoritmo sea muy recomendable para problemas con óptimos locales.

A medida que la temperatura tiende al mínimo, la probabilidad de transición a un estado de mayor energía tiende a cero asintóticamente. Cuando  $T$  llega a cero, el algoritmo sólo aceptará cambios a estados con menor energía. Debido a esta propiedad, la temperatura juega un papel muy importante en el control de la evolución del sistema. A temperaturas altas, el sistema tenderá a saltos de energía grandes entre los estados, mientras que a temperaturas más bajas, los cambios en energía serán menores.

Así, en cada iteración el algoritmo tiende a encontrar estados con menor energía total. Hay muchas maneras de disminuir la temperatura, siendo la más usual la exponencial, donde  $T$  disminuye con un factor  $\alpha < 1$  en cada paso.

Como el nombre del algoritmo sugiere, la variación de la temperatura durante la computación es una característica distintiva de este método. El algoritmo comienza con un valor de  $T$  muy alto, que va decreciendo en cada iteración siguiendo un cierto protocolo de recocido, que puede ser diferente para cada problema, pero que siempre debe terminar con  $T = 0$ . Así, el sistema será libre inicialmente de explorar una gran porción del espacio de búsqueda, ignorando pequeñas variaciones de la energía entre los estados vecinos evaluados, para, más tarde, centrarse en regiones con estados de baja energía y, al final, cambiar solo a estados con energía menor que la inicial, hasta alcanzar un mínimo.

Este algoritmo se encuentra implementado por Xiang et al. (2012) en el programa R mediante la librería GenSA.

#### 3.5.4. Otros algoritmos metaheurísticos

Además de los algoritmos que se mencionaron en la sección anterior con más detalle, por ser éstos los que se utilizarán en esta tesis como propuestas para la optimización de problemas de degradación de materiales, el campo de los algoritmos evolutivos ha experimentado un gran desarrollo en los últimos años, surgiendo nuevos algoritmos cuyos nombres suelen estar relacionados con el modelo biológico que emplean como ejemplo, se podrían citar: los de Evolución Diferencial con Torneos de Pareto (Differential Evolution with Pareto Tournaments - DEPT); los basados en la Búsqueda de Entorno Variable (Variable Neighborhood Search - VNS), en la Colonia Artificial de Abejas (Artificial Bee Colony - ABC), en el Algoritmo de Búsqueda Gravitacional (Gravitational Search Algorithm - GSA); en el Algoritmo de las Luciérnagas (Firefly Algorithm- FA); en el Algoritmo de Enseñanza-Aprendizaje (Teaching-Learning-Based Optimization - TLBO) o los basados en el Algoritmo de las Ranos Saltarinas (Shuffled Frog Leaping Algorithm - SFLA). Las ideas en las que se basan estos algoritmos

están íntimamente relacionados con su nombre, así el algunos aprovechan las características de la naturaleza como la fuerza de la gravedad (MO-GSA), o algoritmos que se apoyan en otros comportamientos colectivos (Swarm Intelligence) de insectos como las abejas, las luciérnagas o anfibios como las ranas.

## 3.6. Aplicaciones de Algoritmos Evolutivos

Las aplicaciones de los algoritmos evolutivos abarcan campos tan diversos como la Biología, la Economía, la Ingeniería, la Medicina, la Informática, etc. Se presentan a continuación algunos ejemplos concretos.

### 3.6.1. El algoritmo genético estadístico (AGE)

El Algoritmo Genético Estadístico, AGE, parte de una muestra estadística de posibles soluciones del problema y, con un algoritmo de escalamiento iterativo basado en la reducción del espacio de búsqueda a partir de la dispersión de la función de aptitud respecto al mejor individuo de la muestra, converge hacia la solución exacta.

Para una muestra estadística de estados iniciales (con un número de estados suficientes), el estado con mayor probabilidad de supervivencia es el óptimo global. El AGE premia las transiciones hacia estados por encima del valor medio, así que utilizando un amplio colectivo de cromosomas divididos en varios estados iniciales, y ejecutándolo el menor número de generaciones posibles, pero suficientes para una buena selección, junto con el escalamiento basado en la contracción del espacio de búsqueda a partir de la dispersión de la función de aptitud respecto al mejor cromosoma de la muestra, permite converger hacia la solución del problema, con el grado de aproximación que se quiera.

Sin embargo, en la práctica para utilizar este criterio se tendría que haber resuelto el problema previamente, porque se necesita conocer los valores de aptitud de todos los estados del espacio. Por este motivo no tiene aplicación directa.

Aplicando AGE y un algoritmo genético simple (AGS) a varias funciones test se observa una mejora en los resultados con AGE. Se comprueba además, que utilizar técnicas determinísticas y estocásticas conjuntamente (métodos “cuasi”) produce mejoras respecto del uso por separado.

Dado que el AGE surge tras forzar la convergencia (hacia el óptimo) de la matriz estocástica que representa las probabilidades de transición entre estados, la dinámica del AGS, opera iterando en cada generación la función objetivo en todos los estados, y penalizando paulatinamente los estados que se encuentran por debajo de la media en cada paso, aumentando así, a través del tiempo, el peso de la probabilidad de saltar hacia el óptimo global. Ese proceso conduce a la convergencia. Como se ha apuntado ya, aporta como novedad la reducción iterativa del espacio de búsqueda, que se logra a partir de la dispersión de la función objetivo respecto al mejor individuo de la muestra ( $\sigma_{\text{máx}}$ ) y del rango de la aptitud en la muestra ( $f_{\text{mejor}} - f_{\text{peor}}$ ). Es de esperar que  $\sigma_{\text{máx}} < f_{\text{mejor}} - f_{\text{peor}}$ .

En varios trabajos sobre AGS, después de un número mínimo de generaciones de AGS, se toma como solución el mejor individuo de cada lanzamiento de esa muestra. Se define:

$$\sigma_{\text{máx}}^1 = \frac{\sqrt{\sum_{i=1}^N (f_{\text{mejor}}^0 - f_i^0)^2}}{N}, \quad (3.6)$$

siendo  $N$  en la ecuación (3.6) el número de poblaciones iniciales. El nuevo radio del dominio para la búsqueda será:

$$r^1 = r_i^1 = \frac{2d_i^0 \sigma_{\text{máx}}^1}{|f_{\text{mejor}}^0 - f_{\text{peor}}^0|} \quad (3.7)$$

donde  $d_i^0$  en (3.7) es la longitud del intervalo inicial de búsqueda en cualquier dimensión del dominio. El proceso se irá completando al construir los intervalos  $(p^1 - r^1, p^1 + r^1)$  donde  $p^1$  será el mejor individuo de la muestra estadística. Se continuará con este tipo de iteración de forma que en la segunda iteración, el resultado para  $\sigma_{\text{máx}}^2$  probablemente será mejor que el correspondiente valor  $\sigma_{\text{máx}}^1$ , y el rango de la función objetivo será menor, resultando:

$$r^2 = r_i^2 = \frac{r_i^1 \sigma_{\text{máx}}^2}{|f_{\text{mejor}}^1 - f_{\text{peor}}^1|}$$

Iterando esta “técnica de contracción del dominio de búsqueda” se obtiene:

$$\begin{aligned} \sigma_{\text{máx}}^t &= \frac{\sqrt{\sum_{i=1}^N (f_{\text{mejor}}^{t-1} - f_i^{t-1})^2}}{N}, \\ r^t &= r_i^t = \frac{r_i^{t-1} \sigma_{\text{máx}}^t}{|f_{\text{mejor}}^{t-1} - f_{\text{peor}}^{t-1}|}. \end{aligned} \quad (3.8)$$

Después de algunas iteraciones, mediante la expresión 3.8, se encuentra un estrecho intervalo que, con alta probabilidad, contendrá al óptimo global.

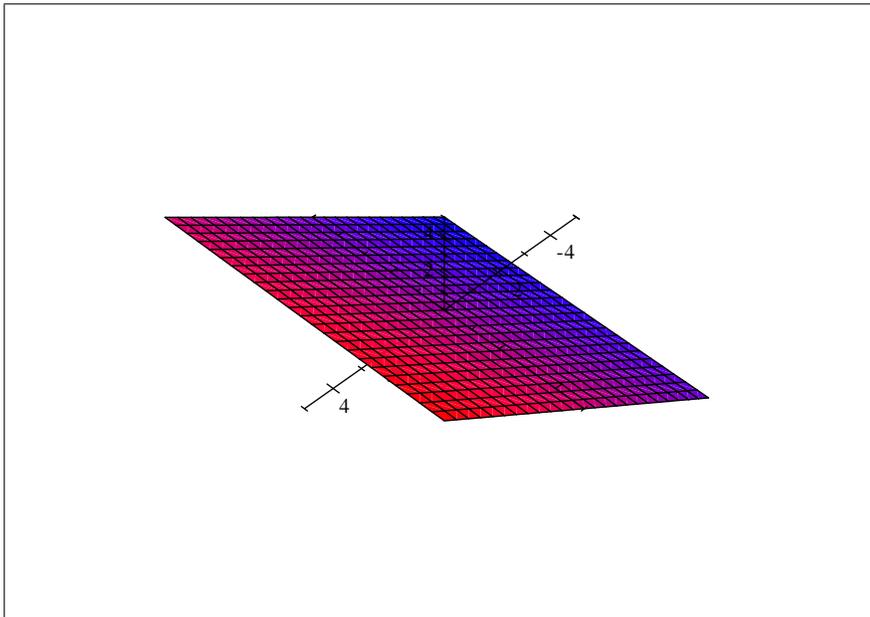
Se ha utilizado el AGE con 4 funciones test distintas, y se ha comparado con otros algoritmos evolutivos:

- CHC (Combination Homogeneous Crossover): utiliza una selección de cruce generacional, basado en cruzar dos a dos los cromosomas de forma aleatoria y uniforme, y la descendencia se mantiene en una población temporal para establecer un torneo entre ambas; se seleccionan los mejores individuos de entre las dos para pasar a la siguiente generación. El cruce utiliza recombinación homogénea como método de prevención del incesto.
- ESGAT (Elitism SGA Tournament): parte del AGS de Goldberg pero introduce elitismo (el mejor individuo se conserva de una generación a

otra) y selección por torneo (en lugar de la proporcional a la aptitud). La selección por torneo proporciona una forma probabilística de ordenar los individuos basada en la selección y no necesita que la población esté clasificada. Además permite la ejecución paralela en la formación de la población intermedia.

- GENITOR (Basado en el cruce OR: Order-based crossover): se genera sólo una descendencia cada vez y reemplaza al peor miembro de la población inicial. Así asegura que en cada nueva selección los individuos se conservan y las generaciones mejoran o simplemente permanecen invariables. Utiliza la selección basada en la clasificación y la población debe seguir ordenada: el individuo medio es seleccionado para recombinación con probabilidad  $1/N$ , el mejor clasificado, con probabilidad  $B/N$  y el peor  $(2 - B)(1/N)$ , siendo  $B$ , entre 1 y 2, un valor máximo de selección. La presión selectiva se interpola linealmente, resultando el paso de un individuo a otro:  $2(B - 1)/(N(N - 1))$ .
  
- RBC: la población se compone de un único individuo y se muta aleatoriamente un solo dígito del genotipo. Si produce mejor adaptación se acepta como nuevo individuo y sustituye al anterior. Es, pues, un algoritmo de búsqueda local.

Los resultados comparativos se han hecho para funciones de 10 variables, con 50000 evaluaciones para todos, excepto AGE y 30 experiencias en cada caso. El AGE consigue mejor valor medio en 3 de las funciones y en la cuarta alcanza una solución aceptable, teniendo en cuenta el gran intervalo de búsqueda inicial. Hay que destacar, además, su facilidad de convergencia en la función de Rosenbrock,  $f(x, y) = 100(x^2 - y)^2 + (1 - x^2)^2$ , que es especialmente difícil dada su forma plana (el óptimo difiere muy poco de los puntos cercanos).



Función de Rosenbrock.

El AGE se convierte así en un algoritmo de búsqueda fiable que permite obtener una buena solución aproximada rápidamente, o la convergencia al extremo absoluto con un coste moderado.

### 3.6.2. El Algoritmo Genético de Selección de Multirrecombinación, (MRS)

El Algoritmo Genético de Selección de Multirecombinación (MRS) parte de una población de  $n$  individuos y a partir de estos se crea una nueva población de mayor tamaño, mediante cruce, que contiene a los individuos padres y a los individuos hijos. Los individuos hijos sufren mutación pero no así los padres. Este proceso de multirecombinación se realiza hasta completar una generación en la que se seleccionan los  $n$  mejores individuos para la nueva población.

El número de pasos para obtener una generación es arbitrario, aunque en este test permanece constante, fijado desde el inicio. Asimismo, se seleccionan los mejores individuos, que pueden estar repetidos y puede seleccionarse más de una copia del mismo cromosoma. Utilizan cruce de 1 punto, mutación uniforme y selección proporcional a la aptitud.

El MRS no es un proceso estocástico homogéneo, pues las probabilidades de transición dependen de las poblaciones de cromosomas existentes en la muestra en cada iteración. Como se trata de una superposición de  $T$  procesos en los que no se conserva el número de estados, ni el tamaño de la muestra por lo que tal proceso no es markoviano. No obstante, el estudio de su convergencia se puede hacer por otros métodos.

El fundamento de la idea parte de la ergodicidad de la matriz de transición del algoritmo genético. Desde cualquier estado poblacional inicial siempre hay una probabilidad no nula de llegar a cualquier otro estado, entre ellos al óptimo global, en un número finito de generaciones.

El número de generaciones para pasar de un estado inicial aleatorio al óptimo global depende de dicho estado inicial. Para el estado inicial creado a partir de una muestra estadística de AGS, su función de aptitud formada por la suma de todas las funciones de aptitud de sus cromosomas, debería estar en promedio más cerca del estado óptimo que cualquier población inicial aleatoria, por lo que se reduciría el número medio de generaciones de búsqueda. Esto redundará en una disminución del tiempo medio de búsqueda del óptimo global en el AG.

### 3.6.3. Combinaciones con otros algoritmos evolutivos

En redes de neuronas artificiales, los AG tienen también aplicaciones en el cálculo de la red de peso óptimo. Cada topología se entrena usando métodos de adiestramiento predeterminados con la idoneidad atribuida basada en su rendimiento.

La técnica de programación genética (PG) utiliza AG para encontrar la relación subyacente entre conjuntos de datos y especifica esto en forma funcional. Esta técnica utiliza los mecanismos de AGS y los aplica a poblaciones simbólicas o S-expresiones (simplemente una lista de átomos, constantes o variables, y funciones que pueden incluir operaciones lógicas y otras no aritméticas). El aspecto fundamental de la PG es que los ordenadores están aprendiendo a resolver el problema planteado, sin programación específica para hacerlo. Esto no garantiza que esta técnica encuentre solución, ni que la solución encontrada sea la más efectiva.

Otro ejemplo interesante propuesto por Koza es el uso de PG para descubrir una estrategia adecuada para el juego PacMan, que implica un agente artificial moviéndose alrededor de un laberinto, consiguiendo juntos lograr ciertas tareas (el juego es, de hecho, un problema de priorización de tareas).

### 3.6.4. Aplicación de Algoritmos evolutivos en el ajuste de controladores del tipo PID

En Reynoso-Meza et al. (2013) se presenta una aplicación de algoritmos evolutivos al problema de los controladores PID (Proporcional Integral Derivativo), que continúan siendo una solución fiable, robusta, práctica y sencilla para el control de procesos industriales. Éstos son mecanismos de control por realimentación, que calculan la desviación entre un valor medio y el que se quiere obtener, para aplicar correcciones que ajusten el proceso y constituyen la primera capa de control de la gran mayoría de las aplicaciones industriales. En este artículo se presenta un problema de ajuste de parámetros mediante el planteamiento de un problema de optimización, el cual además puede llegar a ser no-lineal, no-convexo y con restricciones. Los autores proponen el uso

de algoritmos evolutivos para dicha optimización en que la función de coste pretende reflejar el balance entre los objetivos deseados por el diseñador.

Aunque se puede abordar dicho problema de diseño con métodos de optimización tradicionales (ver Caballero y Grossmann (2011)), en general los problemas planteados para alcanzar mejores prestaciones suelen ser no-lineales y no-convexos. Además, la incorporación de restricciones, imprescindible en problemas de diseño en contexto realista, suele complicar el proceso de optimización. Los algoritmos evolutivos se han utilizado como una alternativa válida para resolver estos planteamientos en diversos campos de la ingeniería como exponen Saridakis y Dentsoras (2008) y Roy et al. (2008), dada su naturaleza de optimizadores globales y su flexibilidad para manejar funciones de coste altamente no-lineales.

Además de la propuesta de un algoritmo evolutivo aplicado a la optimización de parámetros para PID, también se describe un breve ensayo para comparar diferentes algoritmos evolutivos y el tipo de solución que pueden ofrecer para el ajuste de controladores PID convencionales.

En dicho artículo se evalúan los algoritmos que obtuvieron los puestos primero y segundo en la competición organizada por el Congreso en Computación Evolutiva de la IEEE en 2011. La competición giró en torno a la optimización de aplicaciones reales, incluyendo problemas del ámbito químico, eléctrico, energético y aeroespacial como describen Das y Suganthan (2011). Los mejores resultados fueron obtenidos por los algoritmos evolutivos de tipo Differential Evolution, Simulated Annealing y CMA-SE, motivo por el que se han introducido en este capítulo y se emplearán en el capítulo siguiente para resolver el objetivo principal de esta tesis, la optimización de parámetros en curvas termogravimétricas.

### 3.6.5. Optimización para Descubrir Motifs en Secuencias de ADN

Uno de los ejemplos de utilización de diferentes algoritmos evolutivos es el desarrollado en la tesis doctoral de González-Alvarez (González-Álvarez, (2013), en la que se presentan diferentes algoritmos aplicados al problema del descubrimiento de Motifs. Se trata de una aplicación a un problema biológico cuyo principal objetivo consiste en descubrir pequeños patrones repetidos (denominados motifs) en conjuntos de secuencias de ADN que puedan poseer cierto significado biológico, por ejemplo, ser el punto de unión entre proteínas. Estos motifs se encuentran mezclados entre una gran cantidad de información biológica en las secuencias de ADN por lo que descubrirlos no es trivial y el uso de algoritmos evolutivos resultó ser esencial para la propuesta planteada en dicha investigación. Concretamente la utilización del algoritmo Differential Evolution fue uno de los que resultó más eficaz.



## Capítulo 4

# Algoritmo memético para optimización de curvas TG

### 4.1. Introducción

En este capítulo se analizará con más detalle el problema de la optimización de parámetros para el caso de la modelización con mezcla de logísticas de las curvas TG. Para ello se propondrá un algoritmo memético basado en la aplicación reiterada de algoritmos evolutivos de última generación que han sido contrastados con éxito por la comunidad científica.

Como ya se comentó en el capítulo anterior, el problema de la optimización de parámetros es uno de los puntos críticos que tiene la modelización de datos procedentes de análisis térmico, pues debido a la complejidad de los modelos usados para el ajuste, no es fácil encontrar el óptimo global en muchos casos.

Con la intención de encontrar un método automático para la optimización de los parámetros de los modelos cinéticos, se analizarán los 3 algoritmos evolutivos, entre los de mayor éxito hasta el momento, y se propondrá un método iterativo basado en el empleo de estos algoritmos de forma combinada. Para ello se simularán distintos procesos de degradación complejos que reflejen situaciones similares a los datos reales de laboratorio, con objeto de reproducir aquellos procesos con solapamiento de más difícil ajuste. Al tratarse de funciones simuladas, de las que se conoce su expresión real, es posible controlar la bondad de los modelos y así verificar qué combinación de algoritmos permite un mejor ajuste. Para medir la bondad del ajuste se utilizará el método de mínimos cuadrados, cuya su expresión más habitual es:

$$\text{mín} \sum_{i=1}^n ((y_i - m(t_i; \theta_i))^2 \quad (4.1)$$

En (4.1)  $y_i$  hace referencia a los valores de la muestra simulada, o, en el caso real, el valor de la masa y  $m(t_i; \theta_i)$  representa los valores calculados con los

diferentes modelos cinéticos para un conjunto de parámetros  $\theta_i$ . Por ejemplo, supuesta una relación logística para la cinética de descomposición,  $Y(t)$  con el tiempo  $t$ :

$$m(t_i; \theta) = \sum_{i=1}^k w_i f(a_i + b_i t) \quad (4.2)$$

$$f(t) = \frac{\exp t}{1 + \exp t} \text{ (función logística) y donde } \theta_i = (w_i, a_i, b_i) \quad (4.3)$$

donde  $w_i$  representan pérdidas de peso en cada proceso,  $a_i$  son parámetros de escala y  $b_i$  representa la velocidad de pérdida de peso, para los  $k$  escalones que presente la función  $Y(t)$ .

El objetivo es encontrar los parámetros de los diferentes modelos, en nuestro ejemplo modelos basados en mezcla de logísticas y logísticas generalizadas, que mejor se ajusten a los datos reales de las curvas termogravimétricas.

Una forma práctica de conseguir algoritmos más eficaces y adaptados a problemas concretos consiste en hibridar los algoritmos evolutivos con otras técnicas (Moscato, 1999). Los Algoritmos Evolutivos obtenidos mediante la hibridación con técnicas de búsqueda local son denominados algoritmos meméticos. Un procedimiento usual de mejora es aplicar el método de búsqueda local a los nuevos miembros de la población, para explorar las mejores regiones de búsqueda obtenidas durante el muestreo global del algoritmo evolutivo, o bien para utilizar las soluciones como valor de los parámetros iniciales del siguiente algoritmo a aplicar.

## 4.2. Aplicación de Algoritmos Evolutivos para la estimación de parámetros del modelo logístico

En esta sección se presenta uno de los trabajos iniciales llevados a cabo para la estimación de parámetros en un modelo de mezcla de funciones logísticas mediante algoritmos genéticos. Concretamente, se propone la aplicación de un algoritmo genético estándar, que es uno de los trabajos iniciales en este contexto.

Para el ajuste de los datos a una curva logística es necesario calcular los valores de los parámetros que intervienen en la ecuación, se trata de programar un algoritmo para un modelo de mezcla de funciones logísticas, así que sería un problema del tipo de regresión no lineal. Nuestra primera propuesta fue utilizar un algoritmo genético simple, con una probabilidad de cruce del 70% y una probabilidad de mutación del 8%. El cruce se realiza por un punto y la mutación consiste en sumar o restar a cada gen un número aleatorio elevado al cubo, favoreciendo la exploración cercana al valor del gen. La población está formada por 1000 individuos con un cromosoma de longitud variable en función del número máximo de logísticas. El primer gen de cada cromosoma indica

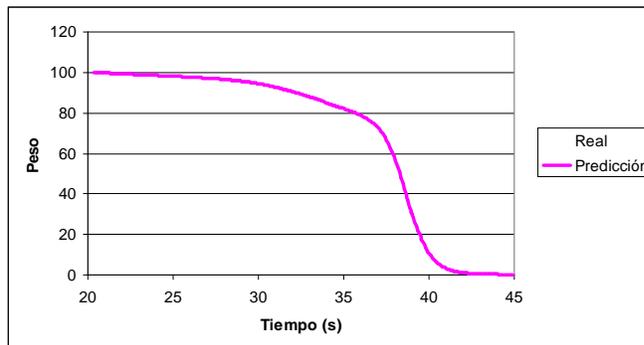


Figura 4.1: Ajuste de una curva TG mediante AG.

cuántas logísticas utilizará el individuo correspondiente. El resto de genes del cromosoma son números reales que representan los tres parámetros básicos de la función logística ( $w$ ,  $a/b$  y  $b$ ) para cada una de las posibles logísticas. Cada uno de estos parámetros están codificados en base al número de dígitos que contienen, por ejemplo, el peso  $w$  se mueve en un rango entre 0 y 99.9 por lo que utilizamos tres genes ( $w_0, w_1, w_2$ ) para codificarlo, cada uno de los cuales tiene un rango de 0 a 9. A la hora de decodificar el parámetro  $w$  se aplicaría la expresión:  $w = w_2 * 10^2 + w_1 * 10^1 + w_0 * 10 - 1$ . De este modo, cada uno de los genes tiene una contribución equitativa al valor del parámetro. El parámetro  $a/b$  está codificado también con tres genes (rango 0 a 99.9) mientras que el parámetro  $b$  está codificado con tres genes pero en el rango 0 a 9.99. El número de generaciones hasta encontrar el óptimo corresponde al punto donde la calidad se vuelve constante. Esta primera aproximación al problema puede verse en Ríos-Fachal et al. 2004. Posteriormente sería analizada la epistasis del algoritmo propuesto, que se divulgaría en dos congresos específicos (Ríos-Fachal et al. 2005 y Ríos-Fachal et al. 2006).

### 4.3. Propuesta de un método para la optimización

Los algoritmos genéticos, como el propuesto en la sección anterior, son un tipo especial de métodos de búsqueda de óptimos basados en la probabilidad. Hace evolucionar una población de individuos sometiénolos a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como a una selección basada en un criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son rechazados.

Bajo una condición muy débil (que el algoritmo mantenga el elitismo: guarda siempre el mejor elemento sin hacerle ningún cambio), el algoritmo converge en probabilidad al óptimo, es decir, al aumentar el número de iteraciones (genera-

ciones), la probabilidad de tener el óptimo en la población tiende a 1. Uno de los problemas que presenta esta primera aproximación al problema es el derivado de la necesidad de dar valores para una solución inicial del algoritmo.

Para evitar este problema se propone una nueva metodología basada en la utilización de algoritmos que no precisen de un conjunto inicial de parámetros, el de Evolución Diferencial o Differential Evolution (DE) o el Covarianza Matrix (CMA-ES), que como se comentó en el capítulo 3, que parten de una región factible de valores iniciales, por lo que al dar un intervalo amplio se evitan problemas de convergencia. Luego, una vez aplicado este primer método, que en algunos casos ya consigue un óptimo global del problema, se utilizará la solución propuesta con un método más preciso y que converge con mayor probabilidad al óptimo global, el Simulated Annealing (SA) para encontrar el óptimo de los parámetros partiendo de esa solución inicial y alcanzar la solución óptima.

Para analizar situaciones complejas que puedan presentarse en la práctica de esta propuesta, se presenta en la sección siguiente un estudio de simulación de diferentes casuísticas, a fin de testar la convergencia del método propuesto. En esta sección se presenta un estudio de simulación con el objeto de analizar cómo se comporta el modelo propuesto. Este estudio presenta una alternativa a la estimación de modelos de ajuste paramétricos utilizados en el análisis térmico. El objetivo consiste en proponer un método de optimización alternativo para evaluar y ajustar curvas de análisis térmico complejas, específicamente curvas TG y sus primeras derivadas con posibles solapamientos. El objetivo es la estimación de los parámetros óptimos correspondientes a los modelos cinéticos de ajuste aplicado a las curvas de termogravimetría, utilizando algoritmos evolutivos, concretamente: el algoritmo de evolución diferencial (DE), el de recocido simulado (SA) y el de estrategia evolutiva mediante adaptación de la matriz de covarianzas (CMAES). Estos métodos no necesitan incluir un vector con los valores iniciales de los parámetros, como se exige habitualmente. A pesar de sus potenciales beneficios, la aplicación de estos métodos no es común en el contexto de la estimación de las curvas de análisis térmico.

La metodología que se propone en esta tesis se ha aplicado a problemas de análisis térmico. Para ello se simuló las curvas de TG y se ajustaron mediante un modelo de mezcla de logísticas generalizadas, en el que cada componente logística representa un proceso de degradación térmica. La simulación de las curvas de TG en cuatro escenarios diferentes suponiendo la existencia de superposición de procesos, nos permite la evaluación de los resultados finales y por lo tanto podemos validar el procedimiento propuesto: dos procesos de degradación no solapada simulada usando dos logísticas generalizadas, dos procesos superpuestos, cuatro procesos no solapados y cuatro procesos solapados dos a dos. Se elige como función objetivo la función del error cuadrático medio y los algoritmos anteriores se han aplicado por separado y juntos, es decir, tomando la solución final del algoritmo DE como solución inicial de los demás. Los resultados muestran que los algoritmos evolutivos proporcionan una buena solución para el ajuste de las curvas de TG simulados, mejor que la proporcionada por los métodos tradicionales.

## 4.4. Estudio de simulación

Como ya se ha comentado en el capítulo 3, las curvas de TG muchas veces están compuestas por una mezcla de diferentes procesos. Estos procesos suelen aparecer superpuestos, impidiendo la correcta caracterización de cada uno. Para resolver este problema, se han propuesto muchos modelos paramétricos de ajuste cinético y el ajuste de un modelo de regresión no lineal basado en una mezcla de funciones logísticas generalizadas ha sido utilizado con éxito para la separación de los procesos superpuestos en las curvas de TG (véase Artiaga et al, 2005). En el caso de las curvas TG, la mezcla de funciones logísticas generalizadas se ha aplicado para la identificación y descripción de los procesos de degradación térmica (López-Beceiro, 2011).

Este modelo propone descomponer la traza de la curva TG en varias funciones logísticas, suponiendo que la cinética de degradación de cada componente de la muestra está representada por una o la suma de unas pocas funciones. Incluso en el caso de materiales homogéneos, se supone que pueden existir varias estructuras diferentes, cada una siguiendo su cinética específica, que puede ser diferente de la de las demás.

En este caso, se han utilizado los paquetes de R relacionados con los algoritmos evolutivos mencionados y se ha considerado el siguiente modelo de regresión lineal:

$$y_i = m(x_i, \theta) + \varepsilon_i, i = 1, 2, \dots, n$$

donde la variable respuesta y la variable independiente son  $y_i$  (la masa en las curvas TG) y  $x_i$  (tiempo o temperatura), respectivamente;  $m(x_i, \theta)$  es el modelo de ajuste,  $\theta$  es el vector de parámetros del modelo, estimado por mínimos cuadrados y  $\varepsilon_i$  son los errores, supuestos con distribución normal de media 0 y varianza constante.

Los residuos  $\varepsilon_i$  están definidos por:

$$\varepsilon_i(\theta) = y_i - m(x_i; \theta), i = 1, 2, \dots, n$$

El vector óptimo de los parámetros de  $\theta$  se obtiene minimizando la función de error (3), habitualmente con la suma de errores cuadráticos.

$$SSE(\theta) = \sum_{i=1}^n (y_i - m(t_i, \theta))^2 \quad (4.4)$$

Para la obtención del mínimo es necesario utilizar un algoritmo de optimización. Muchos programas comerciales utilizan métodos bien conocidos como Nelder-Mead y Newton-Raphson. En este trabajo, se propone el uso de una combinación de algoritmos evolutivos recientes y con otros clásicos como el de Nelder-Mead.

#### 4.4.1. Algoritmos de optimización globales

La presente propuesta consiste en utilizar una combinación de dos métodos de optimización metaheurísticos: el primero se utiliza para la obtención de una solución inicial en una región específica de parámetros, y el segundo se aplica para obtener una solución final partiendo de la solución obtenida con el primer algoritmo.

Se utilizaron tres algoritmos evolutivos diferentes: el de evolución diferencial (DE), el recocido simulado (SA) y el de estrategia evolutiva con adaptación de la matriz de covarianzas (CMAES). Además, los dos algoritmos de optimización global tradicionales se aplicaron con el fin de comparar sus resultados con los correspondientes a la aplicación de algoritmos tradicionales: Nelder-Mead y Newton-Raphson.

El algoritmo DE optimiza un problema intentando mejorar iterativamente una solución candidata respecto a una medida de calidad dada. Es un método heurístico iterativo para optimización global que resulta eficaz en muchos problemas interesantes en ciencia y tecnología. Está particularmente bien adaptado para encontrar el óptimo global de una función de valores reales de parámetros con valores reales, y no exige que la función sea continua o diferenciable. DE ha sido aplicado con éxito en una amplia variedad de campos, desde la física computacional a la investigación de operaciones, como en Price (Price et al., 2001).

Actualmente están disponibles muchas implementaciones de DE y se mantiene una lista de los programas de DE con propósitos de optimización general. Se utilizó el paquete R y la biblioteca "DEoptim". La aplicación de DE fue motivada por el deseo de ampliar el conjunto de algoritmos para la optimización global en el lenguaje R y el entorno para el cálculo estadístico. R permite la creación rápida de prototipos de funciones objetivo, el acceso a una amplia gama de herramientas para el modelado estadístico y tiene gran capacidad de generar fácilmente gráficos personalizados de los resultados (Storn, 2013).

SA es un método para problemas de optimización global utilizado para la localización de una buena aproximación al óptimo global de una función dada en un gran espacio de búsqueda. Lo primero es que presenta una ventaja importante: no es necesario implementar una solución inicial única.

El nombre y la inspiración vienen del recocido en la metalurgia, una técnica que incluye calentamiento y enfriamiento controlados de un material para aumentar el tamaño de sus cristales y reducir sus defectos. Ambos son atributos del material que dependen de la energía termodinámica libre. La calefacción y refrigeración del material afectan a su temperatura y la energía termodinámica libre; mientras una cantidad de enfriamiento lleva a una cantidad de disminución de la temperatura, este proceso de enfriamiento traerá una disminución mayor o menor en la energía termodinámica libre en función de la velocidad a que se produce, es decir, a una velocidad más lenta se proporciona una disminución mayor.

Esta noción de enfriamiento lento se implementa en el algoritmo de recocido simulado como una lenta disminución en la probabilidad de aceptar soluciones

peores, ya que explora el espacio de soluciones. La aceptación de soluciones peores es una propiedad fundamental de las metaheurísticas que permite una más extensa búsqueda de la solución óptima. El método fue descrito de forma independiente por Scott Kirkpatrick, C. Daniel Gelatt y Mario P. Vecchi en 1983 y de Vlado Černý en 1985, y fue implementado en R utilizando el paquete de software "GENSA" de Xiang et al. (Xiang et al., 2012)..

CMAES fue propuesto por Hanse y Ostermeier y es un algoritmo evolutivo basado en principios de la evolución biológica, concretamente en la interacción de variación (vía mutación y recombinación) y la selección: en cada generación (iteración) se generan nuevos individuos (candidatos a solución) por variación, habitualmente de forma estocástica, y luego algunos son seleccionados para la siguiente generación basándose en su idoneidad o valor de la función objetivo. Así, se generan individuos con valores de la función objetivo cada vez menores. En una estrategia evolutiva, los nuevos candidatos a solución se muestrean con una distribución normal multivariante y las dependencias entre variables en la distribución se representan en la matriz de covarianzas.

El método de adaptación de la matriz de covarianzas (CMA) actualiza la matriz de covarianza de la distribución y esta adaptación aumenta el aprendizaje de un segundo modelo de la función objetivo subyacente, similar a la aproximación de la matriz hessiana inversa en el método de optimización clásico Quasi-Newton. Al contrario que los métodos clásicos, se necesitan pocas hipótesis sobre la naturaleza de la función objetivo subyacente.

El algoritmo CMAES está implementado en R por una parte por Trautmann, Mersman y Arnouising en *cmes* y, por otra, por A. Ghalanos en el paquete *parma*.

Este trabajo pretende mostrar una nueva metodología para obtener el ajuste óptimo de los modelos de regresión no lineal de las curvas de TG. El estudio de simulación nos permite comprobar si el procedimiento de optimización propuesto es capaz de proporcionar el ajuste óptimo global, la obtención de la huella real de cada proceso de degradación simulada y evitar mínimos locales. También se muestra una comparación entre métodos y estrategias.

#### 4.4.2. Propuesta de un método para optimizar los parámetros de ajuste

Los parámetros del modelo se estimaron por el método no lineal de mínimos cuadrados. Los fundamentos de este método se describen en Gay (Gay, 1984)

Se utiliza la rutina del método de Levenberg-Marquart para la generación de la secuencia de aproximación al punto mínimo, basado en el algoritmo de "región de confianza", para el cálculo de los valores de los parámetros que minimizan la función objetivo. Este algoritmo fue discutido por Chambers y Hastie (Chambers y Hastie, 1992) y los detalles acerca de su implementación en R se dan en Dennis (Dennis et al., 1981). Hay otros algoritmos comunes como Nelder-Mead, de Newton-Raphson, etc que presentan el mismo problema: la elección de algunos puntos de partida para los diferentes parámetros a estimar. Este es un aspecto

muy importante de la optimización de parámetros ya que, en muchos casos, la solución final depende de la solución inicial.

Como ya se mencionó anteriormente, uno de los problemas que aparecen al utilizar este ajuste es elegir algunos puntos de partida para los diferentes parámetros a estimar. Para evitar este inconveniente, una posibilidad consiste en ejecutar un procedimiento iterativo compuesto por varios métodos de optimización.

El objetivo es implementar un proceso de optimización utilizando la menor cantidad posible de hipótesis. De hecho, la optimización se lleva a cabo con una sola limitación relacionada con el modelo de regresión de mezcla de logísticas: la suma de todos los parámetros  $c_i$  tiene que ser igual a la masa inicial de la muestra. En concreto, se propone la siguiente secuencia: se aplica el algoritmo de DE para comenzar, ya que no necesita la asignación de una única solución inicial y se indica una amplia región de posibles soluciones. Esta región de valores posibles de los parámetros ha sido elegida suponiendo que no se sabe prácticamente nada de los procesos individuales de la degradación que se superponen en cada curva de TG y entonces, se valida el procedimiento en el peor de los casos. La función objetivo a minimizar es la expresión 4.4. La solución final se obtiene después de 120.000 iteraciones en los casos complejos, y después de 10000 iteraciones en las situaciones más simples.

Finalmente, se comparan los algoritmos CMA-ES y SA. La solución final obtenida por el DE se utiliza como solución inicial en estos métodos. La ventaja es que se obtuvo una buena solución inicial por el DE sin saber nada acerca de la importancia, la ubicación y la forma de los procedimientos individuales que componen cada curva de TG. Las soluciones obtenidas por separado por SA y CMA-ES se comparan con un valor de referencia; si la diferencia es menor que una cierta tolerancia (Tol) esa solución es elegida como óptima. En el presente estudio el valor de tolerancia utilizado fue igual a  $10^{-12}$ .

Las diferentes soluciones obtenidas se comparan con las soluciones obtenidas por métodos como Nelder-Mead y Newton-Raphson, aplicada en este segundo paso. Finalmente, los procesos de degradación individuales reales en una curva de TG se comparan con los obtenidos por el ajuste del modelo de mezcla de logísticas, usando la presente metodología de optimización.

La secuencia del procedimiento de optimización propuesto puede verse en el diagrama de flujo de la Figura 4.2.

Es importante tener en cuenta que los valores extremos de esta región inicial de parámetros se definen atendiendo a las características de los datos simulados (se supone evaluado entre 0 y 800 ° C) y la experiencia en el ajuste de este tipo de modelo (se eligieron valores muy extremos).

### 4.4.3. Generación de datos

El modelo empleado para generar los experimentos de simulación sigue la pauta  $y_i = m(x_i) + \varepsilon_i$ , siendo  $m(x_i)$  la función suma de logísticas generalizadas que se ha propuesto en el capítulo III (4.5), y donde  $\varepsilon_i$  son los errores del modelo.

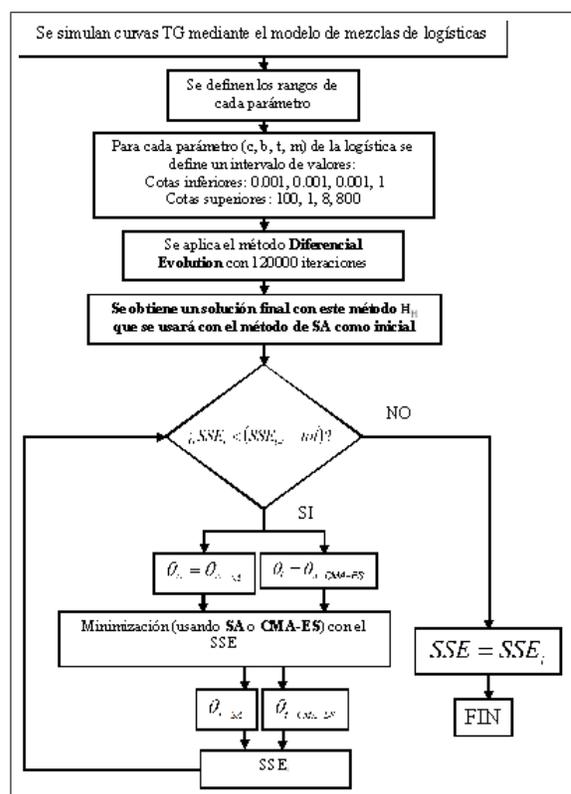


Figura 4.2: Diagrama de flujo para la simulación por Montecarlo de curvas TGA con la secuencia para optimizar los parámetros que minimizan la suma de cuadrados.

A fin de obtener una simulación de la curva con ruido, se han de seguir los siguientes pasos:

1. Siendo  $n$  el tamaño de la muestra, generar los valores  $x_1, x_2, \dots, x_n$ , a partir del valor de  $\delta$ , que representa la distancia entre dos puntos consecutivos. De este modo podemos establecer:

$$x_i = \delta \cdot i, \quad i = 1, \dots, n$$

2. Una vez obtenidos dichos valores  $x_1, x_2, \dots, x_n$ , y a partir de la especificación de  $k$  componentes logísticas generalizadas, podremos simular la señal  $m(x_i)$  para  $i = 1, 2, \dots, n$ , según 4.5.
3. Se obtienen los valores de la señal con ruido:  $Y_i = m(x_i) + \varepsilon_i$ , donde  $\varepsilon_i$  son simulados imitando el error producido por el aparato STA. Estudios previos han permitido ajustar dicho error a un modelo  $AR(1)$ :  $\varepsilon_i = \rho\varepsilon_{i-1} + a_i$ , con  $\rho \in (-1, 1)$  y los  $a_i$  son valores independientes simulados de una  $N(0, \sigma_a^2)$ , con  $\sigma_a^2 = \sigma^2(1 - \rho^2)$ .

La simulación de cada componente  $a_i$  se ha llevado a cabo haciendo uso del método del Montecarlo.

#### 4.4.4. Resultados de las simulaciones

Con el fin de evaluar la fiabilidad de la instalación y por lo tanto validar el procedimiento explicado anteriormente, se llevó a cabo un estudio de simulación.

Los procesos de degradación térmica de los materiales observados se caracterizan por termogravimetría (TG). El modelo de mezcla de logísticas generalizadas es una alternativa muy útil para ajustar las curvas de TG y obtener sus diferentes procesos de degradación que muchas veces aparecen solapados. Es por ello que los datos simulados se obtienen utilizando este modelo de ajuste:

$$m(\theta; t) = \frac{c}{(1 + \exp(-b(m - t)))^{(1/\tau)}} \quad (4.5)$$

donde  $t$  es el tiempo o la temperatura del experimento,  $\theta = (c, b, \tau, m)$ , es el vector de parámetros,  $c$  está relacionada con la masa de cada proceso de degradación,  $b$  es un parámetro relacionado con la velocidad de cambio en cada proceso (en este caso de la tasa de degradación),  $m$  es el valor (por lo general, el tiempo o la temperatura) en que ocurre la velocidad máxima de cambio en cada proceso y, finalmente,  $\tau$  es un parámetro relacionado con la asimetría de cada proceso.

Se simularon cuatro escenarios diferentes para la evaluación de las metodologías propuestas en una muestra representativa de los casos posibles. Estos se definen con diferente grado de superposición y diferente número de procesos de degradación simulados. En las figuras 4.3 y 4.4 se muestran los diferentes escenarios: 2 procesos de degradación no solapados, 2 procesos solapados, 4 procesos de degradación no superpuestas y 4 procesos superpuestos dos a dos.

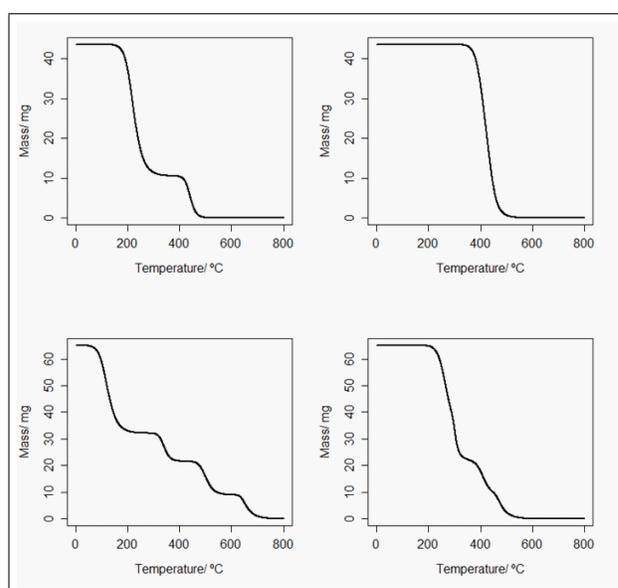


Figura 4.3: Escenarios estudiados simulación: 2 procesos de degradación no superpuestos, 2 procesos solapados, 4 procesos de degradación no superpuestos y 4 procesos superpuestos dos a dos.

Cada curva TG se evaluó en 1600 puntos entre 0° y 800°C. En lo sucesivo, se considerará que la variable dependiente es la masa de la muestra, mientras que la variable independiente es la temperatura.

Como se evalúa la función de ESS en el mismo número de puntos, los valores de estas funciones objetivo obtenidos por diferentes métodos y en diferentes escenarios se pueden comparar (véase la Tabla 4.1).

Como se puede observar en la Tabla 4.1, en el caso de dos procesos, no se necesita el segundo paso de la metodología propuesta, la aplicación de algoritmo de DE es suficiente para obtener un ajuste perfecto. En el caso de los 4 procesos de degradación, la combinación de dos algoritmos evolutivos proporciona una solución mejor que utilizando el algoritmo Nelder-Mead en el segundo paso (Nelder y Mead, 1965). En todos los casos se llegó a una buena solución final, incluso en el más desfavorable de ellos, 4 procesos superpuestos dos a dos. Una idea de la bondad del ajuste se puede obtener también en la observación de las Figuras 4.3 y 4.4. Las curvas ajustadas reproducen fielmente las huellas de las curvas TG simuladas.

Escenario	Método	SSE	SSE_Final
Dos procesos no solapados	DE+SA	0	0
Dos procesos no solapados	DE+CMA-ES	0	0
Dos procesos no solapados	DE+Nelder Mead	0	0
Dos procesos solapados	DE+SA	0	0
Dos procesos solapados	DE+CMA-ES	0	0
Dos procesos solapados	DE+Nelder Mead	0	0
Cuatro procesos no solapados	DE+SA	77,17481	0,00078347
Cuatro procesos no solapados	DE+CMA-ES	77,17481	$1,14 \cdot 10^{-4}$
Cuatro procesos no solapados	DE+Nelder Mead	77,17481	24,85083
Cuatro procesos solapados	DE+SA	8,191	0,126769
Cuatro procesos solapados	DE+CMA-ES	8,191	3,364080
Cuatro procesos solapados	DE+Nelder Mead	8,191	1,587986

**Tabla 4.1.** Valores para SSE obtenidos por varios métodos: DE y SA y CMA-ES o Nelder Mead

Las comparaciones entre la primera derivada de las curvas de TG simuladas (DTG) y la primera derivada del modelo ajustado están representados en las Figuras 4.6 y 4.7. Los errores de ajuste pueden amplificarse cuando se observan las curvas de DTG, pero podemos observar que éste no es el caso de este estudio. La perfecta reproducción de las tendencias DTG es otro indicio de la bondad del ajuste.

Incluso en el peor de los casos en que los 4 procesos de degradación están altamente superpuestos, cada modelo logístico generalizado reproduce la tendencia de la logística simulada correspondiente (Figura 4.8).

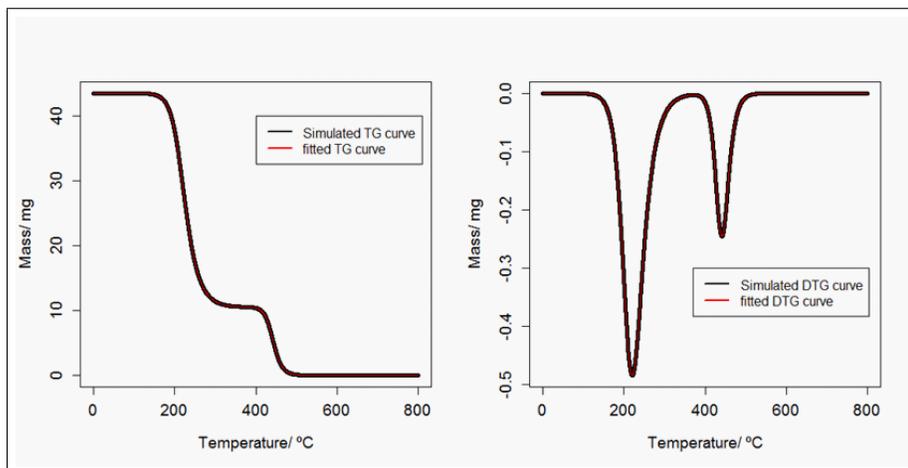


Figura 4.4: Dos procesos de degradación no solapados. Panel izquierdo: ajuste TG. Panel derecho: curvas DTG simulados y ajustados.

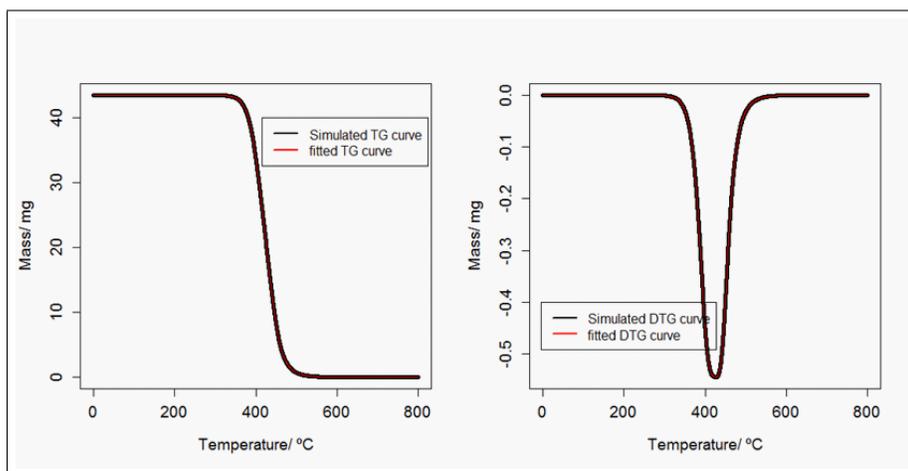


Figura 4.5: Dos procesos de degradación superpuestos. Panel izquierdo: ajuste TG. Panel derecho: curvas DTG simulados y ajustados.

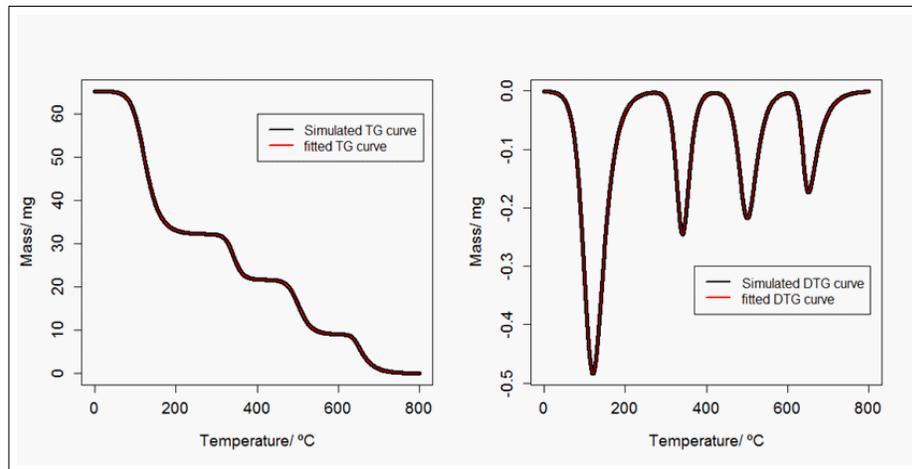


Figura 4.6: Cuatro procesos de degradación no solapados. Panel izquierdo: ajuste TG. Panel derecho: curvas DTG simulados y ajustados.

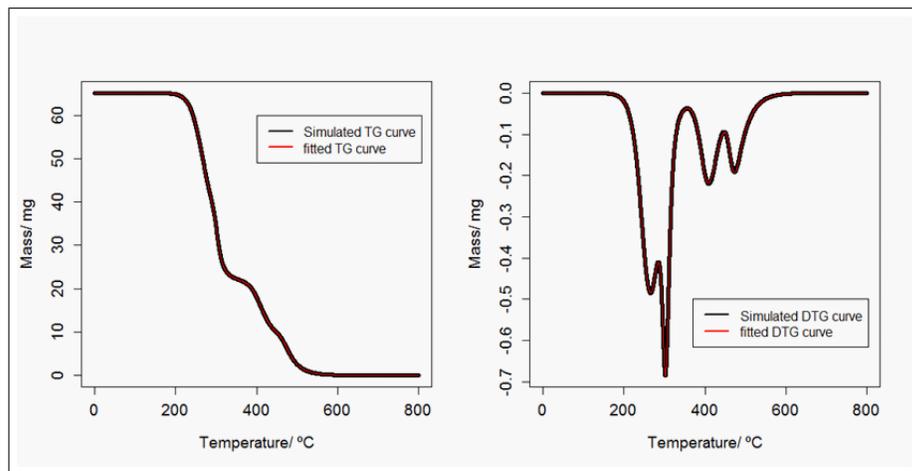


Figura 4.7: Cuatro procesos de degradación superpuestos dos a dos. Panel izquierdo: ajuste TG. Panel derecho: curvas DTG simulados y ajustados.

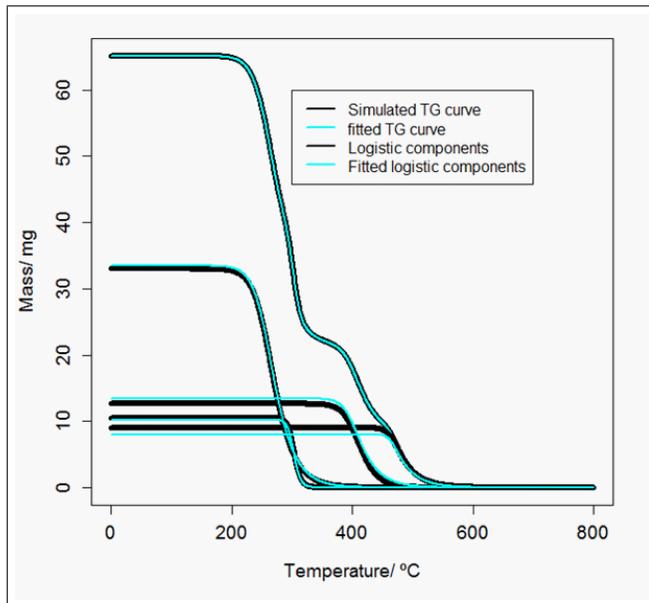


Figura 4.8: Cuatro procesos de degradación superpuestos dos a dos. Comparación entre los procesos de degradación individuales: simulados y ajustados.

## 4.5. Conclusiones sobre el método propuesto

En este capítulo se ha propuesto una metodología alternativa para la obtención de los parámetros óptimos de los modelos de ajuste que fue presentado y aplicado en las curvas TG. Se utilizó para ello el modelo de mezcla de funciones logísticas generalizadas.

Este procedimiento de optimización se puede implementar sin asumir una solución inicial para cada problema, siendo suficiente dar un intervalo en el que tome valores el parámetro concreto a estimar, que en este contexto resulta trivial.

El valor óptimo real se logró en los escenarios de simulación sencillos: dos procesos solapados o no. En los escenarios más complejos se obtiene un valor de la función objetivo cercano a cero. De hecho, los procesos de degradación simulados individuales fueron ajustados con precisión a través de las funciones logísticas generalizadas.

La mejor solución se obtuvo mediante el algoritmo memético que combina dos algoritmos evolutivos, DE y SA en el caso más complejo de cuatro procesos solapados dos a dos y para el escenario de dos procesos solapados se obtienen buenas soluciones con cualquiera de los métodos, recomendando el más rápido que resulta ser el que emplea un algoritmo evolutivo (por ejemplo el, DE) con otro clásico como el de Nelder-Mead.

Este procedimiento mediante el uso de un algoritmo memético se puede aplicar en una amplia gama de problemas de optimización no lineales relacionados con la química analítica y la quimiometría, usando el software libre estadístico R.

## Capítulo 5

# Modelización de la degradación de un nanocompuesto de PS con nanotubos de carbono

### 5.1. Introducción

En este capítulo se aplicarán los modelos y propuestas para su optimización vistos en los capítulos anteriores a una muestra de un nuevo material que se obtiene incorporando a un material polimérico, el poliestireno, diferentes porcentajes de nanotubos de carbono.

El poliestireno (PS) es un polímero termoplástico que se obtiene de la polimerización del estireno. La primera producción industrial de poliestireno fue realizada por BASF, en Alemania, en 1930. Desde entonces los procesos de producción han mejorado sustancialmente y el poliestireno ha dado lugar a una industria sólidamente establecida. El poliestireno es hoy el cuarto plástico más consumido, por detrás del polietileno (PE), el polipropileno (PP) y el policloruro de vinilo (PVC).

Por otra parte, los nanotubos son estructuras tubulares cuyo diámetro es del orden del nanómetro. Pueden ser de muchos materiales como silicio o nitruro de boro, pero generalmente el término se refiere a los nanotubos de carbono.

Los nanotubos de carbono son una forma alotrópica, como el diamante, el grafito o los fullerenos. Su estructura procede de una red hexagonal de carbono enrollada sobre sí misma, y dependiendo del grado de enrollamiento serán monocapa (de pared simple) o multicapa.

Entre sus propiedades están:

- Se pueden considerar unidimensionales por su elevada razón longitud/radio,

ya que el radio suele ser inferior a 2 *nm* y la longitud puede alcanzar los 105 *nm*.

- Desde el punto de vista eléctrico, pueden ser desde semiconductores hasta superconductores.
- Mecánicamente, son la fibra más resistente que puede fabricarse hoy y, además, pueden deformarse notablemente manteniéndose en régimen elástico.
- Térmicamente son excelentes conductores y muy estables, incluso a temperaturas de 2800 °C en el vacío o 750 °C en el aire (los alambres metálicos de los microchips se funden entre 600 °C y 1000 °C). Estas propiedades pueden modificarse encapsulando metales o gases en su interior, por lo que serían excelentes almacenes de hidrógeno.

## 5.2. Nanocompuesto de matriz polimérica y nanotubos de carbono

Se ha comprobado que la introducción de nanotubos de carbono en una matriz de polímero afecta a la estabilidad térmica de los materiales compuestos. Mientras que los efectos observados difieren dependiendo de muchos parámetros, la tendencia general es un aumento en la temperatura de degradación y la reducción en la velocidad de degradación, cuantificada como la disminución de masa en el tiempo.

El propósito de este estudio es determinar cómo influyen los nanotubos de carbono en el proceso de degradación principal de compuestos hechos con poliestireno que contienen 2, 3 y 5 % de nanotubos de carbono.

Los experimentos termogravimétricos se realizaron en una atmósfera de nitrógeno, en múltiples rampas de calentamiento lineales. Se evalúan los efectos de los nanotubos en la degradación del poliestireno, y se aplica un modelo recientemente desarrollado con el fin de obtener los parámetros cinéticos significativos del proceso principal, adaptado al contexto termogravimétrico. El modelo permite separar el proceso principal de otros procesos, que podrían interferir con el análisis cinético, y restar la masa residual, que produciría un efecto estabilizador aparente. El proceso de degradación principal es claramente estabilizado por la presencia de los nanotubos, aunque esta estabilización es más pronunciada con el menor contenido de nanotubos considerado.

## 5.3. Análisis termogravimétrico del nanocompuesto

### 5.3.1. Introducción al estudio termogravimétrico del PS

Como se comentó en el primer capítulo, el análisis termogravimétrico (TG) es la técnica más utilizada para evaluar la estabilidad térmica de polímeros

y compuestos (Bicerano, 1999), ya que permite la comparación directa de las muestras y también su análisis cinético.

En el capítulo II, se presentaron varios métodos de modelización para determinar los parámetros cinéticos de las reacciones en estado sólido. Los más empleados son los "model fitting" y los "model free" o isoconversionales. La propuesta de esta tesis del uso de mezcla de logísticas (Cao et. al. 2004 y Naya et al., 2006) y de logísticas generalizadas (López-Beceiro, 2011) encaja plenamente en la metodología "model fitting", con la particularidad de que, en contra de lo habitual, estos modelos no están basados en el de Arrhenius.

El enfoque de la metodología "model fitting", en principio, supone un mecanismo fijo durante toda la reacción, que implica el ajuste de tiempo de conversión de datos o la razón de conversión a algunos modelos, para determinar las constantes de velocidad y la energía de activación (Conesa, 2000). La bondad del ajuste de estos modelos puede ser mejorada mediante la introducción de parámetros adicionales, aunque esto puede comprometer su significado físico.

Los modelos isoconversionales utilizan datos obtenidos a tres o más velocidades de calentamiento y permiten que la energía de activación varíe con la temperatura. Por tanto, este método libre permite más de un mecanismo en el curso de una reacción. Una de las razones para su uso extensivo es su aptitud para la obtención de valores de energía de activación sin hipótesis modelísticas (Černý, 1985). Sin embargo, aunque no asumen modelos, sí que prácticamente en la totalidad de los casos suponen que éstos modelos "libres" estarán basados en el de Arrhenius (European Polymer Journal 49 (2013) 2233-2246)

Por otra parte, la energía de activación,  $E$ , como función de la conversión,  $\alpha$ , sin elección de modelo de reacción, de datos no isotérmicos puede obtenerse con el método isoconversional de Flynn y Wall (Flynn y Wall, 1966). De acuerdo con éste, los valores de la constante  $E$  deben obtenerse en el caso de un único estado de descomposición, mientras que para procesos con varios pasos  $E$  varía con  $\alpha$ , debido a la variación en las contribuciones relativas de los pasos individuales a la velocidad de reacción total. Además, los métodos de modelo libre tienen desventajas importantes, y suele ser necesario un modelo de reacción para la descripción cinética completa de la reacción de estado sólido (Flynn y Wall, 1966).

El enfoque utilizado en este trabajo para el estudio de poliestireno (PS) que contiene diferentes cantidades de nanotubos de pared múltiple de carbono (MWCNT) es un procedimiento basado en la mezcla de funciones logísticas generalizadas, que consiste en ejecutar el ajuste de un modelo dado a los datos termogravimétricos, ajustando los valores de los parámetros por minimización de la discrepancia del modelo respecto de los valores experimentales, para lo que se han utilizado los algoritmos evolutivos presentados en el capítulo anterior.

### 5.3.2. Datos experimentales

Las muestras de PS conteniendo diferentes cantidades de nanotubos de carbono de pared múltiple fueron preparadas mediante un procedimiento de dos pasos en una extrusora de dos tornillos en un proceso de fundido y mezclado.

En el primer paso, se preparó un baño de matriz inorgánica usando un tornillo a una velocidad de 600 rpm, y en el segundo paso se diluyó el baño con PS para producir nanocompuestos al 2, 3 y 5 % de peso. La cadena extrusada fue finalmente enfriada en un baño de agua y cortada en gránulos.

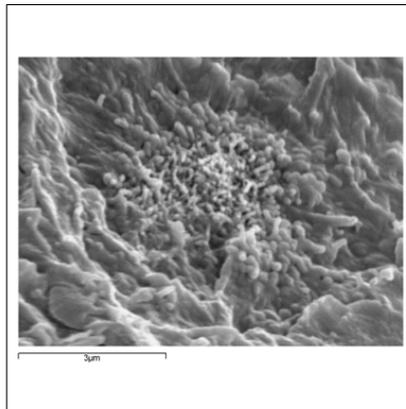


Figura 5.1: Fotografía de una muestra de PS sin nanotubos.

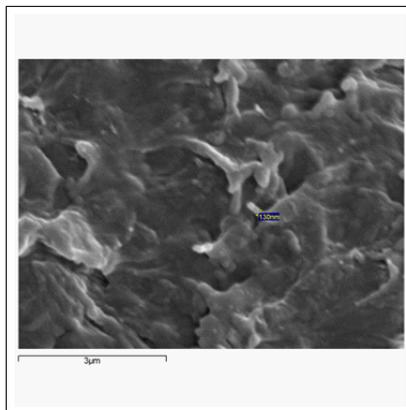


Figura 5.2: Fotografía de una muestra de PS con un 2 % de nanotubos de carbono.

Las muestras fueron analizadas en una termobalanza, TG 2950 de la marca TA Instruments. Por razones de disponibilidad los MWCNT se testaron en otro aparato, un TA Instruments TG Q500. La configuración experimental consistió en rampas de calentamiento lineales en 2, 5 y 10 °C/min con atmósfera de 100 ml/min de N<sub>2</sub>.

La masa de la muestra fue en todos los casos aproximadamente de 3 mg, ex-

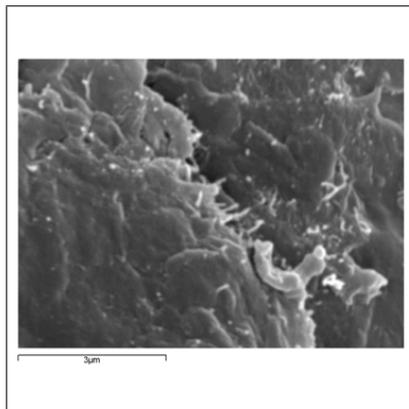


Figura 5.3: Fotografía del PS con un 5% de nanotubos de carbono.

cepto en el de sólo MWCNT para el que se esperaba una menor pérdida de peso. Una muestra de 4.25 mg de MWCNT fue testada en las mismas condiciones, pero sólo a velocidad de  $10^{\circ}\text{C min}^{-1}$ .

Es importante señalar que las masas de las muestras de polímero y compuesto se mantuvieron constantes porque las diferencias en la masa o el volumen podrían afectar seriamente a los resultados termogravimétricos. Ambos instrumentos fueron calibrados siguiendo las instrucciones del fabricante, y para calibrar la temperatura se utilizó la temperatura de Curie del níquel. Se comprobó que la diferencia de temperatura observada en el níquel a las velocidades mínima y máxima de calentamiento utilizadas en este estudio es inferior a  $1^{\circ}\text{C}$ .

### 5.3.3. Modelización de la curva TG del PS-carbono

Previamente a efectuar cualquier ajuste de los datos, se aplicó el método de Flynn y Wall para comprobar si el principal proceso de degradación es un único paso de descomposición o no, suponiendo una dependencia de Arrhenius entre la velocidad de la reacción y la temperatura en el caso de sólo PS. De acuerdo con esto, la energía de activación,  $E$ , y cualquier valor particular de  $\alpha$  se determina por la ecuación 5.1:

$$\ln \beta = cte - \frac{E}{RT} \quad (5.1)$$

donde  $R$  es la constante de los gases y  $\beta$  la velocidad de calentamiento.

Así, en la constante de conversión, el gráfico de  $\ln \beta$  frente a  $\frac{1}{T}$ , obtenido de una serie de experimentos realizados a varias velocidades de calentamiento, debería ser una línea recta, cuya pendiente permite la evaluación de la energía de activación.

La Figura 5.4 muestra que la energía de activación no es constante con la conversión, lo que sugiere que el proceso no puede ser adecuadamente repro-

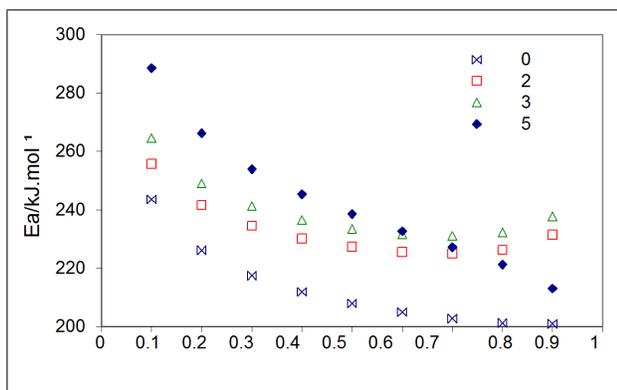


Figura 5.4: Relación entre la energía de activación y la conversión obtenida por el método de Flynn y Wall a partir de muestras PS que contienen diferentes cantidades de MWCNT.

ducido por una única componente de un modelo basado en el de Arrhenius. Por este motivo, se ha propuesto utilizar este nuevo modelo de orden de reacción basado cuya dependencia de la temperatura no sigue la ley de Arrhenius, con los datos obtenidos en la TG.

Debe destacarse que el modelo propuesto es más flexible que el clásico de orden de reacción basado estrictamente en el de Arrhenius (López-Beceiro et al. 2013). Además, los datos TG obtenidos en este trabajo fueron contrastados con éxito con datos de calorimetría diferencial de barrido (DSC) correspondientes a la transición a baja temperatura de una red organometálica (MOF). A fin de adaptar la expresión adecuada a los datos de TG, se hacen algunas consideraciones:

- (1). A diferencia de DSC, donde se espera un cambio continuo de la capacidad calorífica,  $C_p$ , con la temperatura, incluso cuando no hay ninguna transformación, en TG no se espera ninguna pérdida de masa más que la asociada a las transformaciones. Además, no es necesaria la substracción de línea base, asumiendo que la deriva experimental de la línea base es muy pequeña comparada con la pérdida de masa de las muestras.
- (2). Una función logística generalizada puede ajustar los datos, mientras que la primera derivada de esa función debería ajustar los datos DTG. En este trabajo los ajustes se realizan en la curva TG porque éstos se obtienen directamente, mientras que los de DTG se estiman desde los de TG. Los datos TG se consideran más robustos que los DTG.
- (3). A partir de los diagramas de las Figuras 5.10 y 5.5, es claro que, al menos en las muestras que contienen MWCNT, subyacen dos procesos de degradación, mientras que en la muestra de PS hay sólo una degradación.

Los dos procesos estarán representados por una mezcla de dos funciones logísticas generalizadas.

- (4). Hay un residuo al final del experimento, que corresponde al contenido de nanotubos, ya que la curva TG del PS no deja ningún residuo de degradación a 500°C y los nanotubos son estables incluso a mayores temperaturas, como puede verse en la Figura 5.5. Por esta razón, ese residuo será representado por una constante.

Finalmente, el modelo puede ser escrito como:

$$y_{total}(T) = y_1(T) + y_2(T) + Cte \quad (5.2)$$

donde  $y_i(T)$  son las funciones logísticas generalizadas de la forma

$$y_i(T) = \frac{c_i}{(1 + \tau_i \exp(-b_i(T - m_i)))^{\frac{1}{\tau_i}}} \quad (5.3)$$

donde  $c_i$  representa el porcentaje de muestra involucrada en cada etapa de degradación,  $m_i$  la temperatura en la máxima razón de cambio,  $b_i$  está relacionado con la velocidad de cambio,  $\tau_i$  es la medida para la asimetría y es  $T$  la temperatura. Entonces, aunque  $m_i$ ,  $b_i$  y  $\tau_i$  son parámetros de ajuste, en ausencia de procesos con gran solapamiento,  $m_i$  puede ser fácilmente identificado como la temperatura en el pico de la derivada (DTG). Por otra parte,  $\tau_i$  está relacionado con un orden de reacción aparente,  $n$ , por  $n = 1 + \tau$ .

Todas las curvas TG se ajustaron al modelo según el método descrito en trabajos previos (López-Beceiro et al., 2012).

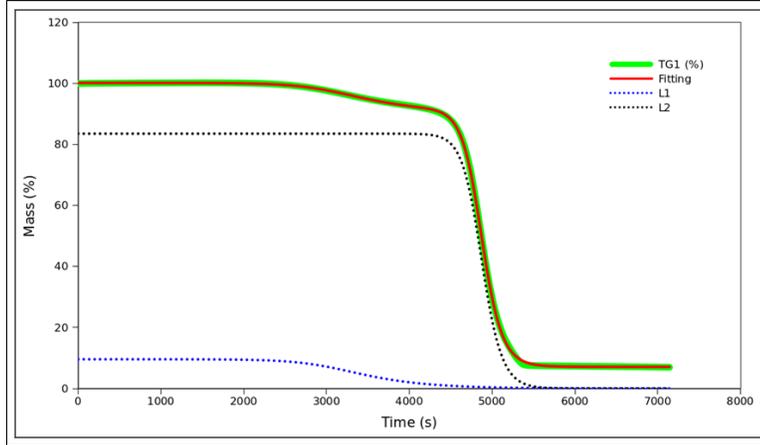


Figura 5.5: Ajuste de la curva de TG y las dos componentes logísticas,  $L_1$  y  $L_2$  obtenidas de la muestra con 5% de nanotubos a  $5\text{ }^\circ\text{C min}^{-1}$ .

La figura 5.5 muestra el ajuste de una curva de TG y las dos componentes logísticas. En este trabajo, la atención se centra en la etapa de degradación principal, que aparece a mayor temperatura.

Si se analizan estos nanocompuestos a tres velocidades de calentamiento, se comprueba que el añadir nanotubos provoca una pequeña pérdida de masa (en torno al 14 %) que se inicia a unos 230 °C que no aparecía en el PS. Sin embargo, el proceso principal de degradación aparece desplazado a temperaturas más altas y parece que esa estabilización es un poco menor al aumentar la proporción de nanotubos (véanse las Figuras 5.6, 5.7 y 5.8).

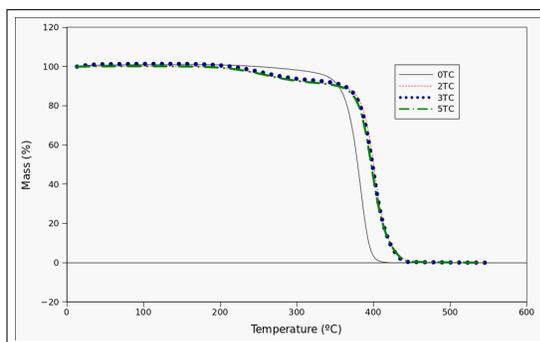


Figura 5.6: Ajuste de la muestra de PS-nanotubos con una velocidad de calentamiento de 2 °C/min.

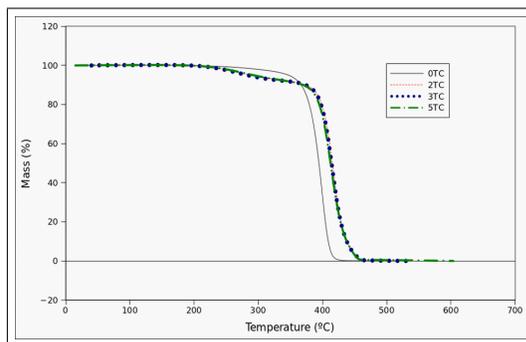


Figura 5.7: Ajuste de la muestra de PS-nanotubos con una velocidad de calentamiento de 5 °C/min.

Según el modelo,  $b_i$  es inversamente proporcional a  $m_i - T_c$ , donde  $T_c$  representa la temperatura crítica. Así, las gráficas del término exponencial de las curvas obtenidas a diferentes velocidades de calentamiento frente a la temperatura deberían cortarse aproximadamente en  $T_c$ .

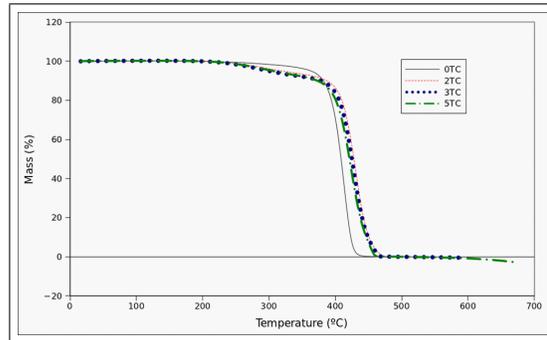


Figura 5.8: Ajuste de la muestra de PS-nanotubos con una velocidad de calentamiento de  $10\text{ }^{\circ}\text{C}/\text{min}$ .

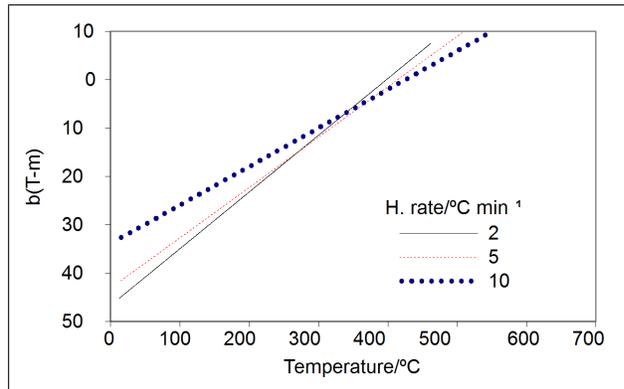


Figura 5.9: Relación entre el término exponencial y la temperatura obtenidos de los datos de nanotubos al 5% con la velocidad de calentamiento indicado.

La Figura 5.9 muestra que en este caso las curvas no se cortan en el mismo punto. Sin embargo, esto no es razón suficiente para rechazar el modelo, ya que tales desviaciones pueden provenir de la existencia de procesos poco solapados. Entonces, ya que en este caso no es posible obtener  $T_c$  como un simple punto de corte hacemos otro enfoque de nuestro trabajo, provisionalmente usado aquí para muestras con el mismo contenido de carga: la distancia de la temperatura a la velocidad máxima de cambio,  $m$ , desde  $T_c$ , debe seguir una tendencia con la velocidad de calentamiento.

Es importante destacar que el parámetro  $m$ , que representa la temperatura a la que se observa mayor pérdida de masa, probablemente es más independiente de un posible proceso solapado que el término exponencial,  $-b(T-m)$ . La razón es que  $m$  viene del pico de la DTG, situado en el eje de la temperatura que podría no estar realmente afectado por la existencia de procesos poco solapados. Pero  $b$

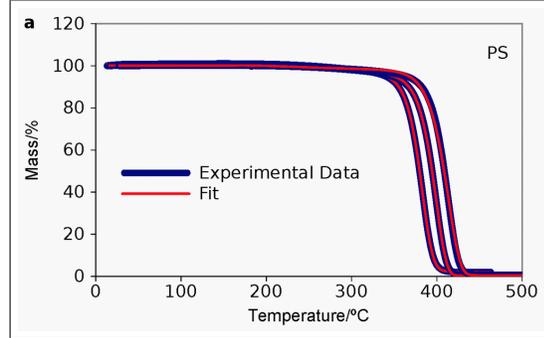


Figura 5.10: Gráficas de los datos originales y estimados de la muestra de PS (sin nanotubos) obtenidos a 2, 5 y 10  $^{\circ}\text{Cmin}^{-1}$ .

está relacionado con la velocidad total del proceso y entonces es muy dependiente de dichos procesos.

Dado que los valores de  $m$  son ya conocidos,  $T_c$  se puede calcular fácilmente por el ajuste de  $(m - T_c)$  frente a velocidad de calentamiento para una ley de potencia (Newman, 2005). Una vez obtenidos los valores de  $T_c$  para las series de muestras con el mismo contenido de refuerzo, se estiman nuevos valores de  $b$  mediante un proceso iterativo, haciendo uso de la expresión:

$$b = \frac{E_b}{R \cdot T_c (T_c - m)} \quad (5.4)$$

donde  $R$  es la constante de los gases y  $E_b$  representa la energía de barrera del proceso. Debería ocurrir que la gráfica de  $b$  en función de  $\frac{1}{R \cdot T_c (T_c - m)}$  fuese lineal.

Sin embargo, dado que los valores inicialmente estimados de  $b$  resultaron con cierta no linealidad, se aplicó iterativamente el siguiente procedimiento:  $E_b$  se estima con los valores de  $b$  inicialmente calculados minimizando el error con respecto a la ecuación 5.3. Luego, se estiman nuevos valores de  $b$  y se ajustan de nuevo las curvas, manteniendo los nuevos valores de  $b$  constantes. De esta manera, se obtienen unos valores ligeramente diferentes de  $T_c$  y  $E_b$  y el proceso se repite hasta que no se observe ningún cambio apreciable en las nuevas estimaciones.

Tabla 5.1. Valores de  $E_b$  y  $T_c$  obtenidos del ajuste de las curvas TG de muestras con diferentes contenidos de nanotubos de carbono.

Contenidos de MWCNT (%)	$E_b / \text{Jmol}^{-1}$	$T_c / ^{\circ}\text{C}$
0	41338	294
2	54626	315
3	48842	312
5	48087	310

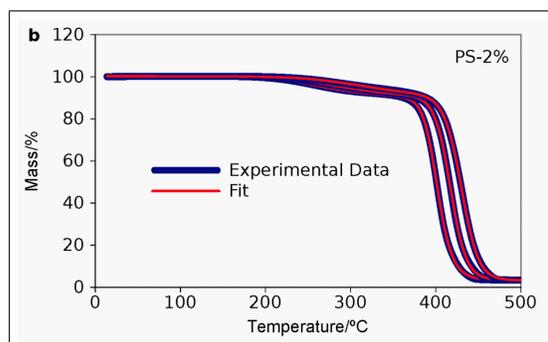


Figura 5.11: Gráficas de los datos originales y estimados de la muestra de PS con un 2% de nanotubos de carbono obtenidos a 2, 5 y 10 °Cmin<sup>-1</sup>.

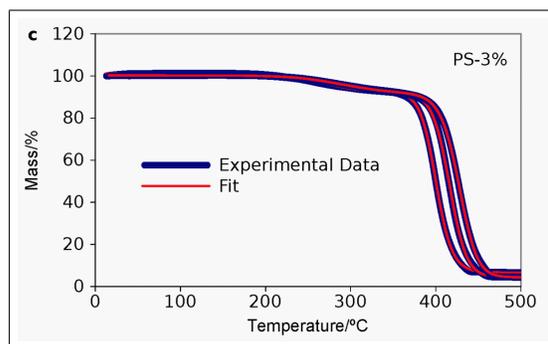


Figura 5.12: Gráficas de los datos originales y estimados de la muestra de PS con un 3% de nanotubos de carbono MWCNT, obtenidos a 2, 5, y 10 °Cmin<sup>-1</sup>.

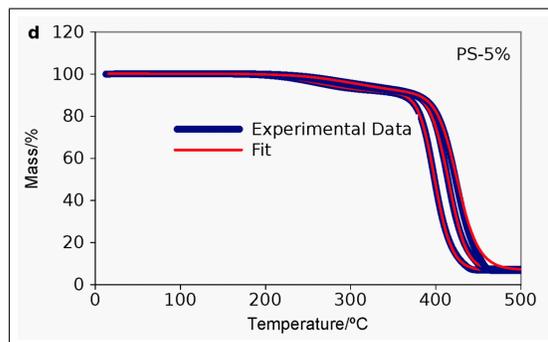


Figura 5.13: Gráficas de los datos originales y estimados de la muestra de PS con un 5% de nanotubos de carbono MWCNT, obtenidos a 2, 5, y 10 °Cmin<sup>-1</sup>.

### 5.3.4. Resultados de la degradación del PS con nanotubos

Las curvas TG de las muestras del nanomaterial PS-nanotubos fueron normalizadas por sustracción de los residuos de nanotubos y reescalados al 100 %, para eliminar un efecto de estabilización “virtual” (Tarrío-Saavedra et al. 2008). A pesar de esto, se observan efectos claros de los nanotubos:

- (1) La adición de nanotubos induce un proceso de pérdida de masa a menor temperatura que el principal proceso de degradación del PS. Aunque no es visible a la misma escala, se observa también una muy pequeña pérdida de masa en la muestra sin carga en el mismo rango de baja temperatura. Este proceso no se observa en las curvas de nanotubos, como se ve en la Figura 5.5.
- (2) La adición de nanotubos produce un cambio importante en el principal proceso de degradación a mayores temperaturas (como puede verse en las Figuras 5.6, 5.7 y 5.8)
- (3) El efecto estabilizador es más notorio para las muestras del 2 %, y decrece sucesivamente para las de 3 y 5 %. Estos efectos se observan también para las demás razones de calor y están en línea con los valores de  $E_b$  y  $T_c$  presentados en la tabla 1, en que el efecto más significativo de los nanotubos aparece para las muestras al 2 % (Figuras 5.6, 5.7 y 5.8).
- (4) Los valores de la barrera energética aumentan con la adición del 2 % de nanotubos. Mayores incrementos en la concentración de nanotubos producen un ligero descenso de la energía de activación y un efecto similar se observa con  $T_c$ , la temperatura mínima a la que el proceso tiene lugar.
- (5) La pérdida de masa a baja temperatura fue descrita en otros trabajos usando atmósfera de helio (Slobodian et al, 2010) y aire (Antonucci et al, 2007).
- De acuerdo con la Figura 5.5, la cantidad de esta pérdida no parece incrementarse apreciablemente con la adición de nanotubos. Se especuló que esta pérdida de masa fuese causada probablemente por fracciones pequeñas de PS o restos de surfactantes no lavados (Slobodian et al, 2010).

La hipótesis de los surfactantes es descartada aquí porque no hubo adición de surfactantes. Sin embargo, el uso general del PS preparado mediante polimerización iniciada por radicales libres contiene enlaces cabeza-cabeza como consecuencia de la terminación polimérica por emparejamiento radical. La tensión térmica provoca la homólisis del vínculo relacionando las componentes cabeza-cabeza. Las cadenas de polímeros que contienen enlaces cabeza-cola no deberían ser propensas a esta fácil descomposición (Howell et al, 2003; Howell et al, 2009). Por otra parte, ¿por qué unas pequeñas fracciones de PS o defectos en la cadena con enlace cabeza-cabeza deberían comportarse de diferente forma debido a los nanotubos?

Los resultados sugieren que la adición de nanotubos promueve la segregación de dos fracciones diferentes, una sería más rica en defectos en los enlaces cabeza-cabeza y la otra en cadenas de polímeros que contienen menos enlaces cabeza-cabeza. Así, no importa cuál de las dos fracciones mezcla mejor con CNTs; la más rica en defectos debería experimentar la pérdida de masa a menor temperatura, y la otra fracción debería ser más estable térmicamente que la muestra original donde ambas fases están profundamente dispersas y operan en medio de ambas componentes segregadas.

Mientras que la adición de pequeñas cantidades de nanotubos es suficiente para provocar segregación de cadenas defectuosas y sin defectos, esa segregación no se incrementa más con adición de CNTs por encima del 2%. Por otra parte, el hecho de que el efecto de estabilización decrezca desde el 2 al 5% de CNTs sugiere que los CNTs están mezclados en la fase estable sin defectos, aumentando su superficie y facilitando la degradación térmica de esta fase.

Las Figuras 5.6, 5.7 y 5.8 muestran las gráficas de las curvas TG obtenidas a diferentes velocidades de calentamiento con los correspondientes ajustes. Las curvas obtenidas a mayores velocidades de calentamiento están desplazadas a mayores temperaturas, típico de la degradación de polímeros (Flynn y Wall, 1966). Puede observarse que la calidad de los ajustes es muy buena y la única desviación relevante se observa al final del proceso de degradación de la curva obtenida a  $10^{\circ}\text{C min}^{-1}$  de la muestra con un 5% de nanotubos. Esto significa que aunque el modelo generalmente explica el comportamiento de la degradación, parece aparecer un nuevo proceso de degradación a la mayor velocidad de calentamiento y con el mayor contenido de MWNT. Este nuevo proceso está tratado en la figura 5.14.

Hay una temperatura de referencia,  $T_c$ , a la que debería ocurrir el proceso de degradación principal a una velocidad muy pequeña. De acuerdo con la tabla 1,  $T_c$  aumenta con la adición de nanotubos, pero el efecto estabilizador disminuye cuando el contenido de nanotubos se aumenta. La barrera de energía del proceso sigue la misma tendencia, que también concuerda con el efecto estabilizante comentado anteriormente.

Además del análisis cinético, la Figura 5.14 puede ayudar a entender mejor el anómalo ajuste descrito en la Figura 5.12 y cómo el proceso de degradación principal cambia debido a la presencia de los nanotubos. Esta figura representa las curvas DTG obtenidas de las muestras con diferentes contenidos de refuerzo a las mismas velocidades de calentamiento ( $10^{\circ}\text{C min}^{-1}$ ).

El pico de la muestra sin refuerzo es más asimétrico, con una pendiente más pronunciada en la parte de mayor temperatura, como habitualmente ocurre en muchos termoplásticos, pareciendo procesos de evaporación (Suwardie et al, 2002). La mayor velocidad de degradación, representada por el valor del pico, es menor en los compuestos y el proceso entero está desplazado a mayores temperaturas como consecuencia de la presencia de nanotubos.

El hecho de que la velocidad de degradación decrezca con la presencia de nanotubos es una indicación de la interacción carga-matriz. Curiosamente, aunque la máxima velocidad de degradación siempre decrece cuando se incrementa el contenido de nanocarga, el mayor desplazamiento del pico se observa para el

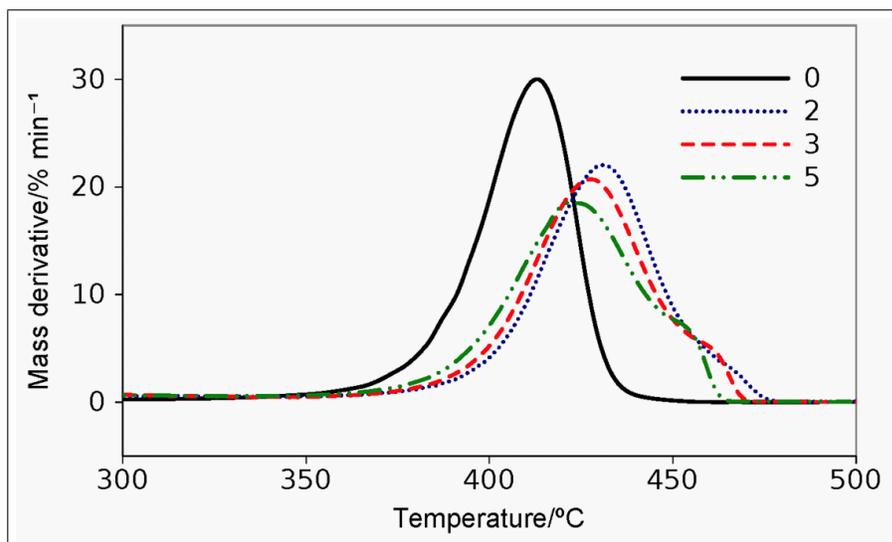


Figura 5.14: Relaciones entre las derivadas de la TGA, DTG, que se obtuvieron a  $10 \text{ }^\circ \text{C min}^{-1}$  a partir de muestras de PS que contienen diferentes cantidades de MWNT.

2% y decrece cuando el contenido en nanotubos se aumenta, de la misma forma que los parámetros cinéticos  $E_b$  y  $T_c$  presentados en la tabla 2.

Hay también un proceso de degradación adicional que se observa sólo en las muestras con refuerzo, que se revela en el *hombro* en el mayor rango de temperaturas del proceso de degradación principal. La temperatura a la que el *hombro* se observa decrece cuando se aumenta el contenido de refuerzo. En el caso de la muestra del 5%, este *hombro* aparece alrededor de los  $451^\circ\text{C}$ . La forma del *hombro* también cambia con el contenido de refuerzo, indicando una aceleración del proceso originado por los CNTs.

Este proceso explica la discrepancia mencionada antes del modelo con respecto a la curva del 5% y podría ser atribuido a una subfracción del polímero prácticamente libre de enlaces cabeza-cabeza. Esta subfracción es más estable térmicamente que la otra, pobre en defectos pero que interactúa igualmente con los CNTs. Como se ve en la Figura 5.5, el primer proceso de degradación fue adecuadamente separado para no interferir el análisis cinético del principal proceso de degradación. Por otra parte, el proceso adicional observado en el rango de las altas temperaturas es bastante pequeño comparado con el proceso principal y permanece prácticamente fuera del ajuste. Así, los valores de los parámetros obtenidos por el ajuste prácticamente representan la cinética del proceso principal.

Investigados los efectos de MWCNT en la degradación térmica del PS en una atmósfera inerte, los resultados sugieren un equilibrio entre efectos posi-

tivos y negativos. El proceso global se desdobra en dos pasos como consecuencia de la adición de nanotubos. El primer paso, que involucra una relativamente pequeña, pero significativa, cantidad de la fracción de PS se completa a menor temperatura que el proceso global de la muestra con sólo PS. El paso de mayor degradación es desplazado a mayores temperaturas. La energía de activación del proceso global del PS y el paso principal de degradación de los compuestos se evaluaron con el método de Flynn y Wall.

Se observa que la energía de activación varía con la conversión, lo que indica que el proceso no puede ser adecuadamente reproducido por un único paso basado en el modelo de Arrhenius. Se aplicó con éxito un modelo cinético construido para transformaciones sólidas para analizar el proceso de degradación principal y se observó un proceso adicional muy pequeño en el rango de las altas temperaturas del paso de degradación principal. Excepto por ese pequeño proceso adicional que cae fuera del ajuste, el principal paso de la degradación se representa por una sola función, con una sola barrera de energía y todas las conversiones para una muestra dada. Este proceso es claramente estabilizado por la presencia de nanotubos, aunque la estabilización es más pronunciada en los menores contenidos de refuerzo.

Los valores de la barrera de energía y la temperatura de referencia son mayores en la concentración del 2% de nanotubos. Mayores incrementos de contenido de nanotubos producen ligeros decrecimientos en estos valores. Esto sugiere que la adición de nanotubos provoca la segregación de cadenas defectuosas y libres de defectos. Este punto es el objeto de la investigación en curso. De acuerdo con la literatura, esas dos fracciones están presentes en el PS y se comportan de diferente manera.



## Capítulo 6

# Conclusiones y líneas futuras

### 6.1. Conclusiones y objetivos alcanzados

En este capítulo final se presenta un resumen de las conclusiones alcanzadas en los capítulos anteriores. Asimismo, se hará una propuesta de trabajos futuros de interés a la luz de estas conclusiones.

En esta tesis se estudió la aplicación de técnicas metaheurísticas de optimización para resolver un importante problema del Análisis Térmico, el modelado óptimo de curvas de degradación térmica. Para ello, y con el objetivo de abordarlo mediante optimización se utilizaron algoritmos evolutivos y algoritmos clásicos y se hace una nueva propuesta de un algoritmo memético.

Se realizó en primer lugar una revisión de la literatura para conocer los trabajos más recientes desde el punto de vista del Análisis Térmico y de la Optimización y las líneas de investigación en las que se está trabajando actualmente. Este estudio, que cumpliría el primer objetivo específico planteado, está incluido en los capítulos I, II y III, y se demostró que las herramientas y algoritmos existentes presentan importantes limitaciones desde el punto de vista de la modelización de los datos procedentes del análisis termogravimétrico de materiales.

Observando que la gran mayoría de las técnicas estaban diseñadas para abordar problemas de optimización de modelos simples, se seleccionaron algunos de estos algoritmos y se adaptaron al contexto de funciones complejas: la mezcla de logísticas y la de logísticas generalizadas. Una vez encontrados los mejores algoritmos, se diseñó una nueva técnica híbrida que permite aprovechar las características de los algoritmos evolutivos SA, CMAES y DE, al combinarlos con algoritmos clásicos como el algoritmo Nelder-Mead, cumpliendo así con otro de los objetivos de la tesis (objetivos específicos 2 y 3).

Con la propuesta de un nuevo algoritmo híbrido que recoge la bondad de los algoritmos evolutivos que no precisan de un valor inicial para el vector de

parámetros y se cumple con el primer objetivo fijado al inicio de este trabajo: definir, formular y abordar el problema de la optimización de curvas TG. Este trabajo se presenta con detalle en el capítulo IV de la tesis, en el que además de la propuesta del algoritmo se realiza un trabajo de simulación en diferentes escenarios, proponiendo situaciones complejas como los procesos solapados, a fin de valorar la bondad del método propuesto (objetivo 4 de esta tesis).

Otro aspecto importante a tener en cuenta es la capacidad que poseen las técnicas desarrolladas para encontrar soluciones de calidad en un tiempo razonable. Para estudiar esta propiedad, se analizó también el tiempo requerido por cada metaheurística para converger en los distintos escenarios simulados, es decir, para obtener los mejores resultados posibles. De nuevo, se comparó el funcionamiento de los algoritmos propuestos y se obtuvieron conclusiones que pueden facilitar a futuros investigadores la labor de elección del algoritmo más adecuado según sus necesidades, entre éstas estaría que el uso de algoritmos combinados en situaciones complejas es mejor alternativa que los métodos clásicos (objetivo 4 de esta tesis).

Finalmente, estos métodos propuestos se aplicaron a un caso real concreto: la modelización mediante un método de mezcla de dos logísticas generalizadas de la degradación térmica en atmósfera inerte de un nanocompuesto, el poliestireno con adición de nanotubos de carbono. Se comprobó cómo estas técnicas son válidas para la modelización de este nanocompuesto y que además permiten el estudio de diferentes propiedades térmicas del material (objetivo específico 5 de esta tesis). Este trabajo se recoge en el capítulo V de esta tesis y forma parte del artículo recientemente publicado en la revista *Journal of Thermal Analysis and Calorimetry* (Ríos-Fachal, et al, 2013), lo que ayudará a la difusión del trabajo desarrollado que era otro de los objetivos específicos de la tesis (objetivo 6).

## 6.2. Líneas futuras

Todo trabajo de investigación provoca que asomen nuevos problemas, y, en este caso se han encontrado posibles extensiones de algunos de los resultados expuestos en esta memoria, y que serán futuras líneas de investigación. Algunos se enumeran a continuación.

Como se destacó en las conclusiones, se han planteado adaptaciones con diferentes algoritmos que resuelven el problema de la modelización de datos térmicos. En este sentido, un trabajo futuro sería el de plantear otros algoritmos, diferentes de los aquí propuestos y ver si pueden igualar o mejorar el algoritmo memético presentado.

En este trabajo se han adaptado algoritmos evolutivos directamente orientados a la resolución de un problema concreto, el modelado de curvas TG. Dado que estos nuevos algoritmos evolutivos han conseguido buenos resultados, además de en este problema de ajuste de curvas, podría trasladarse esta idea a otras situaciones similares pertenecientes a otro tipo de curvas como, por ejemplo, las curvas de calorimetría diferencial de barrido (DSC) o las de análisis mecano-dinámico (DMTA); sería interesante proponer formalmente estos algo-

ritmos con otros modelos, para este tipo de datos.

Igualmente interesante sería diseñar nuevos algoritmos o adaptaciones de los aquí presentados basados en algoritmos de distinta naturaleza. Hemos estado trabajando con algoritmos evolutivos, sin embargo, el funcionamiento interno de todos ellos se apoya en los mismos fundamentos, los conceptos de dominancia, crowding o densidad. Otra alternativa sería aplicar descomposición (véase Zhang y Li, 2007), un concepto que está empezando a cobrar gran importancia en la literatura.

En lo que concierne a la formulación del algoritmo memético presentado, puede ser una buena idea analizar la posibilidad de usar *tomadores de decisión* (decision making) para tratar de dirigir mejor la búsqueda de soluciones y mejorar la calidad de las mismas en contextos más complejos como pueden ser los de la modelización de las derivadas sucesivas de las curvas TG. Con el mismo propósito, se podría también estudiar la incorporación de nuevas funciones para los modelos propuestos como la mezcla de funciones tipo Weibull o la utilización de nuevas estrategias de ajuste paramétrico auto-adaptativas para los requerimientos reales del problema (Said, 2010).

Finalmente, otra interesante propuesta de trabajo futuro está relacionado con la implementación en una librería de R del algoritmo planteado, de forma que a partir de los datos obtenidos experimentalmente pueda hacerse automáticamente el ajuste óptimo tanto de las curvas TG como de sus derivadas. De este modo podría dotarse a la comunidad científica de una herramienta de Software libre, como las librerías de R, específica para la modelización de datos de degradación que podría también ampliarse con otros modelos cinéticos disponibles en la librería.



# Capítulo 7

## Anexos

### 7.1. Anexo I: Cálculos para el modelo mezcla de logísticas

En este anexo se analizan algunas propiedades matemáticas del modelo de mezcla de logísticas que se ha propuesto para modelizar las curvas TG (pérdidas de peso de un material),  $m_\theta(t)$ , frente al tiempo  $t$ :

$$m_\theta(t) = \sum_{i=1}^k w_i f(a_i + b_i t) \text{ con } f(t) = \frac{\exp t}{1 + \exp t}, \quad (7.1)$$

donde  $w_i, a_i$  y  $b_i$  son los parámetros de este modelo, para los  $k$  escalones que presente la función  $m_\theta(t)$ .

#### 7.1.1. Propiedades de las derivadas

Algunos de los métodos de optimización requieren ciertas propiedades del modelo como que las derivadas de la función a optimizar, con respecto a los parámetros, sean continuas o diferenciables:

Para las derivadas primera y segunda de la función  $m$  con respecto a  $t$  y a los parámetros  $(w_i, a_i, b_i)$  se obtienen las expresiones siguientes:

$$m'(x) = \sum_{i=1}^k w_i b_i f'(a_i + b_i x) \quad (7.2)$$

$$m''(x) = \sum_{i=1}^k w_i b_i^2 f''(a_i + b_i x) \quad (7.3)$$

$$Y(t) = \sum_{i=1}^k w_i f(a_i + b_i t) \text{ con } f(t) = \frac{\exp t}{1 + \exp t}, \quad (7.4)$$

$$\begin{aligned}
f'(t) &= \frac{e^t}{e^t + 1} - \frac{e^{2t}}{(e^t + 1)^2} \\
f''(t) &= \frac{e^t}{e^t + 1} - 2 \frac{e^{2t}}{(e^t + 1)^2} - \frac{e^{2t}}{(e^t + 1)^2} + 2e^t \frac{e^{2t}}{(e^t + 1)^3}
\end{aligned} \tag{7.5}$$

Derivadas de la función con respecto a los parámetros del modelo, para comprobar que  $\left(\frac{\partial}{\partial \theta_i} m_\theta(t)\right)^2$  es uniformemente continua.

$$m_\theta(t) = \sum_{i=1}^k w_i \frac{\exp(a_i + b_i t)}{1 + \exp(a_i + b_i t)} = \sum_{i=1}^k y_i(t) \tag{7.6}$$

Será suficiente comprobar la continuidad uniforme para cada una de las derivadas parciales de las funciones  $y_i(t)$ :

Para ello, sea  $y(w, a, b) = w \frac{\exp(a+bt)}{1+\exp(a+bt)}$

Derivadas con respecto a  $w, a$  y  $b$  de  $y(w, a, b)$ :

$$\frac{\partial}{\partial w} m_\theta(t) = \frac{\exp(a + bt)}{1 + \exp(a + bt)}$$

$$\frac{\partial}{\partial a} m_\theta(t) = w \frac{\exp(a + bt)}{\exp(a + bt) + 1} - w \frac{\exp^2(a + bt)}{(\exp(a + bt) + 1)^2}$$

$$\begin{aligned}
\frac{\partial}{\partial b} m_\theta(t) &= w \exp(a + bt) \frac{t}{\exp(a + bt) + 1} - \\
&\quad - w \exp^2(a + bt) \frac{t}{(\exp(a + bt) + 1)^2}
\end{aligned}$$

Como la derivada de  $m_\theta(t)$  es suma de  $k$  expresiones similares a las anteriores, se deduce la propiedad de continuidad uniforme de la hipótesis de los modelos de optimización clásicos, ya que es composición de funciones con esta propiedad.

### Hipótesis necesarias sobre la función de mínimos cuadrados no lineales.

Cálculo de la matriz hessiana

$$\psi_n(\theta) = \sum_{j=1}^n (Y_j - m_\theta(t_j))^2$$

donde, para el caso de 4 logísticas

$$\theta = (w_1, a_1, b_1, w_2, a_2, b_2, w_3, a_3, b_3, w_4, a_4, b_4)$$

La matriz  $H\psi_n(\theta, \vec{t}, \vec{Y}) = \left(\frac{\partial^2 \psi_n}{\partial \theta_i \partial \theta_j}\right)_{i,j}$  tiene inversa con  $H_{i,l} = \frac{\partial^2 \psi_n}{\partial \theta_i \partial \theta_l}$

$$\frac{\partial}{\partial \theta_i} \left( \sum_{j=1}^n (Y_j - m_\theta(t_j))^2 \right) = \sum_{j=1}^n 2(Y_j - m_\theta(t_j)) \left( -\frac{\partial}{\partial \theta_i} m_\theta(t_j) \right)$$

$$H_{i,l} = \frac{\partial^2 \psi_n}{\partial \theta_i \partial \theta_l} = \begin{cases} H_{i,i} & \text{si } i = l \\ H_{i,l} & \text{si } i \neq l \end{cases}$$

$$H_{i,i} = 2 \sum_{j=1}^n \left[ \left( \frac{\partial}{\partial \theta_i} m_\theta(t_j) \right)^2 - (Y_j - m_{\theta_0}(t_j)) \frac{\partial^2}{\partial \theta_i^2} m_\theta(t_j) \right] \quad (7.7)$$

$$H_{i,l} = 2 \sum_{j=1}^n \left[ \frac{\partial}{\partial \theta_i} m_\theta(t_j) \frac{\partial}{\partial \theta_l} m_\theta(t_j) - (Y_j - m_{\theta_0}(t_j)) \frac{\partial^2}{\partial \theta_i \partial \theta_l} m_\theta(t_j) \right]$$

En el caso concreto de mezcla de logísticas se tendrá:

$$m_\theta(t) = \sum_{i=1}^k w_i \frac{\exp(a_i + b_i t)}{1 + \exp(a_i + b_i t)}$$

con lo que  $\psi_n(\theta)$  estará compuesta por elementos del tipo:

$$\psi_n(\theta) = \sum_{j=1}^n (Y_j - m_\theta(t_j))^2 = \sum_{j=1}^n \left[ Y_j - \sum_{i=1}^k w_i \frac{\exp(a_i + b_i t_j)}{1 + \exp(a_i + b_i t_j)} \right]^2$$

Bastará con analizar las propiedades de las funciones del tipo

$$\zeta(w, a, b) = \left[ y - w \frac{\exp(a + bt)}{1 + \exp(a + bt)} \right]^2$$

$$\frac{\partial \zeta(w, a, b)}{\partial w} = \frac{\partial \left[ y - w \frac{\exp(a + bt)}{1 + \exp(a + bt)} \right]^2}{\partial w} =$$

$$= \frac{1}{2e^{a+bt} + e^{2a+2bt} + 1} (2we^{2a+2bt} - 2ye^{a+bt} - 2ye^{2a+2bt})$$

$$\frac{\partial \zeta(w, a, b)}{\partial a} = \frac{\partial \left[ y - w \frac{\exp(a + bt)}{1 + \exp(a + bt)} \right]^2}{\partial a} =$$

$$= \frac{1}{3e^{a+bt} + 3e^{2a+2bt} + e^{3a+3bt} + 1} (2w^2 e^{2a+2bt} - 2wye^{2a+2bt} - 2wye^{a+bt})$$

$$\frac{\partial \zeta(w, a, b)}{\partial b} = \frac{\partial \left[ y - w \frac{\exp(a + bt)}{1 + \exp(a + bt)} \right]^2}{\partial b} =$$

$$= \frac{1}{3e^{a+bt} + 3e^{2a+2bt} + e^{3a+3bt} + 1} 2tw^2 e^{2a+2bt} - 2twye^{2a+2bt} - 2twye^{a+bt}$$

Matriz hessiana H

Diagonal de H;  $H_{i,i} =$

$$2 \sum_{j=1}^n \left[ \left( \frac{\partial}{\partial w_i} \sum_{i=1}^k w_i f(a_i + b_i t_j) \right)^2 - (Y_j - m_{\theta_0}(t_j)) \frac{\partial^2}{\partial w_i^2} \sum_{i=1}^k w_i f(a_i + b_i t_j) \right]$$

$$\left( \frac{\partial}{\partial w} \left( w \frac{\exp(a_i + b_i t_j)}{1 + \exp(a_i + b_i t_j)} \right) \right)^2 = \frac{e^{2(a_i + b_i t_j)}}{(e^{a_i + b_i t_j} + 1)^2}$$

$$\frac{\partial^2}{\partial w^2} \left( w \frac{\exp(a_i + b_i t_j)}{1 + \exp(a_i + b_i t_j)} \right) = 0$$

$$\left( \frac{\partial}{\partial a} \left( w \frac{\exp(a + b t_j)}{1 + \exp(a + b t_j)} \right) \right)^2 = w^2 \frac{e^{2(a + b t_j)}}{(2e^{a + b t_j} + e^{2a + 2b t_j} + 1)^2}$$

$$\frac{\partial^2}{\partial a^2} \left( w \frac{\exp(a + b t_j)}{1 + \exp(a + b t_j)} \right) = \frac{we^{a + b t_j} - we^{2a + 2b t_j}}{3e^{a + b t_j} + 3e^{2a + 2b t_j} + e^{3a + 3b t_j} + 1}$$

$$\left( \frac{\partial}{\partial b} \left( w \frac{\exp(a + b t_j)}{1 + \exp(a + b t_j)} \right) \right)^2 = w^2 t_j^2 \frac{e^{2(a + b t_j)}}{(2e^{a + b t_j} + e^{2a + 2b t_j} + 1)^2}$$

$$\frac{\partial^2}{\partial b^2} \left( w \frac{\exp(a + b t_j)}{1 + \exp(a + b t_j)} \right) = \frac{wt_j^2 e^{a + b t_j} - wt_j^2 e^{2a + 2b t_j}}{3e^{a + b t_j} + 3e^{2a + 2b t_j} + e^{3a + 3b t_j} + 1}$$

Elementos fuera de la diagonal de H

$$H_{i,l} = 2 \sum_{j=1}^n \left[ \frac{\partial}{\partial \theta_i} m_{\theta}(t_j) \frac{\partial}{\partial \theta_l} m_{\theta}(t_j) - (Y_j - m_{\theta_0}(t_j)) \frac{\partial^2}{\partial \theta_i \partial \theta_l} m_{\theta}(t_j) \right]$$

Veamos cada combinación por separado:

$$\frac{\partial}{\partial w} \sum_{i=1}^k w_i \frac{\exp(a_i + b_i t_j)}{1 + \exp(a_i + b_i t_j)} \frac{\partial}{\partial a} \sum_{i=1}^k w_i \frac{\exp(a_i + b_i t_j)}{1 + \exp(a_i + b_i t_j)}$$

$$- (Y_j - m_{\theta_0}(t_j)) \frac{\partial^2}{\partial w \partial a} \sum_{i=1}^k w_i \frac{\exp(a_i + b_i t_j)}{1 + \exp(a_i + b_i t_j)}$$

$$\frac{\partial}{\partial w} \left( w \frac{\exp(a + b t_j)}{1 + \exp(a + b t_j)} \right) \frac{\partial}{\partial a} \left( w \frac{\exp(a + b t_j)}{1 + \exp(a + b t_j)} \right) =$$

$$= 2w \frac{e^{2a + 2b t_j}}{3e^{a + b t_j} + 3e^{2a + 2b t_j} + e^{a + b t_j} e^{2a + 2b t_j} + 1}$$

$$\frac{\partial^2}{\partial w \partial a} \left( w \frac{\exp(a + bt_j)}{1 + \exp(a + bt_j)} \right) = \frac{e^{a+bt_j}}{2e^{a+bt_j} + e^{2a+2bt_j} + 1}$$

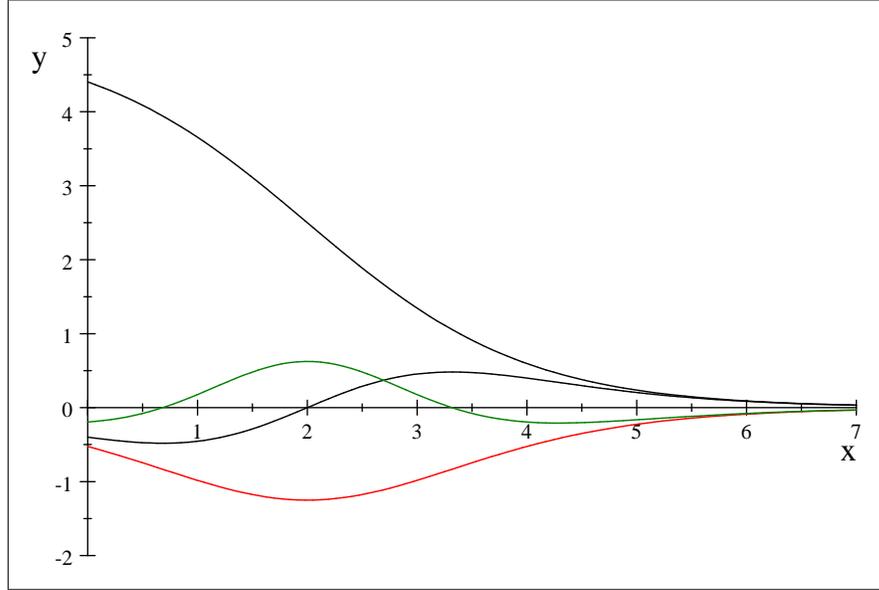
$$\begin{aligned} \frac{\partial}{\partial w} \left( w \frac{\exp(a + bt_j)}{1 + \exp(a + bt_j)} \right) \frac{\partial}{\partial b} \left( w \frac{\exp(a + bt_j)}{1 + \exp(a + bt_j)} \right) &= \\ = 2wt_j \frac{e^{2a+2bt_j}}{3e^{a+bt_j} + 3e^{2a+2bt_j} + e^{a+bt_j}e^{2a+2bt_j} + 1} \end{aligned}$$

$$\frac{\partial^2}{\partial w \partial b} \left( w \frac{\exp(a + bt_j)}{1 + \exp(a + bt_j)} \right) = t_j \frac{e^{a+bt_j}}{2e^{a+bt_j} + e^{2a+2bt_j} + 1}$$

$$\frac{\partial^2}{\partial a \partial b} \left( w \frac{\exp(a + bt_j)}{1 + \exp(a + bt_j)} \right) = \frac{wt_j e^{a+bt_j} - wt_j e^{2a+2bt_j}}{3e^{a+bt_j} + 3e^{2a+2bt_j} + e^{3a+3bt_j} + 1}$$

Cálculo de las derivadas primera, segunda y tercera para un caso concreto:

$$\begin{aligned} f(x) &= 5 \frac{\exp(-x + 2)}{1 + \exp(-x + 2)} \\ f'(x) &= 5 \frac{e^{2(2-x)}}{(e^{2-x} + 1)^2} - 5 \frac{e^{2-x}}{e^{2-x} + 1} \\ f''(x) &= 5 \frac{e^{2-x}}{e^{2-x} + 1} - 10 \frac{e^{4-2x}}{(e^{2-x} + 1)^2} - 5 \frac{e^{2(2-x)}}{(e^{2-x} + 1)^2} + 10e^{2-x} \frac{e^{4-2x}}{(e^{2-x} + 1)^3} \\ f'''(x) &= 30 \frac{e^{4-2x}}{(e^{2-x} + 1)^2} - 5 \frac{e^{2-x}}{e^{2-x} + 1} + 5 \frac{e^{2(2-x)}}{(e^{2-x} + 1)^2} - 60e^{2-x} \frac{e^{4-2x}}{(e^{2-x} + 1)^3} \\ &\quad + 30e^{2(2-x)} \frac{e^{4-2x}}{(e^{2-x} + 1)^4} \end{aligned}$$



Función logística (negro), primera derivada (rojo), segunda (negro) y tercera (verde)

## 7.2. Anexo II: Propiedades relativas a la Epistasis

Existen muchos factores que pueden hacer difícil la optimización de una función mediante Algoritmos Evolutivos. Los resultados experimentales atestiguan que uno de ellos es la relación entre los genes en la representación de la función a optimizar, fenómeno que se conoce como epistasis.

En 1991 se formaliza este concepto para el alfabeto binario y en 1995 se define la epistasis global de una función. En esta sección se analiza cómo se calcula la epistasis en algún caso particular.

### 7.2.1. Definición y cálculo de la *epistasis*

Supongamos que  $\Omega = \Omega_\ell = \{0, 1\}^\ell$  y que se consideran los subconjuntos:

$$\begin{aligned}\Omega_i(a) &= \{t \in \Omega; t_i = a\} \\ \Omega_{i,k}(a,b) &= \{t \in \Omega; t_i = a, t_k = b\}.\end{aligned}$$

Los valores promedio se definen como:

$$f_i(a) = \frac{1}{2^{\ell-1}} \sum_{t \in \Omega_i(a)} f(t).$$

$$f_{i,k}(a, b) = \frac{1}{2^{\ell-2}} \sum_{t \in \Omega_{i,k}(a,b)} f(t).$$

Para un alelo  $i \in \{0, \dots, \ell - 1\}$  se considera su valor de exceso:

$$E_i(a) = f_i(a) - \bar{f},$$

y, para un par de alelos  $(i, k)$  se define el valor de exceso del par como:

$$E_{i,k}(a, b) = f_{i,k}(a, b) - \bar{f}.$$

Para  $s \in \Omega_\ell$  el valor de exceso génico del par  $(i, k)$  se define:

$$\tilde{E}(s) = \sum_{0 \leq i < k < \ell} E_{i,k}(s_i, s_k) - (\ell - 2) \sum_{0 \leq i < \ell} E_i(s_i).$$

Por tanto, el valor génico previsto es  $\tilde{f}(s) = \tilde{E}(s) + \bar{f}$ .

La *epistasia* de la cadena  $s \in \Omega$  se define como:  $\varepsilon(s) = f(s) - \tilde{f}(s)$ .

y se puede expresar como:

$$\begin{aligned} \varepsilon(s) &= f(s) - \tilde{f}(s) \\ &= f(s) - \left[ \tilde{E}(s) + \bar{f} \right] \\ &= f(s) - \left[ \sum_{0 \leq i < k < \ell} E_{i,k}(s_i, s_k) - (\ell - 2) \sum_{0 \leq i < \ell} E_i(s_i) + \bar{f} \right] \end{aligned}$$

### 7.2.2. Cálculo de la epistasia para funciones de segundo orden

En este caso se considera una función  $f$  de segundo orden, es decir, una función que se puede escribir en la forma:

$$f(s) = \sum_{0 \leq i < \ell} g_i(s) + \sum_{0 \leq i < j < \ell} g_{ij}(s)$$

donde  $g_i(s)$  depende sólo del alelo  $s_i$ , y  $g_{ij}(s)$  sólo lo hace de  $s_i$  y  $s_j$ .

### 7.2.3. Cálculo de valores extremos de la epistasia

El problema de la búsqueda del mínimo está resuelto, si tenemos en cuenta el valor de la *epistasia* en términos de los coeficientes de Walsh para vectores unitarios.

$$\varepsilon(f) = 1 - \frac{\omega_0^2 + \sum_{0 \leq i < \ell} \omega_{2^i}^2 + \sum_{0 \leq i < j < \ell} \omega_{2^i + 2^j}^2}{\sum_{0 \leq i < 2^\ell} \omega_i^2}$$

## 7.3. Anexo III: Implementación de sentencias en R

### 7.3.1. Introducción al programa R

El objetivo de este anexo es mostrar los programas utilizados tanto para la simulación de las curvas de degradación usadas en el capítulo IV de la tesis como para los distintos ajustes con los algoritmos evolutivos implementados con las librerías del R.

R, también conocido como “GNU S”, es un lenguaje para el cálculo estadístico y la generación de gráficos. R fue desarrollado inicialmente por Gentleman y Ross Ihaka del departamento de estadística de la Universidad de Auckland. A partir de 1997 la lista de colaboradores de este software libre ha ido creciendo con aportaciones en todos los campos de la estadística.

Puede decirse que R es un dialecto (o clon) del lenguaje S, que fue desarrollado en los Laboratorios Bell (AT&T) por John Chambers et al. (1994). Su nombre está vinculado al programa C, adaptado al área de la estadística. Entre las características del programa R están:

1. La capacidad de combinar análisis *pre-empaquetados* con análisis *ad-hoc*, específicos para una situación determinada.
2. Posibilidad de crear gráficos de alta calidad.
3. La comunidad de R es muy dinámica y está integrada por estadísticos de gran renombre.
4. Hay extensiones específicas a áreas nuevas (bioinformática, geoestadística, estadística espacial, ...)
5. Es un lenguaje orientado a objetos.
6. Es similar a Matlab y con sintaxis que recuerda a C/C++.
7. Aunque depende del sistema operativo, tiene la ventaja de adaptarse a distintos sistemas (Linux, Windows, MacOS).

Este programa ya se puede considerar como la *lingua franca* del análisis de datos. Por lo tanto, tratar de usar el R como herramienta también para la modelización y optimización en Análisis Térmico es algo que está totalmente justificado.

Una de las grandes ventajas de este programa es que se trata de Software libre y ofrece la posibilidad de instalación de paquetes especializados de forma gratuita. El programa instalado por defecto es el denominado programa *base*, y dependiendo de los datos y del estudio a realizar, podemos instalar cualquier otro programa, directamente, desde el menú de *Packages*. Es posible instalar otros paquetes más específicos a través de la Web del programa o desde el mismo menú, siempre que tengamos nuestro equipo conectado a la red y entre ellos se encuentran la mayoría de los algoritmos de optimización existentes.

Tanto el programa como distintos manuales se puede encontrar en la siguiente dirección Web <http://cran.es.R-project.org/> CRAN (Comprehensive R Arquivos Network).

### 7.3.2. Simulación de un modelo logístico generalizado con R

Simulación de una muestra de datos que reflejen un comportamiento de un experimento de análisis termogravimétrico.

Generación de 1000 datos en el intervalos (0,60) de forma equiespaciada

```
x<-seq(0,60, length.out=1000)
```

### se crean las funciones logísticas generalizadas con 3 componentes en este ejemplo (l1, l2 y l3)

```
f=function(x) c*(1-1/(1+t*exp(-b*(x-m))))^(1/t)
```

```
l1=function(x) 7*(1-1/(1+1*exp(-2*(x-20))))^(1/1)
```

```
l2=function(x) 0.5*(1-1/(1+1*exp(-1*(x-48))))^(1/1)
```

```
l3=function(x) 3*(1-1/(1+1*exp(-1.8*(x-21))))^(1/1)
```

```
logistica=function(x) l1(x)+l2(x)+l3(x)
```

```
win.graph()
```

```
plot(x,logistica(x))
```

```
lines(x, l1(x), col="red")
```

```
lines(x, l2(x), col="blue")
```

```
lines(x, l3(x), col="green")
```

```
legend("topright",c("Datos", "L1", "L2", "L3"),
```

```
col=c("black", "red", "blue", "green"), lwd=c(2,2,2,2), bty="n")
```

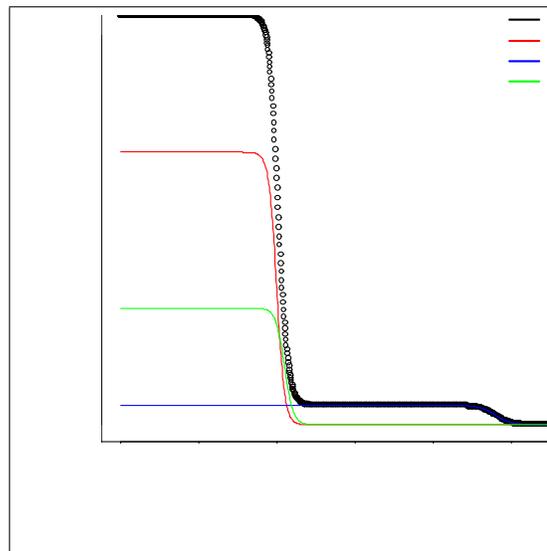


Figura 7.1: Datos simulados de un modelo de mezcla de 3 logísticas.

Se podría fácilmente, haciendo uso de suavizado local con la sentencia `locpoly`, y usando un método tipo kernel, representar también las derivadas de estas funciones simuladas, para cualquier orden de derivación, por ejemplo en `dm3`

se calcularía el suavizado de la tercera derivada y en dm5 la quinta derivada suavizada.

```
dm3<-locpoly(x, logistica(x),gridsize=1000,range.x=c(0,60), drv = 3, kernel
=.ep anechnikov",bandwidth = 0.10)
dm5<-locpoly(x, logistica(x),gridsize=1000,range.x=c(0,60), drv = 5, kernel
="
```

### 7.3.3. Cálculo de máximos y mínimos con algoritmos genéticos en R

Utilización el paquete de R para calcular extremos (máximos y mínimos) usando algoritmos genéticos

```
library(rgenoud)
claw1 <- genoud(logistica, nvars = 1, max = TRUE, pop.size = 3000)
Mon Feb 20 21:44:13 2012
Domains:
-1.000000e+01 <= X1 <= 1.000000e+01
Data Type: Floating Point
Operators (code number, name, population)
(1) Cloning..... 372
(2) Uniform Mutation..... 375
(3) Boundary Mutation..... 375
(4) Non-Uniform Mutation..... 375
(5) Polytope Crossover..... 375
(6) Simple Crossover..... 376
(7) Whole Non-Uniform Mutation..... 375
(8) Heuristic Crossover..... 376
(9) Local-Minimum Crossover..... 0
HARD Maximum Number of Generations: 100
Maximum Nonchanging Generations: 10
Population size : 3000
Convergence Tolerance: 1.000000e-03
Using the BFGS Derivative Based Optimizer on the Best Individual Each
Generation.
Checking Gradients before Stopping.
Using Out of Bounds Individuals.
Maximization Problem.
GENERATION: 0 (initializing the population)
Fitness value... 1.050000e+01
mean..... 1.050000e+01
variance..... 6.998711e-18
#unique..... 3000, #Total UniqueCount: 3000
var 1:
best..... -2.791581e+00
mean..... 1.782730e-01
```

```
variance..... 3.326497e+01
GENERATION: 1
Fitness value... 1.050000e+01
mean..... 1.050000e+01
variance..... 7.864113e-19
#unique..... 1830, #Total UniqueCount: 4830
var 1:
best..... -2.791581e+00
mean..... -2.431630e+00
variance..... 1.163064e+01
...
...
GENERATION: 11
Fitness value... 1.050000e+01
mean..... 1.050000e+01
variance..... 2.621124e-19
#unique..... 1806, #Total UniqueCount: 23090
var 1:
best..... -2.791581e+00
mean..... -2.581007e+00
variance..... 9.533667e+00

GENERATION: 11
Fitness value... 1.050000e+01
mean..... 1.050000e+01
variance..... 2.621124e-19
#unique..... 1806, #Total UniqueCount: 23090
var 1:
best..... -2.791581e+00
mean..... -2.581007e+00
variance..... 9.533667e+00
'wait.generations' limit reached.
No significant improvement in 10 generations.
Solution Fitness Value: 1.050000e+01
Parameters at the Solution (parameter, gradient):
X[ 1] : -2.791581e+00 G[ 1] : 0.000000e+00
Solution Found Generation 1
Number of Generations Run 11
Mon Feb 20 21:44:16 2012
Total run time : 0 hours 0 minutes and 3 seconds.
```

## 7.4. Anexo IV: Implementación en R de algoritmos evolutivos

Para el empleo del Software R con algoritmos evolutivos es necesario cargar en el programa las librerías específicas que vaya a utilizarse (en este caso la DEoptim, GenSA y la cmaes). Una vez instalados estos paquetes o librerías, es precisa la formulación adecuada del problema a resolver, como se verá en el código que se presenta en esta sección.

En el caso del método Differential Evolution, no es preciso dar valores iniciales para los parámetros, aunque si es preciso dar un intervalo de valores mínimos y máximos para cada parámetro. Esto se consigue con la orden:

```
lower = c(1, 10, 0.0001, 10,10,0.001,34,20,0.001,0.1),
```

```
upper = c(30, 12, 0.1, 40,14,0.1,40,22,0.1,1),
```

El código en R con este método será:

```
library(DEoptim)
#####
### Partícula 1 ###
#####
MeanSquares=function(param, V1, V2){
  sum((V2-( param[1] * exp(param[2] - param[3] * V1))/(1 + exp(param[2] -
param[3] * V1)) + (param[4] * exp(param[5] - param[6] * V1))/(1 + exp(param[5]
- param[6] * V1)) + (param[7] * exp(param[8] - param[9] * V1))/(1 + exp(param[8]
- param[9] * V1)) + param[10])^2)
}
Optimizacion <- DEoptim(fn = MeanSquares,
lower = c(1, 10, 0.0001, 10,10,0.001,34,20,0.001,0.1),
upper = c(30, 12, 0.1, 40,14,0.1,40,22,0.1,1),
control = list(itermax = 1500, NP = 90))
```

Un ejemplo para un caso concreto de una muestra de PS con nanotubos:

```
Fichero PSNanotubos PS0TC_2gmin004.txt
datos<- read.delim("PS0TC_2gmin004.txt", dec=".",
sep="\t",fill=TRUE,na.strings="NA",header=TRUE)
attach(datos)
```

Con sólo una logística y nls sale el ajuste:

```
nls(W~(w1 * exp(a1 - b1 * T))/(1 + exp(a1 - b1 * T)) + cte,
start=list(w1=3.04,a1=180,b1=1/60,cte=0.027),
trace = TRUE,data=datos, nls.control(maxiter = 500))
nls(W~(w1 * exp(a1 - b1 * V1))/(1 + exp(a1 - b1 * V1)) + cte,
start=list(w1=3,a1=180,b1=1,cte=0.5),trace = TRUE,data=datos)
Con T=V1*60
nls(W~(w1 * exp(a1 - b1 * T))/(1 + exp(a1 - b1 * T)) + cte,
start=list(w1=3.04,a1=180,b1=1/60,cte=0.027),trace =
TRUE,data=datos, nls.control(maxiter = 500))
```

Salidas:

```
512.0957 : 3.04000000 180.00000000 0.01666667 0.02700000
Nonlinear regression model
model: W ~ (w1 * exp(a1 - b1 * T))/(1 + exp(a1 - b1 * T)) + cte
data: Datos
w1 a1 b1 cte
3.045373 41.314734 0.003748 0.027406
residual sum-of-squares: 15.86
```

A la vista de los resultados habrá que probar con dos logísticas generalizadas y comprobar que se disminuye la suma de cuadrados, que se hace a continuación.

#### 7.4.1. Algoritmo memético: DEoptim y Simulated Annealing

```
datos<- read.delim("tga3.txt ", dec=".", sep="\t",header=TRUE)
library(DEoptim)
library(sfsmisc)
#####
## NOTA: logtotal son los datos TG
## NOTA2: derlogtotal es la derivada primera
## NOTA3: der2logtotal es la derivada segunda
#####
#####
## FUNCIONES #
#####
# definición de la logística generalizada
log=function(c,b,tau,m,x)
{
  c/(1+tau*exp(-b*(m-x)))^(1/tau)
}
# definición de la derivada logística generalizada
derlog=function(c,b,tau,m,x)
{
  c*b*exp(-b*(m-x))/(1+tau*exp(-b*(m-x)))^((1+tau)/tau)
}
# definición de la 2ª derivada de la logística generaliza
der2log=function(c,b,tau,m,x)
{
  (b^2*c*exp(-b*(m-x)))/(tau*exp(-b*(m-x))+1)^((tau+1)/tau)-
  b^2*c*(tau+1)*exp(-2*b*(m-x))*(tau*exp(-b*(m-x))+1)^(-(tau+1)/tau-1)
}
# definición de la 3ª derivada de la logística generalizada
der3log=function(c,b,tau,m,x)
```

```

{
-b^3*c*(tau+1)*exp(-2*b*(m-x))*(tau*exp(-b*(m-x))+1)^(-(tau+1)/tau-1)+
(b^3*c*exp(-b*(m-x)))/(tau*exp(-b*(m-x))+1)^((tau+1)/tau)-
2*b^3*c*(tau+1)*exp(-2*b*(m-x))*(tau*exp(-b*(m-x))+1)^(-(tau-1)/tau-1)-

b^3*c*((-tau-1)/tau-1)*tau*(tau+1)*
exp(-3*b*(m-x))*(tau*exp(-b*(m-x))+1)^((-tau-1)/tau-2)
}
# definición de la 4ª derivada de la logística generalizada
der4log=function(c,b,tau,m,x)
{
-3*b^4*c*(tau+1)*exp(-2*b*(m-x))*(tau*exp(-b*(m-x))+1)^(-(tau+1)/tau-
1)-
b^4*c*tau*(tau+1)*(-tau+1)/tau-1)*exp(-3*b*(m-x))*
(tau*exp(-b*(m-x))+1)^(-(tau+1)/tau-2)+
(b^4*c*exp(-b*(m-x)))/(tau*exp(-b*(m-x))+1)^((tau+1)/tau)-4*b^4*c*
(tau+1)*exp(-2*b*(m-x))*(tau*exp(-b*(m-x))+1)^((-tau-1)/tau-1)-
5*b^4*c*((-tau-1)/tau-1)*tau*(tau+1)*exp(-3*b*(m-x))*
(tau*exp(-b*(m-x))+1)^((-tau-1)/tau-2)-
b^4*c*((-tau-1)/tau-2)*((-tau-1)/tau-1)*tau^2*(tau+1)*exp(-4*b*(m-x))*
(tau*exp(-b*(m-x))+1)^((-tau-1)/tau-3)
}
# definición de la 5ª derivada de la logística generalizada
der5log=function(c,b,tau,m,x)
{
-b^5*c*(tau+1)*exp(-2*b*(m-x))*(tau*exp(-b*(m-x))+1)^(-(tau+1)/tau-1)+
(b^5*c*exp(-b*(m-x)))/(tau*exp(-b*(m-x))+1)^((tau+1)/tau)-
14*b^5*c*(tau+1)*exp(-2*b*(m-x))*(tau*exp(-b*(m-x))+1)^((-tau-1)/tau-1)-

25*b^5*c*((-tau-1)/tau-1)*tau*
(tau+1)*exp(-3*b*(m-x))*(tau*exp(-b*(m-x))+1)^((-tau-1)/tau-2)-
10*b^5*c*((-tau-1)/tau-2)*((-tau-1)/tau-1)*tau^2*(tau+1)*
exp(-4*b*(m-x))*(tau*exp(-b*(m-x))+1)^((-tau-1)/tau-3)-
b^5*c*((-tau-1)/tau-3)*((-tau-1)/tau-2)*
((-tau-1)/tau-1)*tau^3*(tau+1)*exp(-5*b*(m-x))*
(tau*exp(-b*(m-x))+1)^((-tau-1)/tau-4)
}
#####
## Ajuste con DEoptim ##
#####
# Ponderación de los mínimos cuadrados con la primera derivada numérica
de los datos
# param c,b,tau,m
library(DEoptim)
y=datos$V2[seq(1,3201,length.out=800)]
x=as.matrix(datos$V1)[seq(1,3201,length.out=800)] # Con 1500 datos

```

```

MeanSquares=function(param,X,Y) {
if (((param[1] + param[5]+param[9]+param[13] + param[17]+param[21])>101)
|
((param[1] + param[5]+param[9]+param[13] + param[17]+param[21]) < 99)){
r <- Inf}
else{
# param[21]=(100-param[5]-param[1]-param[9]-param[13]-param[17])
r1<- sum(((y -(log(param[1],param[2],param[3],param[4],x)+
log(param[5],param[6],param[7],param[8],x)+
log(param[9],param[10],param[11],param[12],x)+
log(param[13],param[14],param[15],param[16],x)+
log(param[17],param[18],param[19],param[20],x)+
log(param[21],param[22],param[23],param[24],x))))^2)/var(y))
r2<- sum(((D1ss(x,y) -(derlog(param[1],param[2],param[3],param[4],x)+
derlog(param[5],param[6],param[7],param[8],x)+
derlog(param[9],param[10],param[11],param[12],x)+
derlog(param[13],param[14],param[15],param[16],x)+
derlog(param[17],param[18],param[19],param[20],x)+
derlog(param[21],param[22],param[23],param[24],x))))^2)/var(D1ss(x,y)))
r3<- sum(((D2ss(x,y)$y -(der2log(param[1],param[2],param[3],param[4],x)+
der2log(param[5],param[6],param[7],param[8],x)+
der2log(param[9],param[10],param[11],param[12],x)+
der2log(param[13],param[14],param[15],param[16],x)+
der2log(param[17],param[18],param[19],param[20],x)+
der2log(param[21],param[22],param[23],param[24],x))))^2)/var(D2ss(x,y)$y))
r4<- sum(((D1ss(x,D2ss(x,y)$y) -(der2log(param[1],param[2],param[3],param[4],x)+
der2log(param[5],param[6],param[7],param[8],x)+
der2log(param[9],param[10],param[11],param[12],x)+
der2log(param[13],param[14],param[15],param[16],x)+
der2log(param[17],param[18],param[19],param[20],x)+
der2log(param[21],param[22],param[23],param[24],x))))^2)
/var(D1ss(x,D2ss(x,y)$y)))
r<-r1+r2+r3+r4
}
return(r)
}
set.seed(1234)
Optimizacion = DEoptim(fn=MeanSquares,
lower = c(1.5,0.01,0.05,200,1.5,0.01,0.05,200,1.5,0.01,0.05,200,
1.5,0.01,0.05,200,1.5,0.01,0.05,200,1.5,0.01,0.05,200),
upper = c(40,0.2,3,500,40,0.2,3,500,40,0.2,3,500,40,0.2,3,500,
40,0.2,3,500,40,0.2,3,500),
control = list(itermax = 10000, NP = 240,strategy=3))
summary(Optimizacion)
Resultados:
***** summary of DEoptim object *****

```

```

best member : 25.08225 0.01 3 500 1.68226 0.17289 3 407.0701
27.69058 0.01 3 200 19.41555 0.01 0.0779 200 21.70477 0.01633
3 500 3.45536 0.09488 3 323.0312
best value : 3331.997
after : 10000 generations
fn evaluated : 20002 times
param=as.numeric(Optimizacion[[1]]$bestmem)
fit=(der5log(param[1],param[2],param[3],param[4],x)+
der5log(param[5],param[6],param[7],param[8],x)+
der5log(param[9],param[10],param[11],param[12],x)+
der5log(param[13],param[14],param[15],param[16],x)+
der5log(param[17],param[18],param[19],param[20],x)+
der5log((100-param[5]-param[1]-param[9]-param[13]-param[17]),
param[22],param[23],param[24],x))
fit_DTG=(derlog(param[1],param[2],param[3],param[4],x)+
derlog(param[5],param[6],param[7],param[8],x)+
derlog(param[9],param[10],param[11],param[12],x)+
derlog(param[13],param[14],param[15],param[16],x)+
derlog(param[17],param[18],param[19],param[20],x)+
derlog((100-param[5]-param[1]-param[9]-param[13]-param[17]),
param[22],param[23],param[24],x))
fit_tga=(log(param[1],param[2],param[3],param[4],x)+
log(param[5],param[6],param[7],param[8],x)+
log(param[9],param[10],param[11],param[12],x)+
log(param[13],param[14],param[15],param[16],x)+
log(param[17],param[18],param[19],param[20],x)+
log((100-param[5]-param[1]-param[9]-param[13]-param[17]),
param[22],param[23],param[24],x))
windows()
plot(x,der5logtotal,type="l",lwd=4,lty=1,col=1,xlab="Tiempo (s)",ylab="Masa
(%)")
lines(x,fit,col=2,lwd=1)
windows()
plot(x,y,type="l",lwd=4,lty=1,col=1,xlab="Tiempo (s)",ylab="Masa (%)")
lines(x,fit_tga,col=2,lwd=1)
windows()
plot(x,derlogtotal,type="l",lwd=4,lty=1,col=1,xlab="Tiempo (s)",ylab="Masa
(%)")
lines(x,fit_DTG,col=2,lwd=1)
windows()
plot(logtotal,derlogtotal,type="l",lwd=4,lty=1,col=1,xlab="Tiempo (s)",ylab="Masa
(%)")
lines(x,fit_DTG,col=2,lwd=1)
plot(Optimizacion)
#####
#####ANNEALING#####

```

```
#####
library(GenSA)
par=param
### TERCERA SERIE ###
par=param
set.seed(1234)
tol <- 1.05e-16
global.min <- 0
Optimizacion4 <- GenSA(par=par,fn = MeanSquares,
lower = c(1.5,0.01,0.05,200,1.5,0.01,0.05,200,1.5,0.01,0.05,200,1.5,0.01,
0.05,200,1.5,0.01,0.05,200,1.5,0.01,0.05,200),
upper = c(40,0.2,3,500,40,0.2,3,500,40,0.2,3,500,40,0.2,3,
500,40,0.2,3,500,40,0.2,3,500),
control=list(threshold.stop=global.min+tol,verbose=TRUE))
Resultados:
[1] 25.08225018 0.01000000 3.00000000 499.99999998
[5] 1.68225851 0.17289125 3.00000000 407.07013558
[9] 27.69058201 0.01000000 3.00000000 200.00000000
[13] 19.41554655 0.01000000 0.07790158 200.00000000
[17] 21.70476680 0.01632715 3.00000000 500.00000000
[21] 3.45535720 0.09487845 3.00000000 323.03118081
```

Introducción de estos resultados óptimos con el SA en el segundo algoritmo:

```
param=Optimizacion4$par
fit_anneal3=(der5log(param[1],param[2],param[3],param[4],x)+
der5log(param[5],param[6],param[7],param[8],x)+
der5log(param[9],param[10],param[11],param[12],x)+
der5log(param[13],param[14],param[15],param[16],x)+
der5log(param[17],param[18],param[19],param[20],x)+
der5log(param[21],param[22],param[23],param[24],x))
fit_DTG_anneal3=(derlog(param[1],param[2],param[3],param[4],x)+
derlog(param[5],param[6],param[7],param[8],x)+
derlog(param[9],param[10],param[11],param[12],x)+
derlog(param[13],param[14],param[15],param[16],x)+
derlog(param[17],param[18],param[19],param[20],x)+
derlog(param[21],param[22],param[23],param[24],x))
fit_tga_anneal3=(log(param[1],param[2],param[3],param[4],x)+
log(param[5],param[6],param[7],param[8],x)+
log(param[9],param[10],param[11],param[12],x)+
log(param[13],param[14],param[15],param[16],x)+
log(param[17],param[18],param[19],param[20],x)+
log(param[21],param[22],param[23],param[24],x))
windows()
plot(x,der5logtotal,type="l",lwd=4,col=1,xlab="Tiempo (s)",ylab="Masa (%)")
lines(x,fit_anneal3,col=2,lwd=1)
windows()
```

```

plot(x,logtotal,type="l",lwd=4,col=1,xlab="Tiempo (s)",ylab="Masa (%)")
lines(x,fit_tga_anneal3,col=2,lwd=1)
windows()
plot(x,derlogtotal,type="l",lwd=4,col=1,xlab="Tiempo (s)",ylab="Masa (%)")
lines(x,fit_DTG_anneal3,col=2,lwd=1)

MeanSquares=function(param,X,Y){
if ((param[1] + param[5]+param[9]+param[13] + param[17]+ param[21]) >
100.5 | (param[1]+param[5]+param[9]+param[13] + param[17]+ param[21])<99.5){
r <- Inf}
else{
r=sum((datos$V2-
(param[1]/(1+param[4]*exp(-param[2]*(param[3]-datos$V1)))^(1/param[4])+
param[5]/(1+param[8]*exp(-param[6]*(param[7]-datos$V1)))^(1/param[8])+
param[9]/(1+param[12]*exp(-param[10]*(param[11]-datos$V1)))^(1/param[12])+
param[13]/(1+param[16]*exp(-param[14]*(param[15]-datos$V1)))^(1/param[16])+
param[17]/(1+param[20]*exp(-param[18]*(param[19]-datos$V1)))^(1/param[20])+
param[21]/(1+param[24]*exp(-param[22]*(param[23]-datos$V1)))^(1/param[24])))^2)
}
return(r)
}
par=c( 18.159597, 3.424683, 379.645328, 25.203417, 17.542868, 60.149703,
284.999216, 94.903534, 21.320016, 9.848879, 243.792671, 83.663536,
7.053069, 2.985781, 482.916516, 14.888552, 14.641411, 5.035730,
401.795922, 99.283466, 20.783040, 4.597338, 307.112739, 9.934396)
set.seed(1234)
tol <- 1e-13
global.min <- 0
Optimizacion <- GenSA(par=par,fn = MeanSquares,
lower = c(0, 0, 150, 0,0,0,150,0,0,0,150,0,0,0,150,0,0,0,150,0),
upper = c(60, 100, 600, 100, 60, 100, 600, 100, 60, 100, 600, 100,60,
100, 600, 100, 60, 100, 600, 100, 60, 100, 600, 10)
, control=list(threshold.stop=global.min+tol,verbose=TRUE))
Optimizacion[c("value", "par", "counts")]
summary(Optimizacion)
param=Optimizacion$par
#param=as.numeric(Optimizacion[[1]]$bestmem)
fit=param[1]/(1+param[4]*exp(-param[2]*(param[3]-datos$V1)))^(1/param[4])+
param[5]/(1+param[8]*exp(-param[6]*(param[7]-datos$V1)))^(1/param[8])+
param[9]/(1+param[12]*exp(-param[10]*(param[11]-datos$V1)))^(1/param[12])+
param[13]/(1+param[16]*exp(-param[14]*(param[15]-datos$V1)))^(1/param[16])+
param[17]/(1+param[20]*exp(-param[18]*(param[19]-datos$V1)))^(1/param[20])+
param[21]/(1+param[24]*exp(-param[22]*(param[23]-datos$V1)))^(1/param[24])
plot(datos$V1,datos$V2,type="l",lwd=4,lty=1,col=1,xlab="Tiempo (s)",ylab="Masa
(%)" )

```

```

lines(datos$V1,fit,col=2,lwd=1)
windows()
plot(datos$V1,D2ss(datos$V1,D1ss(datos$V1,datos$V2))$y
,type="l",lwd=4,lty=1,col=1,xlab="Tiempo (s)",ylab="Masa (%)")
lines(datos$V1,D2ss(datos$V1,D1ss(datos$V1,fit))$y,col=2,lwd=1)

#####
##### APLICACIÓN ANNEALING #####
#####
#####
#Logística a logística:
log1=log(8.993725,0.13800685, 3.73305343,473.86844810,x)
log2=log(33.00008825, 0.07150258, 1.73023941, 265.27801090,x)
log3=log(12.63065777, 0.08002381,1.52918745, 409.40017752,x)
log4=log( 10.51404629, 0.20235151, 1.32351246, 302.95809492,x)
derlog1=derlog(8.993725,0.13800685, 3.73305343,473.86844810,x)
derlog2=derlog(33.00008825, 0.07150258, 1.73023941, 265.27801090,x)
derlog3=derlog(12.63065777, 0.08002381,1.52918745, 409.40017752,x)
derlog4=derlog( 10.51404629, 0.20235151, 1.32351246, 302.95809492,x)
log1f=log(8.03822203, 0.16113507, 4.36549628, 474.02370035,x)
log2f=log(33.46784844, 0.07177504, 1.81092002, 265.32946239,x)
log3f=log( 13.41877908, 0.09016107, 2.36579706, 408.28812950,x)
log4f=log( 10.21330887, 0.20129822, 1.23001492, 303.06383271,x)
derlog1f=log(8.03822203, 0.16113507, 4.36549628, 474.02370035,x)
derlog2f=log(33.46784844, 0.07177504, 1.81092002, 265.32946239,x)
derlog3f=log( 13.41877908, 0.09016107, 2.36579706, 408.28812950,x)
derlog4f=log( 10.21330887, 0.20129822, 1.23001492, 303.06383271,x)
setEPS(10)
windows(14,7)
op <- par(mfrow = c(1, 2)) # square plotting region,
plot(x,logtotal,type="l",lwd=6,lty=1,cex.axis=1.2,cex.lab=1.2,col=1,xlab="Temperature/
°C",ylab="Mass/ mg")
lines(x,fit_tga_anneal,col=5,lwd=2)
lines(x,log1,col=1,lwd=6)
lines(x,log2,col=1,lwd=6)
lines(x,log3,col=1,lwd=6)
lines(x,log4,col=1,lwd=6)
lines(x,log1f,col=5,lwd=2)
lines(x,log2f,col=5,lwd=2)
lines(x,log3f,col=5,lwd=2)
lines(x,log4f,col=5,lwd=2)
legend(400,60,c("Simulated TG curve and components","fitted TG curve and
components"),col=c(1,2),lwd=2)
plot(x,-derlogtotal,type="l",lwd=6,lty=1,cex.axis=1.2,cex.lab=1.2,col=1,xlab="Temperature/
°C",ylab="Mass/ mg")
lines(x,-fit_DTG_anneal,col=5,lwd=2)
lines(x,-derlog1,col=1,lwd=6)

```

```

lines(x,-derlog2,col=1,lwd=6)
lines(x,-derlog3,col=1,lwd=6)
lines(x,-derlog4,col=1,lwd=6)
lines(x,-derlog1f,col=5,lwd=2)
lines(x,-derlog2f,col=5,lwd=2)
lines(x,-derlog3f,col=5,lwd=2)
lines(x,-derlog4f,col=5,lwd=2)

legend(370,-0.5,c("Simulated DTG curve and components","fitted DTG curve
and components"),col=c(1,2),lwd=2)
par(op)
#####
### CMA: Covariance matrix adapting evolutionary strategy ##
#####
library(parma)
set.seed(1)
ctrl = cmaes.control()
ctrl$options$StopOnWarnings = FALSE
ctrl$cma$active = 1# 1
ctrl$options$TolFun = 1e-12 # 1e-12
ctrl$options$DispModulo=10
ctrl$options$Restarts = 0
ctrl$options$MaxIter = 1000#1000
ctrl$options$PopSize = 600#600
Optimizacion5=cmaes(pars=param, fun= MeanSquares,
lower = c(0.001,0.001,0.001,1,0.001,0.001,0.001,1,0.001,0.001,0.001,1,
0.001,0.001,0.001,1),
upper = c(100,2,8,800,100,2,8,800,100,2,8,800,100,2,8,800),
insigma = 1, ctrl = ctrl)
Coef.Determ=1-sum((fit_tga_anneal-logtotal)^2)/
sum((logtotal-mean(logtotal))^2)
Coef.Determ
#####
#### evolutionary search algorithms with derivative-based (Newton
#### or quasi-Newton) methods
#####
MeanSquares2=function(param,X,Y) {
# if (((param[1] + param[5]+param[9]+param[13] + param[17]) > 80.1) |
# ((param[1] + param[5]+param[9]+param[13] + param[17]) < 79.9)){
# r1 <- Inf}
if (((param[1] + param[5]+param[9]+param[13] + param[17]) != 100)){
r1 <- 99999999999999}
else{
r1<- sum(abs(derlogtotal)*((logtotal -(log(param[1],param[2],param[3],param[4],x)+
log(param[5],param[6],param[7],param[8],x)+
log(param[9],param[10],param[11],param[12],x)+

```

```

log(param[13],param[14],param[15],param[16],x)+
log(param[17],param[18],param[19],param[20],x)))^2))
}
return(r1)
}
library(rgenoud)

Domains=cbind(c(0.001,0.001,0.001,1,0.001,0.001,0.001,1,0.001,0.001,0.001,
1,0.001,0.001,0.001,1,0.001,0.001,0.001,1),
c(100,1,3,800,100,1,3,800,100,1,3,800,100,1,3,800,100,1,3,800))

Optimizacion6 <- genoud(MeanSquares2, default.domains=200,
nvars=20,pop.size=200,max=FALSE,Domains=Domains,max.generations=1000,
starting.values=Optimizacion2$par)
param=Optimizacion6$par
fit_DTG_anneal=(derlog(param[1],param[2],param[3],param[4],x)+
derlog(param[5],param[6],param[7],param[8],x)+
derlog(param[9],param[10],param[11],param[12],x)+
derlog(param[13],param[14],param[15],param[16],x)+
derlog(param[17],param[18],param[19],param[20],x)
#+derlog(param[21],param[22],param[23],param[24],x)
)
fit_tga_anneal=(log(param[1],param[2],param[3],param[4],x)+
log(param[5],param[6],param[7],param[8],x)+
log(param[9],param[10],param[11],param[12],x)+
log(param[13],param[14],param[15],param[16],x)+
log(param[17],param[18],param[19],param[20],x)
#+log(param[21],param[22],param[23],param[24],x)
)
#####
## NELDER MEAD and NEWTON-RAPHSON ##
#####
Optimizacion7=optim(par=param, fn= MeanSquares,
method = c("Nelder-Mead"),#, "BFGS", "CG", "L-BFGS-B", "SANN", "Brent"),
#lower = c(0.001,0.001,0.001,1,0.001,0.001,0.001,1,0.001,0.001,0.001,
#1,0.001,0.001,0.001,1),
#upper = c(100,1,3,800,100,1,3,800,100,1,3,800,100,1,3,800)
control = list(maxit=20000)
Optimizacion7 # valor del fn=20.03897
Optimizacion8=optim(par=param, fn= MeanSquares,
method = c("BFGS"),#, "BFGS", "CG", "L-BFGS-B", "SANN", "Brent"),
#lower = c(0.001,0.001,0.001,1,0.001,0.001,0.001,1,0.001,0.001,0.001,
#1,0.001,0.001,0.001,1),
#upper = c(100,1,3,800,100,1,3,800,100,1,3,800,100,1,3,800)
)
Optimizacion8

```

```

# Optimización8 no alcanza solución
Optimizacion9=optim(par=param, fn= MeanSquares,
  method = c("CG"),#, "BFGS", "CG", "L-BFGS-B", "SANN", "Brent"),
#lower = c(0.001,0.001,0.001,1,0.001,0.001,0.001,1,0.001,0.001,0.001,
#1,0.001,0.001,0.001,1),
#upper = c(100,1,3,800,100,1,3,800,100,1,3,800,100,1,3,800)
control = list(maxit=5000))
Optimizacion9 #
param=Optimizacion7$par
fit_tga=(log(param[1],param[2],param[3],param[4],x)+
log(param[5],param[6],param[7],param[8],x)+
log(param[9],param[10],param[11],param[12],x)+
log(param[13],param[14],param[15],param[16],x)
#+log(param[21],param[22],param[23],param[24],x)
)
Coef.Determ=1-sum((fit_tga-logtotal)^2)/
sum((logtotal-mean(logtotal))^2)
Coef.Determ

```

### 7.4.2. Aplicación a nanotubos

En este caso, se aplica el algoritmo memético propuesto en el capítulo IV a los datos de PS con nanotubos presentados en el capítulo V.

PSNanotubos/PS0TC\_5gmin001.txt

Primer ajuste con sólo una logística y con métodos clásicos usando el nls del

R:

```

nls(W~(w1 * exp(a1 - b1 * V1))/(1 + exp(a1 - b1 * V1)),
start=list(w1=3.7,a1=42,b1=0.5),trace = TRUE,data=Datos)

```

Nonlinear regression model

model:  $W \sim (w1 * \exp(a1 - b1 * V1)) / (1 + \exp(a1 - b1 * V1)) + cte$

Number of iterations to convergence: 9

Achieved convergence tolerance: 1.889e-06

summary(nls\_nanotubos)

Formula:  $W \sim (w1 * \exp(a1 - b1 * V1)) / (1 + \exp(a1 - b1 * V1)) + cte$

Soluciones propuestas para muestra con datos de nanotubos de carbono y

ps:

Parámetros	Estimación	Error	valor t	$P(T >  t )$
w1	3,758119	0,002766	1358,496	$2e - 16$
a1	42,550577	0,216177	196,832	$2e - 16$
b1	0,554386	0,002820	196,569	$< 2e - 16$
cte	-0,013228	0,002381	-5,556	$2,99e - 08$

Ajuste con dos logísticas y los parámetros de entrada calculados con GenSA.

```

fit2_2<-nls(TG2 ~(c1*(1-1/(1+t1*exp(-b1*(Temp-m1)))^(1/t1)))+
(c2*(1-1/(1+t2*exp(-b2*(Temp-m2)))^(1/t2))),start=list(c1=94.9394,

```

```

b1=0.1914,m1=383.12,t1=2.478, c2=3.914,b2=0.0306,
m2=276.4,t2=0.43), trace = TRUE, data=Datos, nls.control(maxiter = 500))
summary(fit2_2)
Formula: TG2 ~ (c1 * (1 - 1/(1 + t1 * exp(-b1 * (Temp - m1))))^(1/t1))) +
(c2 * (1 - 1/(1 + t2 * exp(-b2 * (Temp - m2))))^(1/t2)))
Resultados de los parámetros estimados:
Estimate Std. Error t value Pr(>|t|)
c1 9.707e+01 4.035e-02 2405.59 <2e-16 ***
b1 1.925e-01 2.394e-04 804.12 <2e-16 ***
m1 3.832e+02 1.021e-02 37544.02 <2e-16 ***
t1 2.506e+00 5.752e-03 435.74 <2e-16 ***
c2 3.793e+00 4.006e-02 94.69 <2e-16 ***
b2 3.539e-02 1.143e-03 30.98 <2e-16 ***
m2 2.764e+02 5.090e-01 543.03 <2e-16 ***
t2 6.142e-01 5.168e-02 11.88 <2e-16 ***
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.1539 on 6755 degrees of freedom
Number of iterations to convergence: 14
Achieved convergence tolerance: 9.66e-06
Parameter      Estimate.    t value    Pr(>|t|)
c1          97.1      2405.59    <2e-16
b1           0.19      804.12    <2e-16
m1         383.2     37544.02    <2e-16
t1           2.5      435.74    <2e-16
c2           3.79      94.69    <2e-16
b2           0.035     30.98    <2e-16
m2          276.4     543.03    <2e-16
t2           0.61     11.88    <2e-16
plot(Temp,TG1, main = "nls(*), data, true function and fit")
lines(Temp, predict(fit2_2), col=2)

```

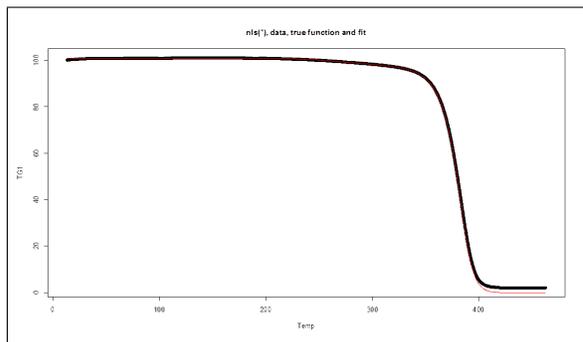


Figura 7.2: Simulación y ajuste de TGA con R.

Propuesta de optimización con el algoritmo evolutivo DE y con mezcla de dos logísticas generalizadas.

```

MeanSquares=function(param,V1,V2){
  sum((V2-( param[1] * exp(param[2] - param[3] * V1))/(1 + exp(param[2] -
param[3] * V1))
  + (param[4] * exp(param[5] - param[6] * V1))/(1 + exp(param[5] - param[6]
* V1))
  + (param[7] * exp(param[8] - param[9] * V1))/(1 + exp(param[8] - param[9]
* V1)) + param[10]^2)
}
Optimizacion <- DEoptim(fn = MeanSquares,
lower = c(1, 10, 0.0001, 10,10,0.001,34,20,0.001,0.1),
upper = c(30, 12, 0.1, 40,14,0.1,40,22,0.1,1),
control = list(itermax = 1500, NP = 90))
summary(Optimizacion)

```

# Bibliografía

- [1] Antonucci, V., Faiella, G., Giordano, M., Nicolais, L. y Pepe, G. (2007). Electrical Properties of Single Walled Carbon Nanotube Reinforced Polystyrene Composites. *Macromolecular Symposia*. **247**, 172–181.
- [2] Arnold, M., Veress, G. E., Paulik, J. y Paulik, F. (1982) A critical reflection upon the application of the Arrhenius model to non-isothermal thermogravimetric curves. *Thermochimica Acta*. **52**, 67-81.
- [3] Artiaga, R.; Varela, A.; Mier, J. L.; García, A.; Losada, R. y Naya, S. (2003). Study of a curing reaction of an epoxy resin. *Materials Science Forum*. **426-432**, 2163-2168.
- [4] Artiaga, R., Varela, A., Mier J.L., García, A., Losada, R. y Naya, S. (2003). Microstructural changes in a Ni-Based Super-Alloy induced by Thermal Treatment. *Materials Science Forum*. **426-432**, 749-754.
- [5] Artiaga, R., Cao, R., Naya, S., González-Martín, B., Mier, J.L. y García, A. (2005). *Separation of overlapping processes from TGA data and verification by EGA*. Journal of the ASTM International.
- [6] Artiaga, R., López-Beceiro, J., Tarrío-Saavedra, J. Gracia-Fernández, C. Naya, S. Mier, J. L. (2011). Estimating the reversing and non-reversing heat flow from standard DSC curves in the glass transition region. *Journal of Chemometrics*. **25**, (6), 287-294.
- [7] Artiaga R., López-Beceiro J., Tarrío-Saavedra J., Mier J., Naya S., Gracia C. (2011). Oxidation Stability of Soy and Palm Based Biodiesels Evaluated by Pressure Differential Scanning Calorimetry. *ASTM Special Technical Publication*. **1477**, 29-41.
- [8] Barbadillo, F., Fuentes, A. Naya, S., Cao, R., Mier, J. L. y Artiaga, R. (2007). Evaluating the logistic mixture model on real and simulated TG curves. *Journal Journal of Thermal Analysis and Calorimetry*. **87**, 1, 223-227.
- [9] Bates, D. M. y Watts, D. G. (1988). *Nonlinear Regression Analysis and Its Applications*. Wiley. London.

- [10] Bates, D. M. y Chambers, J. M. (1992). *Nonlinear models*. En M. Chambers and T. J. Hastie, Wadsworth and Brooks. London.
- [11] Broomhead, D.S. y Lowe, D. (1988). *Multivariable functional interpolation and adaptive networks*. Complex System. New York.
- [12] Berry R. y Smith G. (1993). *Using a genetic algorithm to investigate taxation-induced in capital budgeting*, Proceedings of the International Conference on Neural Networks and Genetic Algorithms. Innsbruck (1993).
- [13] Brown. M.E. Maciejewski. M. Vyazovkin S. Nomen R. Chao. L. Malek J. y Mitsuhashi T. (2000). *Computational aspects of kinetic analysis. Thermochimica Acta*. **355**, 125-143.
- [14] Caballero, J. A. y Grossmann, I. E. (2011). Una revisión del estado del arte en optimización. *Revista Iberoamericana de Automática e Informática Industrial*. **4**, (1), 5 – 23.
- [15] Cao R., Naya S., Artiaga R., García A. y Varela A. (2004). Logistic approach to polymer degradation in dynamic TGA. *Polymer Degradation and Stability*. **85**, 667-674.
- [16] Carrasco F. y Costa J. (1989). Modelo Cinético de la descomposición térmica del poliestireno. *Revista de Ingeniería Química*. 121-129.
- [17] Chatterjee, P. K. y Conrad C. (1970). *Thermal degradation of wood components*. U. S. D. A. Forest Service Research Paper. Madison.
- [18] Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization. Theory and Applications*. **45**, 41–51.
- [19] Cogan, B. (2001). *The evolution of Genetic Algorithms*, Scientific Computing World.
- [20] Coley, U. (1999). *An introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific.
- [21] Conesa, J. A. (2000). *Curso Básico de Análisis Térmico*. Editorial Club Universitario. Madrid.
- [22] Conesa, J. A. y Marcilla, A. (1996). Kinetic study of the thermogravimetric behavior of different rubbers. *Journal of Analytical and Applied Pyrolysis*, **37**, 95-110.
- [23] Davis, L. (1991). *Handbook of Genetic Algorithms*, Van Nostrand, New York. USA.
- [24] Das, S., Suganthan, P. N. (2010). Differential evolution: A survey of the state-of-the-art. *Evolutionary Computation, IEEE Transactions on PP*, **99**, 1 –28.

- [25] Das, S. y Suganthan, P. (2011). Problem definitions and evaluation criteria for cec 2011 competition on testing evolutionary algorithms on real world optimization problems. *Tech. rep., Jadavpur university and Nanyang Technological University*.
- [26] Dennis, J. E., Gay, D. M. y Welsch, R. E. (1981). An Adaptive Nonlinear Least-Squares Algorithm. *ACM Transactions on Mathematical Software*, **7**, 348-368.
- [27] Doyle, C. D. (1961) Kinetic analysis of thermogravimetric data. *Journal of Applied Polymer Science*, **15**, 285-292.
- [28] Fan, J. (1993). Local linear regression smoothers and their minimax efficiency. *The Annals of Statistics*, **21**, 196–216.
- [29] Fan, J. y Gijbels, I. (1996). *Local polynomial modelling and its applications*. Chapman and Hall. London.
- [30] Ferraty, F. y Vieu P. (2001). *Statistical Functionnelle: Modèles de Régression pour Variables Aléatoires Uni, Multi et Infiniment Dimensionées*. Technical Report, Laboratoire de Statistique et Probabilités. Toulouse.
- [31] Flynn, J.H. y Wall, L.A (1966). A Quick, Direct Method for the. Determination of Activation Energy from Thermogravimetric Data. *Polymer Letters*. **4**, 323–328.
- [32] Freeman, B. y Carroll, B. (1958). The application of thermoanalytical techniques to reaction kinetics. The thermogravimetric evaluation of the kinetic of the decomposition of calcium oxalate monhydrate. *Journal of Physical Chemistry*. **62**, 394-397.
- [33] Friedman, H. L. (1964). Kinetics of thermal degradation of char forming plastics from thermogravimetry. Application to a phenolic plastic. *Journal of Polymer Science*. Part C, **6**, 183-195.
- [34] Gay, D. M. (1984). *A trust region approach to linearly constrained optimization in Numerical Analysis*. Springer, Berlin. 171-189.
- [35] Gracia Fernández, C., Gómez Barreiro, S., López Beceiro, J., Tarrío Saavedra, J., Naya, S. y Artiaga, R. (2010). Comparative study of the dynamic glass transition temperature by DMA and TMDSC. C. A. *Polymer Testing*. **29**, 1002-1006.
- [36] Goldberg, D. E. (1989). *Genetic Algorithms in search, optimization, and machine learning*, Addison Wesley, Massachussets. USA.
- [37] Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operation Research*. **13**, 5,533–549.

- [38] González-Álvarez, D. (2013). *Metaheurísticas, Optimización Multiobjetivo y Paralelismo para Descubrir Motifs en Secuencias de ADN*. Tesis Doctoral. Universidad de Extremadura.
- [39] Guoxing, S., Guangming Ch., Jun L., Jiping Y., Jianyun X., Zhengping L. (2009). A facile gemini surfactant-improved dispersion of carbon nanotubes in polystyrene. *Polymer*. **50**, 24, 5787–5793.
- [40] Hansen, N. y Ostermeier, A. (1996). *Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation*. In Proceeding of the IEEE International Conference on Evolutionary Computation (ICEC '96). 312–317.
- [41] Hansen, N. y Kern, S. (2004). *Evaluating the CMA evolution strategy on multimodal test functions*. In *Proc. of the Parallel Problem Solving for Nature - PPSN VIII*, LNCS 3242- 282–291.
- [42] Holland, J. H. (1998). *Adaptation in natural and artificial systems*, 5ª Ed. Massachusetts Institute of Tecnology, MIT Press.
- [43] Howell, B. A, Cui Y, Priddy, D. B. (2003). Assessment of the thermal degradation characteristics of isomeric poly (styrene)s using TG, TG/MS and TG/GC/MS. *Thermochimica Acta*. **396**, 167–177.
- [44] Howell, BA, Chaiwong, K. (2009). Thermal stability of poly(styrene) containing no head-to-head units. *Journal of Thermal Analysis and Calorimetry*. **96**, 22219–22321.
- [45] Iglesias, M.T. Vidal, C. y Verschoren, A. (1996). A Global Approach to Schemata. *International Conference on Intelligent Technologies in Human-Related Sciences (ITHURS'96)*. **I**, 147-152.
- [46] Levenberg, K. (1944). A Method for the Solution of Certain Non-Linear Problems in Least Squares. *Quarterly of Applied Mathematics*. **2**, 164–168.
- [47] López Beceiro, J. (2011). *Modelización de la transición vítrea con relajación entálpica a partir de datos térmicos*. Tesis Doctoral. Universidade da Coruña.
- [48] López-Beceiro, J., Artiaga, R., Gracia-Fernández, C., Tarrío- Saavedra, J., Naya, S. y Mier J. L. (2011). Comparison of olive, corn, soybean and sunflower oils by PDSC. *Journal of Thermal Analysis and Calorimetry*. **104**, (1), 169-175.
- [49] López-Beceiro, J., Pascual-Cosp, J., Artiaga, R., Tarrío- Saavedra, J. y Naya S. (2011). Thermal characterization of ammonium alum. *Journal of Thermal Analysis and Calorimetry*. **104**, (1), 127-130.

- [50] López-Beceiro, J., Gracia-Fernández, C., Gómez-Barreiro, S., Castro-García, S., Sánchez-Andújar, M., Artiaga, R. (2012). Kinetic study of the low temperature transformation of  $co(HCOO)_3[(CH_3)_2NH_2]$ . *Journal of Physical Chemistry, serie C.* **116**, (1), 1219-1224.
- [51] López-Beceiro, J., Gracia-Fernández, C. and Artiaga, R. (2013). A kinetic model that fits nicely isothermal and non-isothermal bulk crystallizations of polymers from the melt. *European Polymer Journal*, **49**, 8, 2233-2246.
- [52] Marquardt, D. (1963). An Algorithm for Least-Squares Estimation of Non-linear Parameters. *SIAM Journal on Applied Mathematics.* **11**, (2), 431-441.
- [53] Merz, P. (2000). *Memetic Algorithms for Combinational Optimization Problems: Fitness Landscapes and Effective Search Strategies*, Ph.D. thesis, Gesamthochschule Siegen, Germany.
- [54] Molina, D., Lozano, M., García-Martínez, C. y Herrera, F. (2010). Memetic algorithms for continuous optimization based on local search chains. *Evolutionary Computation archive*, **18**, 1, 27-63.
- [55] Mohammad, M. y Karen, I. W. (2006). Polymer Nanocomposites Containing Carbon Nanotubes. *Macromolecules.* **39**, 5194-5205.
- [56] Moscato, P.A. (1998). *On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms*. Tech. Rep., Technical Report Caltech Concurrent Computation Program Report 826, Caltech, Pasadena, California.
- [57] Moscato, P.A. (1999). *Memetic algorithms: a short introduction*. McGraw-Hill, London.
- [58] Nadaraya, E.A. (1964). Remarks on nonparametric estimates for density functions and regression curves. *Theory of Probability and Applications.* **15**, 134-137.
- [59] Naya, S., Cao, R. y Artiaga, R. (2003). Local polynomial estimation of TGA derivatives using logistic regression for pilot bandwidth selection. *Thermochimica Acta.* **6**, 319-322.
- [60] Naya, S., Cao, R., López-de-Ullibarri, I., Artiaga, R., Barbadillo, F. y García, A. (2006). Logistic mixture model vs Arrhenius for kinetic study of degradation of materials by dynamic thermogravimetric analysis. *Journal of Chemometrics.* **20**, 158-163.
- [61] Naya, S, Martínez-Vilariño, S. y Artiaga, R. (2009). Effects of thermal cycling on permeability and thermal properties of nanoclay-epoxy composites. *DYNA.* **84**, 2, 151-156.

- [62] Naya, S. (2011). *Modelización de curvas en Análisis Térmico*. Editorial Academia Española. Saarbrücken. Alemania.
- [63] Newman, M. E. J. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*. **46**, 323–351.
- [64] Ozawa, T. (1965). A New Method of Analyzing Thermogravimetric Data. *Bulletin of the Chemical Society of Japan*. **38**, 1881–1886.
- [65] Peña, D. (1987). *Estadística modelos y métodos. Modelos lineales y series temporales*. Alianza editorial. Madrid.
- [66] Popescu, C. (1984). Variation of the maximum rate of conversion and temperature with heating rate in non-isothermal kinetics. Part II. *Thermochimica Acta*, **82**, 387–389.
- [67] Price, K. V., Storn, R. M., Lampinen J. A. (2006). *Differential Evolution: A Practical Approach to Global Optimization*. Springer-Verlag, Berlin.
- [68] Prime, R. B. (1997). *Thermosets. En Thermal characterization of polymeric materials*. Academic Press, San Diego. USA.
- [69] Prime, R. B., Bair, H. E., Gallagher, P. K. y Riga, A. (2009). *Thermogravimetric analysis (TGA). En Thermal Analysis of Polymers Fundamentals and Applications* (editado por J. D. Menczel y R. B. Prime). John Wiley and Sons, San José, USA.
- [70] Qin, A.K. y Suganthan, P.N. (2005). *Self-adaptive Differential Evolution Algorithm for Numerical Optimization*, In 2005 IEEE Congress on Evolutionary Computation. 1785–1791.
- [71] R Development Core Team. (2013). *R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing*, Vienna, Austria. <http://www.R-project.org>.
- [72] Ríos-Fachal, M. Iglesias M. T. y Naya S. (2004). Aplicación de Algoritmos Genéticos para optimizar curvas termogravimétricas. *Anales de Ingeniería Mecánica*. **2**, 15, 1395–1403.
- [73] Ríos-Fachal, M. e Iglesias, M. T. (2005). La varianza epistática como medida de optimización en el empleo de Algoritmos Genéticos. *CD Actas del I Congreso de Estatística e Investigaçao Operacional da Galiza e Norte de Portugal*. Guimarães. Portugal.
- [74] Ríos-Fachal, M., Iglesias, y Vidal, C. (2006). Some features about the convergence process of a Genetic Algorithm. *CD Actas del International Congress of Mathematicians*. Madrid.

- [75] Ríos-Fachal, M., Gracia-Fernández, C. López-Beceiro, J. Gómez-Barreiro, S. Tarrío-Saavedra, J. Ponton, A. Artiaga R. (2013). Effect of nanotubes on the thermal stability of polystyrene. *Journal of Thermal Analysis and Calorimetry*. **113**, 2, 481-487.
- [76] Ríos-Fachal, M., Tarrío-Saavedra, J., López-Beceiro, J, Naya, S. y Artiaga R. (2013). Proposal of a method for optimizing fitting parameters in Thermal Analysis. *NATAS Notes*, 2013. Chentucky. USA.
- [77] Ríos-Fachal, M., Tarrío-Saavedra, J., López-Beceiro, J, Naya, S. y Artiaga, R. Proposal of a method for optimizitting parameters in Thermal Analysis. *Enviado a Journal of Thermal Analysis and Calorimetry*, en Agosto 2013. Enviada revisión en Octubre 2013.
- [78] Reynoso-Meza, G. Sanchis, J. Blasco, X y Martínez, M. (2013). *Revista Iberoamericana de Automática e Informática industrial*. **10**, 251–268.
- [79] Roy, R., Hinduja, S., Teti, R. (2008). Recent advances in engineering design optimisation: Challenges and future trends. *CIRP Annals - Manufacturing Technology*. **57** (2), 697–715.
- [80] Said, L. B. (2010). The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making. *IEEE Transactions on Evolutionary Computation*. **14**, 5, 801–818.
- [81] Saridakis, K. y, Dentsoras, A. (2008). Soft computing in engineering design - a review. *Advanced Engineering Informatics*. **22**, (2), 202 – 221.
- [82] Savitzky, A. and Golay, M. (1964). Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*. **36**, 1627-1639.
- [83] Sebio-Puñal, T., Naya, S., López-Beceiro, J., Tarrío-Saavedra, J. y Artiaga R. (2012). Thermogravimetric analysis of wood, holocellulose, and lignin from five wood species. *Journal of Thermal Analysis and Calorimetry*. **109**, (3), 1163-1167.
- [84] Sesták, J. y Berggren, G. (1971). Study of the kinetics of the mechanism of solid-state reactions at increasing temperaturas. *Thermochimica Acta*. **3**, 1-12.
- [85] Slobodian, P, Králová, D, Lengálová, A, Novotný, R. y Sáha, P. (2010). Adaptation of polystyrene/multi-wall carbon nanotube composite properties in respect of its thermal stability. *Polymer Composites*. **31**, 452–458.
- [86] Storn, R. M., Price, K. V. (1997). Differential Evolution. A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*. **11**, (4), 341-359.

- [87] Storn, R. M. (2010). *Differential Evolution Homepage*. International Computer Science Institute. <http://www.icsi.berkeley.edu/~storn/code.html>. University of California, Berkeley.
- [88] Storn, R. M. (2013). Differential evolution homepage. In: International Computer Science Institute. University of California, Berkeley, 2013. <http://www.icsi.berkeley.edu/~storn/code.html>.
- [89] Sung, T. K., Hyoung, J. Ch. y Soon, M. H. (2007). Bulk polymerized polystyrene in the presence of multiwalled carbon nanotubes. *Colloides Polymer Science*. **285**, 593–598
- [90] Suwardie, J. H. y Artiaga, R. (2000). The determination of heat of curing accompanied by reactant volatilization using simultaneous thermal analysis (STA). *Thermochimica Acta*. **357**, 205-210.
- [91] Suwardie, J. H, Artiaga, R. y Barbadillo, F. (2002). Simultaneous thermal analysis of hexahydrophthalic anhydride. *Thermochim Acta*. **392-393**, 289–294.
- [92] Tarrío-Saavedra, J., López-Beceiro, J. Naya S. y Artiaga R. (2008). Effect of silica content on fumed silica/epoxy composites. *Polymer Degradation and Stability*. **93**, 12, 2133-2137.
- [93] Tarrío-Saavedra, J, Lopez-Beceiro, J, Naya, S. y Artiaga, R. (2010). Influential factors on the oxidation stability of biodiesel: statistical study. *DYNA*. **85-4**. 341-350.
- [94] Tarrío-Saavedra, J. Lopez-Beceiro, S. Naya, C. Gracia y R. Artiaga. (2011). Controversial effects of fumed silica on the curing and thermomechanical properties of epoxy composites. *Express Polymer Letters*. **48**, 4-6, 382- 395.
- [95] Turi, A. (1997). *Thermal characterization of polymeric materials*. Academic Press, San Diego, CA. USA.
- [96] Vyazovkin, S. (1996). A unified approach to kinetic processing of non-isothermal data. *International Journal of Chemical Kinetics*. **28**, 95-101.
- [97] Vyazovkin, S, Sbirrazzuoli N. (1999). Kinetic methods to study isothermal and nonisothermal epoxyanhydride cure. *Macromolecular Chemistry and Physics*. **200**, 230294–230320.
- [98] Wright, A. H. (1991). *Genetic Algorithms for Real Parameter Optimization*. Morgan Kaufman.
- [99] Zhang, Q. y Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on descomposition, *IEEE Transactions on Evolutionary Computation*. **11**, 6, 712–731.