

# Modelado de sistemas de crecimiento microbiano mediante redes de regresión evolutivas

Alfonso C. Martínez<sup>1</sup>, Francisco J. M. Estudillo<sup>1</sup>, Nicolás García<sup>2</sup>, César Hervás<sup>2</sup>, Rosa M<sup>a</sup> García<sup>3</sup>

<sup>1</sup> Facultad de Ciencias Económicas y Empresariales. ETEA  
14005- Córdoba- España acme@etea.com

<sup>2,3</sup> Departamentos de {Computación, Bromatología y TA} Universidad de Córdoba  
14071- Córdoba- España

**Resumen-** El problema que se presenta en este trabajo es el modelado de sistemas a través de un algoritmo evolutivo aplicado a un determinado tipo de redes neuronales denominadas redes de regresión. Mediante este algoritmo se determina la arquitectura óptima de las redes de regresión asociadas a una tipología de funciones relacionada con el sistema que se desea modelar. La metodología propuesta se aplica a un problema de microbiología predictiva en el que se analizan las relaciones que existen entre los parámetros de crecimiento de los microorganismos y las condiciones del envasado de los alimentos. Concretamente, se determinan los parámetros de crecimiento microbiano asociados al efecto que sobre dicho crecimiento tienen las características de manipulación y almacenamiento de los alimentos como son la temperatura, el pH, la concentración de NaCl y el nivel de NaNO<sub>2</sub> en ambiente de aerobiosis. El método tradicional de modelar estas relaciones es mediante superficies de respuesta, donde los parámetros del modelo polinómico se obtienen a través de un algoritmo de Levenberg-Marquard. La comparación de este procedimiento con los modelos obtenidos con nuestra metodología, presenta unos errores medios muy inferiores, tanto en el conjunto de entrenamiento como en el de generalización, aunque el número de parámetros a estimar aumente.

**Palabras clave-** Redes de regresión; Algoritmos evolutivos; Microbiología predictiva; Superficies de respuesta.

## I. INTRODUCCIÓN

El modelado de sistemas es en la actualidad uno de los problemas de mayor interés en numerosas ramas de la ciencia. En diversas áreas de aplicación, surge el problema de establecer una relación funcional entre las diferentes variables que intervienen en el fenómeno que se está estudiando. Un ejemplo interesante lo encontramos en el campo de la microbiología predictiva en donde se plantea un problema que consiste en el estudio de la relación que existe entre los parámetros de crecimiento de microorganismos y las condiciones del envasado de los alimentos. La resolución de este problema se ha abordado tradicionalmente mediante superficies de respuesta, donde los parámetros del modelo polinómico se obtienen a través de un algoritmo

de búsqueda local del tipo Levenberg-Marquard. En otros trabajos se ha intentado modelar estas relaciones causa efecto mediante el diseño de redes neuronales artificiales [12], [13], aunque la principal dificultad de estos modelos es su falta de interpretabilidad asociada a su estructura de caja negra, lo que complica el análisis de sensibilidad de los modelos.

Una alternativa interesante es tratar el modelado de sistemas usando algoritmos evolutivos que, por una parte, permitan obtener expresiones analíticas del modelo que puedan ser interpretables, y por otra, no queden atrapados en mínimos locales durante el proceso de búsqueda. Los algoritmos evolutivos son un tipo de algoritmos de búsqueda estocástica que permiten encontrar la solución en espacios complejos. Dichos algoritmos se han utilizado con éxito en el campo de las redes neuronales para encontrar la estructura adecuada de la red. Incluso, también se han utilizado para calcular el valor de los pesos de las conexiones, evitando quedar atrapado en mínimos locales, generalmente derivados de un sobreentrenamiento [5],[6],[7].

En el proceso de evolución, son esenciales los operadores de selección, replicación y mutación, para introducir por una parte presión selectiva y por otra diversidad, de forma tal que el algoritmo de búsqueda de la mejor red presente un compromiso entre su capacidad de explotar la localización de las mejores soluciones y explorar otras localizaciones del espacio, con el fin de obviar el problema de la obtención de óptimos locales. La no inclusión del operador de cruce se debe al problema de la permutación [5] asociado a este tipo de codificación del cromosoma, lo que hace que sea en general poco eficiente.

En el trabajo propuesto, a partir del conocimiento previo de la tipología de funciones que se quiere modelar, se plantea el diseño de algoritmos evolutivos para realizar la búsqueda de la función que modelice el fenómeno estudiado [14]. En el algoritmo, se utiliza como elemento de representación de las funciones del modelo, un tipo de redes neuronales conocidas como redes de regresión [1]. Este tipo de redes se caracteriza fundamentalmente por ser fácilmente interpretables.

El método de trabajo propuesto consiste en aplicar las técnicas y algoritmos propios de la computación

evolutiva a las redes de regresión, evolucionando la estructura y los pesos de las conexiones de éstas con el objetivo de llegar a encontrar la mejor función dentro de la familia de funciones propuesta en el sentido de que modelice el conjunto de datos con pequeños errores de generalización y con una expresión lo más simplificada posible.

## II. METODOLOGÍA PROPUESTA

Comenzamos definiendo la familia de funciones que se usará en el proceso de modelización y su representación a través de la correspondiente red de regresión. La tipología de funciones a analizar en este trabajo está asociada a combinaciones lineales de expresiones basadas en la resolución de ecuaciones cinéticas de primer orden, cinéticas que aunque sencillas para una variable independiente no lo son para varias variables y menos cuando consideramos una mixtura de ellas.

### A. Tipología de Funciones y Redes de Regresión asociadas.

Análiticamente, el tipo de función con el que trabajaremos es el siguiente:

$$f: A \subset \mathbb{R}^{N^{(0)}} \rightarrow \mathbb{R}$$

$$f(x_1, x_2, \dots, x_{N^{(0)}}) = \sum_{j=1}^{N^{(1)}} \bar{w}_j \left( \prod_{i=1}^{N^{(0)}} x_i^{w_{ji}} \right) \quad (1)$$

donde

$$\bar{w}_j \in [-M, M] \subset \mathbb{R}, w_{ji} \in [-L, L] \subset \mathbb{R},$$

$$i = 1, 2, \dots, N^{(0)}, j = 1, 2, \dots, N^{(1)}$$

siendo  $N^{(0)}$  el número de variables independientes y  $N^{(1)}$  el número de términos del modelo. El dominio de definición de  $f$  es

$$A = \left\{ (x_1, x_2, \dots, x_{N^{(0)}}) \in \mathbb{R}^{N^{(0)}} : 0 < x_i \leq K \right\}$$

En el caso más sencillo, con dos variables independientes, la función definida en (1), vendría dada por:

$$f(x_1, x_2) = \sum_{j=1}^{N^{(1)}} \bar{w}_j x_1^{w_{j1}} x_2^{w_{j2}} \quad (2)$$

La tipología de funciones definidas en (1) puede considerarse una generalización de las superficies de respuesta [10]. Observemos, por ejemplo, que si en (1) elegimos convenientemente los exponentes  $w_{ji} \in \{0,1\}$  se obtiene una superficie de respuesta cuadrática del tipo:

$$f(x_1, x_2, \dots, x_n) = c_0 + \sum_{i=1}^{N^{(0)}} c_i x_i + \sum_{i,j=1}^{N^{(0)}} c_{ij} x_i x_j \quad (3)$$

Para representar este tipo de funciones utilizamos una estructura de red de regresión [1] en la forma que puede verse en la figura 1. A partir de las redes de regresión, representamos la tipología de funciones definida en (1). Las redes de regresión que consideraremos tienen las siguientes características: una capa de entrada para cada una de las variables de entrada, una sola capa oculta con varios nodos y una única capa de salida con un nodo. Además, los nodos de una misma capa no pueden estar conectados entre sí y no existen conexiones directas entre las capas de entrada y salida.

La estructura de la red es por tanto la siguiente:

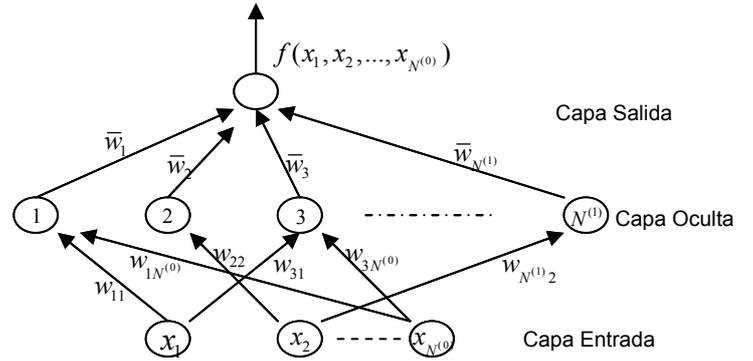


Figura 1. Estructura de red de regresión

Suponemos que la red tiene  $N^{(0)}$  entradas representadas por las variables independientes,  $(x_1, x_2, \dots, x_{N^{(0)}})$ ,  $N^{(1)}$  nodos en la capa oculta, que representa el número de términos del modelo y  $N^{(2)} = 1$  nodo en la capa de salida, que representa la única variable dependiente del modelo. La activación de cada nodo  $j$  de la capa oculta está dada por

$$\phi_j = \prod_{i=1}^{N^{(0)}} x_i^{w_{ij}}, \quad w_{ij} \in [-L, L] \quad (4)$$

donde  $w_{ji}$  son los pesos que conectan el nodo  $j$  de la capa oculta con el nodo  $i$  de la capa de entrada. Por otra parte, la activación de cada nodo de la capa de salida está dada por

$$\sum_{j=1}^{N^{(1)}} \bar{w}_j \phi_j, \quad \bar{w}_j \in [-M, M] \quad (5)$$

siendo  $\bar{w}_j$  el peso que conecta el nodo  $j$  de la capa oculta con el nodo de salida. Además, las funciones de transferencia de cada nodo de la capa oculta y la función de transferencia del único nodo de la capa de salida son iguales a la función identidad. De esta forma, la tipología de funciones definidas en (1) queda representada por la red de regresión que acabamos de definir.

### B. Función de Error.

El problema del modelado de un sistema mediante un conjunto de datos se traduce en encontrar el mínimo global de una función de error que mida la diferencia

existente entre los valores conocidos de la variable predictiva y la predicción de los mismos realizada a través del modelo. Normalmente, la función de error utilizada es la suma de residuos al cuadrado. En el presente trabajo, debido a la tipología de las funciones consideradas y al dominio en el que se definen, se ha elegido la siguiente función de error:

$$E = \frac{1}{N^{(p)}} \left[ \sum_{k=1}^{N^{(p)}} \frac{(y_k - \hat{y}_k)^2}{|y_k|} \right]^{1/2} \quad (6)$$

La función de error definida en (6) ha dado buenos resultados en el modelado de redes de regresión [1].

La aptitud  $A(R)$  de una red  $R$  se calcula como una función decreciente del error  $E(R)$  a partir de la expresión:

$$A(R) = \frac{1}{1 + E(R)}, \quad 0 < A(R) \leq 1 \quad (7)$$

Definida la tipología de funciones y la función de error con las que trabajaremos, pasamos a continuación a describir el algoritmo evolutivo usado en el proceso. El algoritmo está basado en los trabajos de Fogel [2], Angeline et al. [5] y N. García y otros [7].

### C. Proceso General de Evolución.

La idea básica es el uso de los operadores de selección, de replicación y de mutación (paramétrica y estructural) en el proceso de evolución. La evolución de la topología de la red corresponde, en el contexto que acabamos de describir de las redes de regresión, a la búsqueda de la estructura de la función que mejor se ajuste a la nube de puntos del conjunto de entrenamiento, determinando el número de monomios de la función y las variables presentes en cada monomio. Mientras que la evolución de los pesos de la red se corresponde con la evolución de los coeficientes de cada monomio y de los exponentes de cada una de las variables. El algoritmo se estructura en los siguientes pasos:

- 1.- Generación de la población
- 2.- Repetir hasta que se cumpla la condición de parada
  - Calcular la aptitud para cada individuo
  - Ordenar de mayor a menor según la aptitud
  - Copiar el mejor en la siguiente generación
  - Replicar el 10% mejor de la población por el 10% peor
  - Aplicar mutación paramétrica al 10% mejor
  - Aplicar mutación estructural al 90% restante

Los porcentajes se han determinado mediante prueba y error, aunque en la actualidad se están haciendo diseño de experimentos para determinar valores óptimos para estos parámetros.

### D. Algoritmo Evolutivo.

El algoritmo se inicia generando aleatoriamente  $N_R$  redes. Se comienza eligiendo el número de nodos ocultos a partir de una distribución uniforme en el intervalo  $(0, N^{(1)})$ , donde  $N^{(1)}$  corresponde al número máximo de nodos ocultos de cada una de las redes de la población. El número de conexiones entre cada nodo de la capa oculta y los nodos de la capa de entrada queda determinado a partir de una distribución uniforme en el intervalo  $(0, N^{(0)})$ , siendo  $N^{(0)}$  el número de variables independientes. Definida la topología de la red, se asigna a cada conexión un peso, a partir de una distribución uniforme en el intervalo  $[-L, L]$  para los pesos entre la capa de entrada y la oculta y  $[-M, M]$  para los pesos entre la capa oculta y la de salida. Tras la generación aleatoria de la población de  $N_R$  redes, se inicia el proceso de selección. Se realiza una selección por elitismo del 10%. Las redes seleccionadas sustituyen al 10% con peor aptitud, de forma que el número  $N_R$  de redes permanezca constante durante la evolución. Las mutaciones realizadas en el algoritmo son de dos tipos: paramétricas y estructurales. Las mutaciones paramétricas afectan a los pesos de la red y las mutaciones estructurales afectan a la topología de la red (nodos ocultos y conexiones). La severidad de las mutaciones depende de la temperatura  $T(R)$  de la red definida por:

$$T(R) = 1 - A(R), \quad 0 \leq T(R) \leq 1 \quad (8)$$

Cuando el proceso de búsqueda se está iniciando, la aptitud de la red es cercana a cero y la temperatura es alta, por lo que las mutaciones son más fuertes. A medida que nos acercamos a la solución, aumenta la temperatura, por lo que los cambios que se realizan tanto en los pesos como en la estructura de las redes van siendo más pequeños.

Las mutaciones de tipo paramétricas consisten en la alteración de cada peso  $w_{ji}, \bar{w}_j$  con ruido gaussiano, donde la varianza de la distribución de Gauss depende de la temperatura de la red. Concretamente, los pesos  $w_{ji}$  de la capa intermedia quedan actualizados en la forma:

$$w_{ji}(t+1) = w_{ji}(t) + \xi_1(t) \quad (9)$$

$$i \in [1, \dots, N^{(0)}], j \in [1, \dots, N^{(1)}]$$

siendo  $\xi_1(t)$  una variable aleatoria,  $N(0, \alpha_1(t)T(R))$  mientras que los pesos de la capa de salida se actualizan de forma análoga,

$$\bar{w}_j(t+1) = \bar{w}_j(t) + \xi_2(t) \quad (10)$$

siendo  $\xi_2(t)$  una variable aleatoria,  $N(0, \alpha_2(t)T(R))$  donde  $\alpha_1(t) \ll \alpha_2(t), \forall t$ . Las mutaciones de los pesos  $w_{ji}$ , correspondientes a los exponentes de las variables

de la función que queremos modelar, han de ser menores que las mutaciones realizadas en los pesos  $\bar{w}_j$ , correspondientes a los coeficientes de los monomios de la función, debido al diferente efecto que tienen dichas modificaciones sobre los valores de la función.

Una vez realizado el cambio en el espacio de los pesos, la aptitud del individuo es recalculada y se aplica un algoritmo estándar de enfriamiento simulado. Si llamamos  $\Delta A$  a la variación de la aptitud antes y después del cambio en los pesos, resulta que

- Si  $\Delta A \geq 0$ , se acepta el cambio
- Si  $\Delta A < 0$ , se acepta el cambio si  $e^{\frac{\Delta A}{T}} < \gamma$ , donde  $\gamma \in U(0,1)$

Los parámetros  $\alpha_1$  y  $\alpha_2$  que con la temperatura determinan las varianzas de las distribuciones normales, van cambiando durante el proceso de evolución, a diferencia de lo que ocurre en [5],[7] donde permanecen fijos durante toda la evolución, en este sentido se realiza un aprendizaje adaptativo. De esta forma evitamos quedar atrapados en mínimos locales y aceleramos el proceso de evolución cuando las condiciones del proceso lo permitan.

Concretamente, la evolución de los parámetros  $\alpha_1$  y  $\alpha_2$  viene dada por:

$$\alpha_i(t+1) = \begin{cases} (1 + \beta)\alpha_i(t), & \text{si } A(s) > A(s-1), \forall s \in \{t, t-1, \dots, t-r\} \\ (1 - \beta)\alpha_i(t), & \text{si } A(s) = A(s-1), \forall s \in \{t, t-1, \dots, t-r\} \\ \alpha_i(t) & \text{en el resto de casos} \end{cases} \quad (11)$$

$i = 1, 2$ , donde  $A(s)$  representa la aptitud de la red con aptitud máxima en la generación  $s$ .

La mutación estructural que se usa en el algoritmo modifica el número de nodos ocultos y las conexiones entre los nodos de la capa intermedia y los nodos de las capas de entrada y salida, afectando de esta forma a la topología de la red. Hemos utilizado 5 tipos de mutaciones utilizadas en [5]: añadir nodos (AN), eliminar nodos (EN), añadir conexiones (AC) y eliminar conexiones (EC), a las que se añade la mutación unir nodos (UN).

La mutación (UN) consiste en sustituir dos nodos de la capa oculta  $a$  y  $b$  elegidos al azar por otro nuevo  $c$ . Se conservarán las conexiones con los nodos comunes y con una probabilidad de 0.5 las que no sean comunes. El peso de las conexiones comunes se calculará de la siguiente manera:

$$\bar{w}_c = \bar{w}_a + \bar{w}_b, \quad w_{ic} = \frac{w_{ia} + w_{ib}}{2} \quad (12)$$

El peso de las conexiones no comunes conservará el valor que tuviera.

A cada individuo afectado por la mutación estructural se le aplicarán secuencialmente cada uno de los diferentes tipos de mutación estructural [5],[7].

Con el fin de limitar la carga computacional del algoritmo, fijamos un intervalo  $[\Delta_{MIN}, \Delta_{MAX}]$  para el número de modificaciones de cada tipo de mutación estructural- excepto para la mutación (UN)-, siendo el número de modificaciones:

$$\Delta_{MIN} + [U(0,1)T(R)(\Delta_{MAX} - \Delta_{MIN})] \quad (13)$$

Los valores mínimo y máximo establecidos para cada tipo de mutación se muestran en la tabla 1:

	AN	EN	AC	EC
$\Delta_{MIN}$	1	1	1	1
$\Delta_{MAX}$	2	3	$3N^{(0)}$	$3N^{(0)}$

TABLA I. NÚMERO DE MODIFICACIONES

La condición de parada del algoritmo se alcanza cuando se llega a una aptitud previamente marcada, o bien, los valores  $\alpha_1(t)$  y  $\alpha_2(t)$  llegan a un valor próximo a cero.

### III. APLICACIÓN EN MICROBIOLOGÍA PREDICTIVA

Uno de los campos de la microbiología predictiva que está teniendo más interés es el análisis de las relaciones que existen entre los parámetros de crecimiento de los microorganismos y las condiciones mediambientales del almacenamiento y empaquetado de alimentos. Sin embargo, es difícil encontrar modelos matemáticos que describan las ecuaciones diferenciales asociadas a la cinética que explica estas relaciones causa-efecto. Los factores más importantes asociados a la manipulación de alimentos son la temperatura, el nivel de pH, la concentración de cloruro sódico (NaCl) y la concentración de nitrito sódico (NaNO<sub>2</sub>).

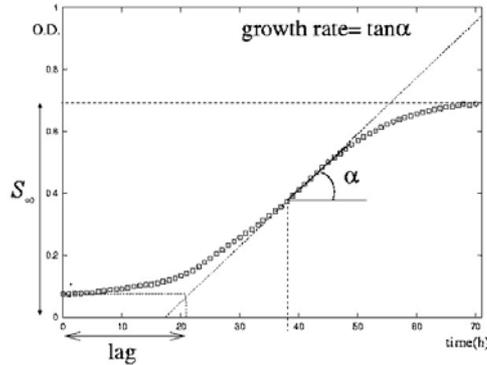


Figura 2. Parámetros de una curva de crecimiento microbiano

Por otra parte, los parámetros de una curva de crecimiento microbiano, Figura 2, más comúnmente utilizados son: la tasa máxima de crecimiento, que denominaremos *grate*, la intersección de la recta de máxima pendiente con un eje paralelo al eje de abscisas que denominaremos *lag*, y la señal a tiempo infinito, esto es, el número de microorganismos existentes cuando la curva de crecimiento tiende a acercarse a su asíntota horizontal o densidad máxima de la población, y que denominaremos  $S_{\infty}$ . En otros trabajos se ha intentado modelar estas relaciones causa-efecto mediante el diseño de redes neuronales artificiales [12] [13], pero la principal dificultad que presentan estos modelos es su falta de interpretabilidad asociada a su estructura de caja negra, lo que dificulta el análisis de sensibilidad de los modelos.

La técnica más utilizada para modelar este tipo de relaciones es la superficie de respuesta, SR, donde se utilizan ecuaciones polinómicas del tipo (3), donde  $f$  es

$$\text{lnlag} = -0.076T + 1.539(\text{NaCl}) + 0.019(\text{NaNO}_2) + 0.059(\text{pH})^2 - 0.215(\text{pH}) * (\text{NaCl}) - 0.003(\text{pH}) * (\text{NaNO}_2) + 0.001(\text{NaCl}) * (\text{NaNO}_2)$$

$$\text{grate} = -0.067T + 0.193(\text{pH}) - 0.044(\text{NaCl}) - 0.001(\text{NaNO}_2) + 0.003T^2 - 0.011(\text{pH})^2 + 0.012(\text{NaCl})^2 + 2.2e-6(\text{NaNO}_2)^2 - 0.008(\text{pH}) * (\text{NaCl})$$

$$S_\infty = 0.048T - 0.235(\text{pH}) - 0.706(\text{NaCl}) - 0.005(\text{NaNO}_2) + 0.022(\text{pH})^2 - 0.023(\text{NaCl})^2 + 0.102(\text{pH}) * (\text{NaCl})$$

ECUACIONES I. RESULTADOS OBTENIDOS CON SUPERFICIES DE RESPUESTA

el parámetro de crecimiento genérico de respuesta y  $c_0$ ,  $c_i$  y  $c_{ji}$  son coeficientes de regresión y  $x_i, i=1, \dots, N^0$ , las variables independientes asociadas a las condiciones de envasado.

A continuación se detalla la forma en la que hemos realizado los experimentos y los valores usados en los parámetros que aparecen en el algoritmo.

#### A. Conjunto de ejemplos.

Los datos utilizados han sido 210 curvas, señal-tiempo, de crecimiento microbiano de *Leuconostoc Mesenteroides* bajo diferentes condiciones de temperatura (10.5, 14, 17.5, 21 y 24°C), pH (5.5, 6, 6.5, 7 y 7.5), concentraciones de cloruro sódico (0.25, 1.75, 3.25, 4.75 y 6.25%) y nitrito sódico (0, 50, 100, 150 y 200ppm).

Estas 210 curvas se corresponden con 30 condiciones experimentales diferentes, de las que se realizaron 7 réplicas, de ellas cinco, elegidas al azar, se tomaron para formar el conjunto de entrenamiento y las otras dos para formar el conjunto de generalización, de esta forma el conjunto de entrenamiento está formado por 150 curvas y el de generalización por 60. Los datos de las variables temperatura, nivel de pH, concentración de cloruro sódico (NaCl) y la concentración de nitrito sódico (NaNO<sub>2</sub>) fueron normalizados en el intervalo [0.1,1.1]. Los valores de entrenamiento y generalización de los parámetros de crecimiento lag, grate y  $S_\infty$  fueron obtenidos a través de modelos de crecimiento exponencial proporcionados por un programa diseñado por Baranyi [11].

#### B. Parámetros usados en el algoritmo evolutivo

Por otra parte, los parámetros que intervienen en el algoritmo evolutivo han tenido los siguientes rangos de variación: los exponentes  $w_{ji}$  han variado en el intervalo  $[-3,3]$  y los coeficientes  $\bar{w}_j$  de cada uno de los monomios han variado en el intervalo  $[-5,5]$ . El número máximo de monomios considerados ha sido  $N^{(1)} = 8$ . El tamaño de la población  $N_R$  ha sido 1000 redes. Por último, los valores considerados para los parámetros  $\alpha_1, \alpha_2, \beta$  y  $r$  en todos los experimentos realizados han sido:  $\alpha_1(0) = 1, \alpha_2(0) = 5, \beta = 0.1,$

$r = 10$ , aunque el algoritmo es bastante robusto con respecto a los valores  $\alpha_1(0)$  y  $\alpha_2(0)$ .

#### IV. RESULTADOS

Los modelos de regresión basados en superficies de respuesta necesitan que se especifique, a priori, el orden del modelo (esto es, si es de primer, segundo o tercer orden); así como valores iniciales para los coeficientes del modelo, de forma tal que el algoritmo de Levenberg-Marquard de búsqueda local pueda obtener el modelo de la forma más eficiente. Por otra parte, con las Redes de Regresión intentamos modelar de forma tal que implícitamente se encuentre el modelo que mejor explique las relaciones causa-efecto entre las variables de envasado y los parámetros de crecimiento.

Utilizando Superficies de Respuesta los mejores modelos para cada parámetro se muestran en las ecuaciones (I), donde para la variable lag se hizo una transformación logarítmica previa. En la tabla 2, columnas 3 y 5, se pueden ver los errores estándar de predicción (SEP), tanto para el conjunto de entrenamiento (SEPESR), como para el conjunto de generalización, (SEPGSR), donde definimos un % de error relativo (SEP), cuya expresión es:

$$SEP = \frac{100}{\bar{f}} \sqrt{\frac{\sum_{i=1}^n (f_i - \hat{f}_i)^2}{n}} \quad (14)$$

siendo  $f$  el valor genérico del parámetro de crecimiento,  $\hat{f}$  el valor estimado y  $n$  el tamaño del conjunto de entrenamiento o del conjunto de generalización según queramos calcular (SEPE) o (SEPG).

	Entrenamiento		Generalización	
	SEPERR	SEPESR	SEPGRR	SEPGSR
Lnlag	5.59	8.82	5.58	8.25
Grate	6.99	9.54	7.71	9.96
$S_\infty$	-13.40	-15.42	-12.31	-15.59

TABLA 2. ERRORES ESTANDAR DE PREDICCIÓN

Utilizando Redes de Regresión se hicieron cinco ejecuciones del algoritmo evolutivo (el coste de cada ejecución en tiempo medio fue de 10 minutos en un PC

$$\begin{aligned}
\text{Inlag} = & 0.03 \frac{T^{1.28}}{(\text{NaCl})^{0.73}} + 0.41 T^{1.23} (\text{pH})^{0.18} (\text{NaNO}_2)^{2.92} + 0.61 \frac{T^{1.72} (\text{pH})^3}{(\text{NaCl})^{0.53}} \\
& -1.33 \frac{T^{2.04} (\text{pH})^{0.28} (\text{NaNO}_2)^{0.17}}{(\text{NaCl})^{0.09}} + 1.20 \frac{(\text{NaCl})^3 (\text{NaNO}_2)^{1.18}}{T^{0.29} (\text{pH})^{0.39}} \\
& + 2.32 \frac{(\text{pH})^{0.02} (\text{NaCl})^{0.19} (\text{NaNO}_2)^{0.05}}{T^{0.03}} - 5 \frac{(\text{pH})^{2.72} (\text{NaCl})^{2.96} (\text{NaNO}_2)^{1.58}}{T^{0.10}} \\
\text{grate} = & -1.28 T^{1.64} (\text{NaCl})^{0.72} (\text{NaNO}_2)^{1.16} + 0.64 \frac{T^{2.39} (\text{pH})^{0.03} (\text{NaNO}_2)^{0.17}}{(\text{NaCl})^{0.015}} \\
& + 0.19 \frac{T^{0.05} (\text{pH})^{0.09} (\text{NaNO}_2)^{0.06}}{(\text{NaCl})^{0.10}} + 2.17 \frac{T^{2.15} (\text{NaCl})^{2.13} (\text{NaNO}_2)^3}{(\text{pH})^{0.04}} \\
S_\infty = & -1.14 \frac{(\text{NaNO}_2)^{2.43}}{(\text{NaCl})^{0.28}} + 1.69 \frac{(\text{pH})^{1.33} (\text{NaCl})^{1.42}}{(\text{NaNO}_2)^{0.1}} + 0.68 \frac{T^{1.74} (\text{pH})^{0.11} (\text{NaNO}_2)^{0.85}}{(\text{NaCl})^{0.31}} \\
& + 2.61 \frac{(\text{pH})^{1.26} (\text{NaCl})^{0.71} (\text{NaNO}_2)^{2.6}}{T^{0.39}} - 2.36 \frac{(\text{pH})^{0.06} (\text{NaCl})^{1.25} (\text{NaNO}_2)^{0.17}}{T^{0.10}} \\
& - 3 \frac{(\text{pH})^3 (\text{NaCl})^{1.4} (\text{NaNO}_2)^{2.06}}{T^{0.48}}
\end{aligned}$$

ECUACIONES II. RESULTADOS OBTENIDOS CON REDES DE REGRESIÓN

pentium IV) y se eligió el mejor modelo para cada parámetro de crecimiento. El número medio de generaciones realizadas por el algoritmo fueron para Inlag, 780, para grate, 2100 y para  $S_\infty$ , 1825. En la tabla 2, columnas 2 y 4, se muestran los errores estándar de predicción (SEP), tanto para el conjunto de entrenamiento, (SEPERR), como para el de generalización, (SEPGRR), asociados a cada uno de los mejores modelos obtenidos para cada parámetro de crecimiento. (véanse Ecuaciones (II))

Se consideró una transformación logarítmica previa en la variable lag para poder comparar nuestros modelos con los obtenidos mediante superficie de respuesta.

Los modelos propuestos en las ecuaciones I son más sencillos de interpretar y tienen menos parámetros a estimar, pero adolecen de falta de precisión; en cambio los modelos propuestos en las ecuaciones II son más difíciles de interpretar puesto que son combinaciones lineales de soluciones de ecuaciones cinéticas de primer orden, tienen más parámetros a estimar, pero son más precisos y permiten un análisis posterior de las cinéticas, de primer orden, subyacentes.

## V. CONCLUSIONES

Los modelos propuestos utilizando redes de regresión y algoritmos evolutivos para optimizar, tanto los errores cometidos como el número de términos, presentan valores del SEP tanto de entrenamiento (5.99,6.99,

-13.40), como de generalización (5.58, 7.71, -12.31) muy inferiores a los propuestos mediante superficies de respuesta (8.82, 9.54, -15.42) y (8.25, 9.96, -15.59) respectivamente. La metodología propuesta abre un campo de modelado de sistemas dinámicos de cinética química y de crecimiento de poblaciones donde la interpretabilidad del modelo pasa por el reconocimiento del tipo de cinética a modelar, pero como hemos visto este reconocimiento se puede hacer mediante la representación de los modelos como redes de regresión y donde la búsqueda de “buenos” modelos está guiada por el algoritmo evolutivo

## VI. AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por la CICYT y por fondos FEDER en los proyectos TIC2001-2577 y TIC2002-04036-C05-02

## VII. REFERENCIAS

- [1] T. Van der Walt, E. Barnard, and J. Deventer, “Process Modeling with the Regression Network”, IEEE Transactions on Neural Networks, vol. 6, No. 1, January 1995
- [2] D. B. Fogel, “An introduction to simulated evolutionary optimisation” IEEE Trans. On Neural Networks, vol. 5, no 1, pp. 3-14, 1994.
- [3] T. Bäck, U. Hammel, and H. \_P. Schwefel, “Evolutionary computation: comments on the history and current state,” IEEE Transactions on Evolutionary Computation, vol 1, no. 1, pp. 3-17, 1997.
- [4] D. B. Fogel, “Using evolutionary programming to greater neural networks that are capable of playing Tic-Tac-Toe,” in International Conference on Neural Networks, San Francisco, CA; IEEE Press, 1993, pp. 875-880.

- [5] P. J. Angeline, G. M. Saunders, and J. B. Pollack, "An evolutionary algorithm that construct recurrent neural networks", IEEE Transactions on Neural Networks, vol. 5, no. 1, January 1994.
- [6] Xin Yao "Evolving Artificial Neural Networks", Proceedings of the IEEE, vol.9, no. 87, pp. 1423-1447,1999.
- [7] N. García, C. Hervás and J. Muñoz, "Covnet: Cooperative coevolución of neural networks", IEEE Transactions on Neural Networks. Conditionally accepted.2003
- [8] S. Wang. "Neural Network techniques for monotonic nonlinear models". Computers Operation Research. vol.21 no. 2, 143-154, 1994.
- [9] Wang. "Nonlinear regression: a hybrid model". Computers Operation Research 26 799-817. 1999.
- [10] R. H. Myers and D. C. Montgomery. Response Surface Methodology: process and product optimization using designed experiments. John Wiley and Sons 2002
- [11] Baranyi, J. Draft manual for the DMFit curve-fitting program. Account, Institute of Food Research, Norwich. 1998
- [12] García-Gimeno, R.M<sup>a</sup>, Hervás-Martinez, C. and de Sioniz, I. Improving artificial neural networks with a pruning methodology and genetic algorithms for their application in microbial growth prediction in foods. Int. J. Food Microbiol. 72, 19-30. 2002
- [13] Hervás, C., Zurera, G., García-Gimeno, R.M<sup>a</sup> and Martínez, J.A. Optimisation of computational neural network for its application to the prediction of microbial growth in foods. Food Sci. Technol. Int. 7, 159-163. 2001
- [14] A. C. Martínez, F.J. Martínez, N. García, C. Hervás. Modelado de Sistemas mediante modelos híbridos de redes neuronales y computación evolutiva, Workshop Aprendizaje y minería de datos Iberamia'02. Sevilla. 67-76..2002