



Redes Neuronales Evolutivas: Aplicaciones

César Hervás-Martínez
Pedro A. Gutiérrez Peña
Grupo de Investigación AYRNA

**Departamento de Informática y Análisis
Numérico**
Universidad de Córdoba
Campus de Rabanales. Edificio Einstein.
Email: chervas@uco.es
pagutierrez@uco.es

2021-2022

1

1



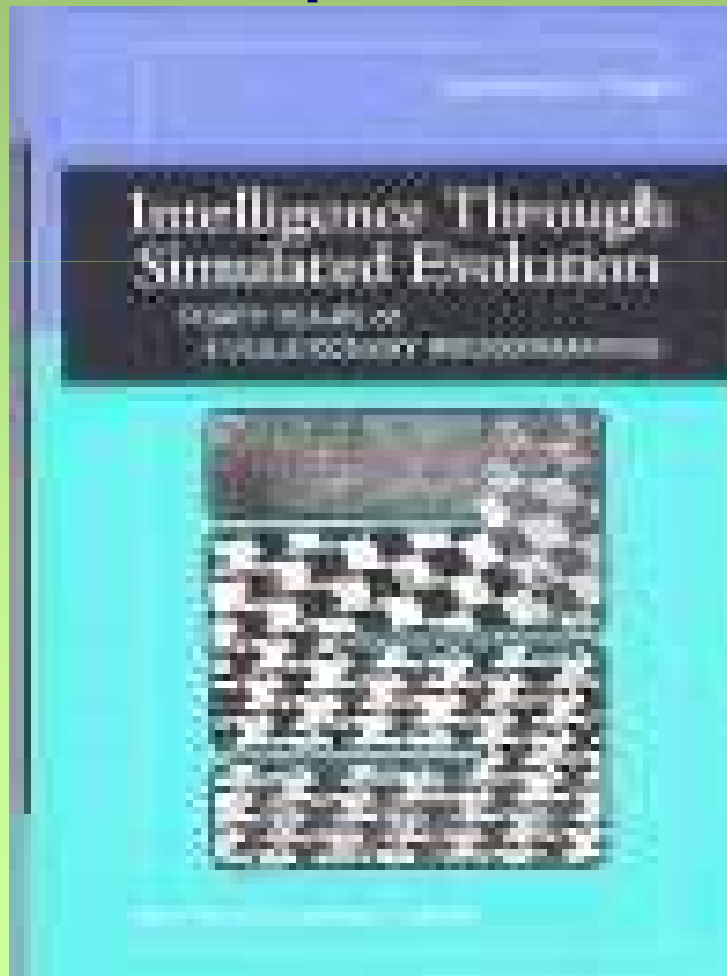
Computación Natural (NC)



Evolutionary Computation
Artificial immune systems
Neural computation
Amorphous computing
Ant colony optimization
Swarm intelligence
Harmony search
Coral reef algorithms
Cellular automata
Artificial life
Membrane computing
Molecular computing
Quantum computing
etc



- Lawrence J. Fogel propuso en 1960 una técnica denominada "programación evolutiva", en la cual la inteligencia se ve como un comportamiento adaptativo.



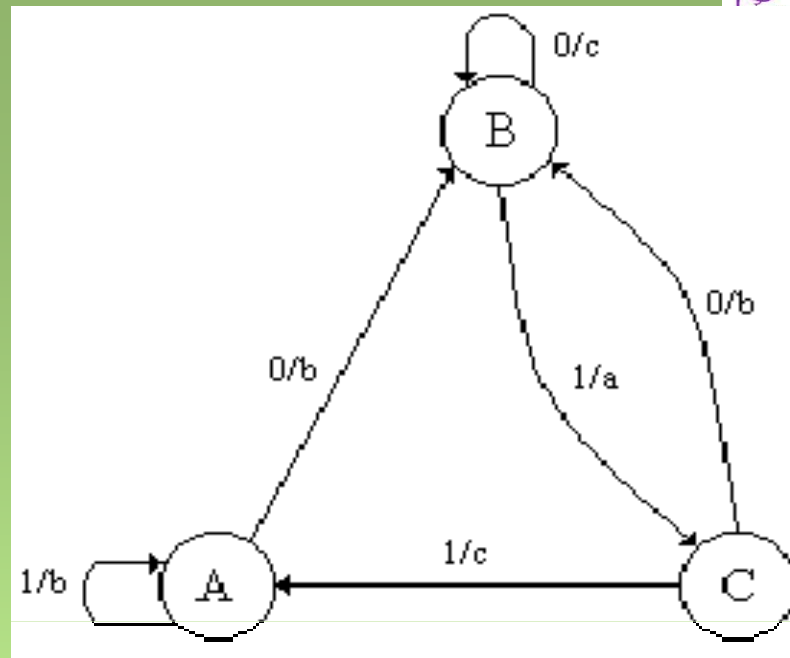


Programación Evolutiva

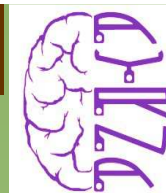


Autómata finito de 3 estados.

Los símbolos a la izquierda del símbolo "/" son de entrada, y los de la derecha son de salida. El estado inicial es C.



Estado Actual	C	B	C	A	A	B
Símbolo de Entrada	0	1	1	1	0	1
Estado siguiente	B	C	A	A	B	C
Símbolo de salida	b	a	c	b	b	a



Para este ejemplo, existen 5 tipos de mutaciones:

Cambiar un símbolo de salida,

Cambiar una transición,

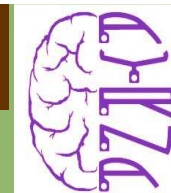
Agregar un estado,

Borrar un estado y

Cambiar el estado inicial.



Programación Evolutiva



Borrar un estado y cambiar el estado inicial sólo se permiten en máquinas de más de un estado.

Las mutaciones se eligen con respecto a una distribución probabilística, que es normalmente uniforme.



Programación Evolutiva



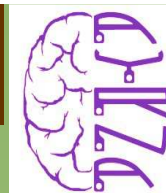
□ El algoritmo básico de la Programación Evolutiva es el siguiente:

- Generar aleatoriamente una población inicial.
- Aplicar operador de mutación.

Calcular la aptitud de cada hijo y usar un proceso de selección mediante torneo (normalmente estocástico) para determinar cuáles serán las soluciones que se retendrán.



Programación Evolutiva



- ❑ La Programación Evolutiva es una abstracción de la evolución al nivel de las especies, por lo que no se requiere el uso de un operador de recombinación (diferentes especies no se pueden cruzar entre sí).
- ❑ Asimismo, usa selección probabilística.



Programación Evolutiva



□ **Algunas aplicaciones de la programación evolutiva son:**

- **Predicción**
- **Generalización**
- **Juegos**
- **Control automático**
- **Planificación de rutas**
- **Diseño y entrenamiento de redes neuronales**
- **Reconocimiento de patrones**



Algoritmo evolutivo



Redes Neuronales Evolutivas



Elementos

- ☐ Pesos de las conexiones
- ☐ Estructura
- ☐ Funciones de transferencia

Representaciones

- ☐ Pesos de las conexiones
 - Representación binaria
 - Representación real
- ☐ Estructura
 - Codificación directa
 - Codificación indirecta

Metodologías

- ☐ Solo pesos de las conexiones
- ☐ Solo estructura
- ☐ Conjuntamente

Paradigmas de la C. Evolutiva

- ☐ Algoritmos Genéticos
- ☐ Programación Evolutiva
(Sin Operador de Cruce)
- ☐ Coevolución...



- ❑ Lawrence J. Fogel propuso en 1960 una técnica denominada "programación evolutiva", en la cual la inteligencia se ve como un comportamiento adaptativo.
- ❑ Las mutaciones se eligen con respecto a una distribución probabilística, que es normalmente uniforme.
- ❑ La Programación Evolutiva enfatiza los nexos de comportamiento entre padres e hijos, en vez de buscar emular operadores genéticos específicos (como en el caso de los algoritmos genéticos).



Esquema del proceso de evolución



Generación de la Población Inicial

Fase de Selección de individuos

**Fase de Generación de
nuevos individuos**

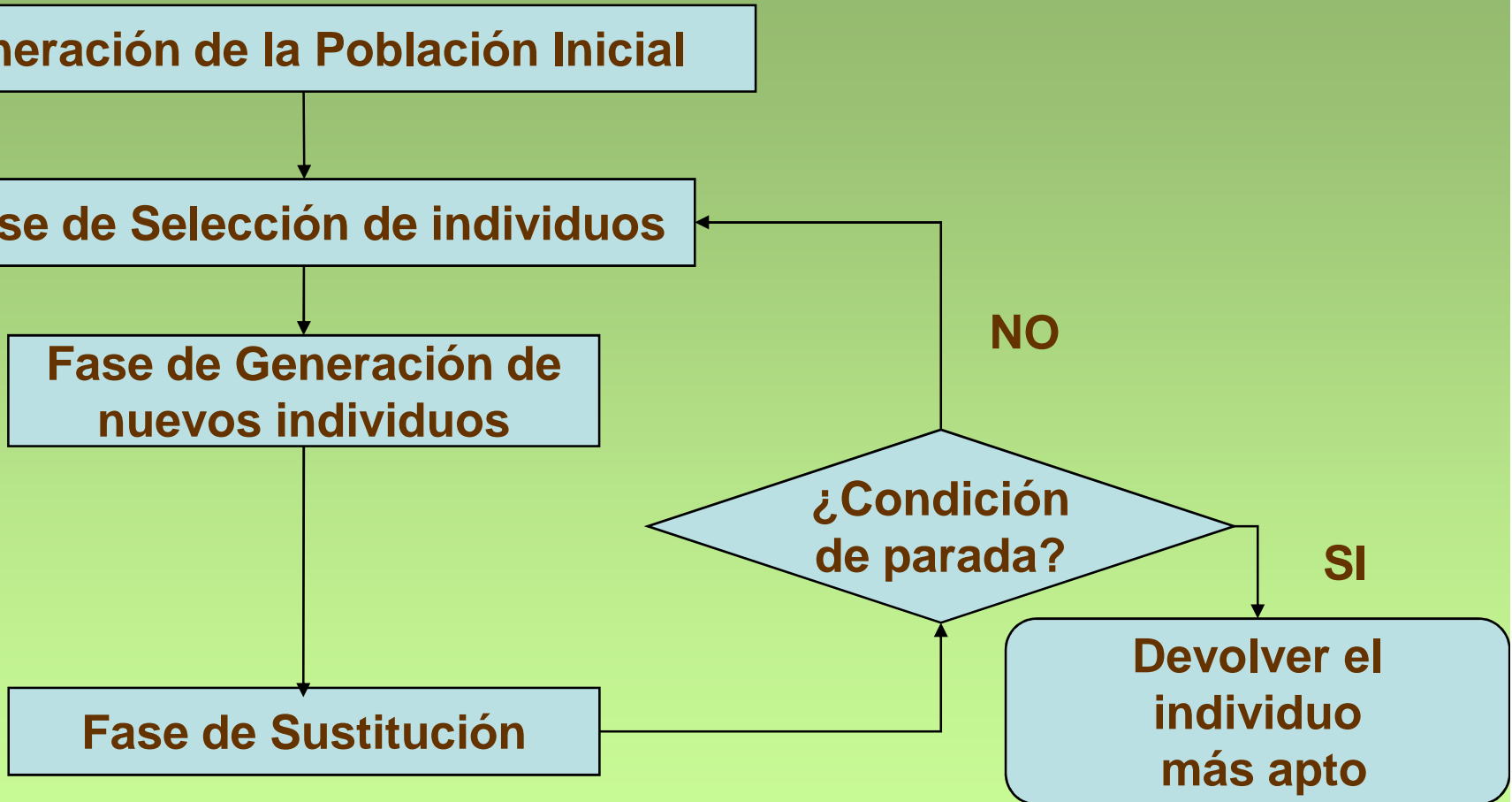
Fase de Sustitución

**¿Condición
de parada?**

NO

SI

**Devolver el
individuo
más apto**

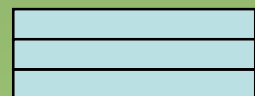




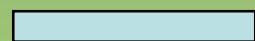
Redes Neuronales Evolutivas



Generación i



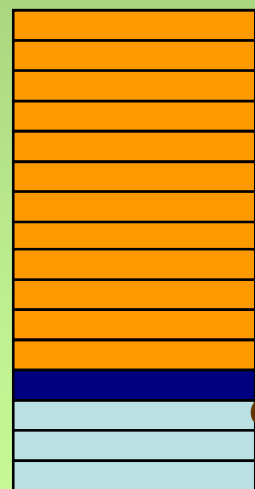
...



mutación
paramétrica



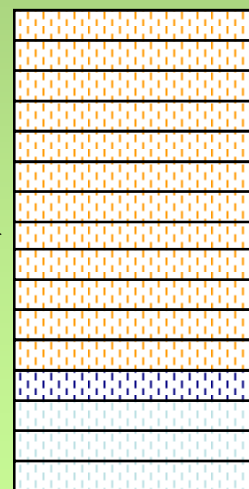
...



...



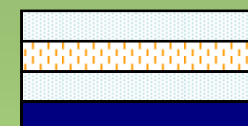
mutación
estructural



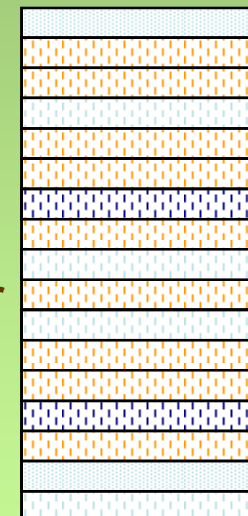
...



Generación $i+1$



...



...



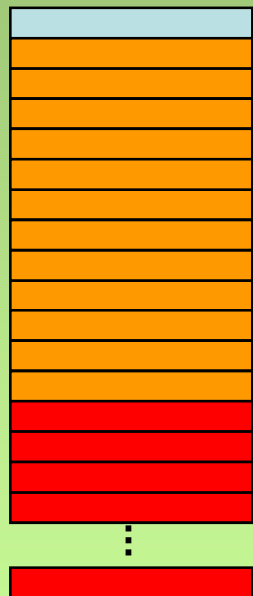
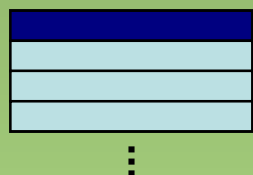
Ordenar



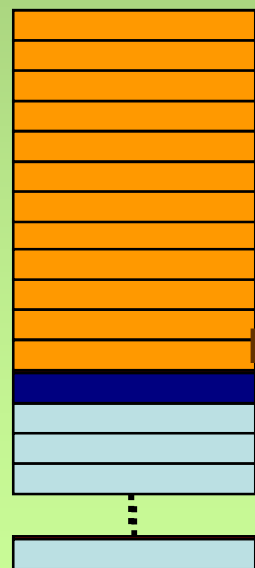
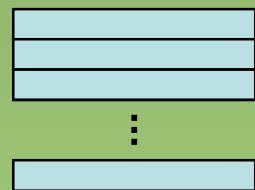
Redes Neuronales Evolutivas



Generación i

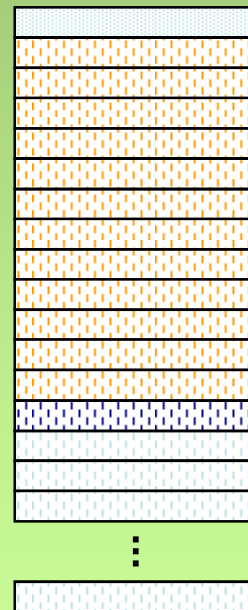
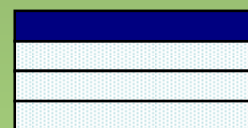
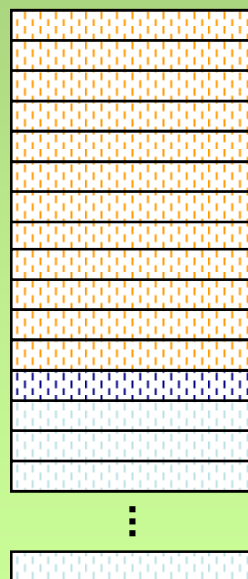
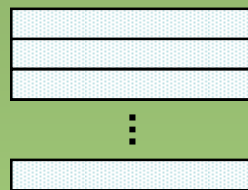


Población de
modelos
(ordenada por
su aptitud)

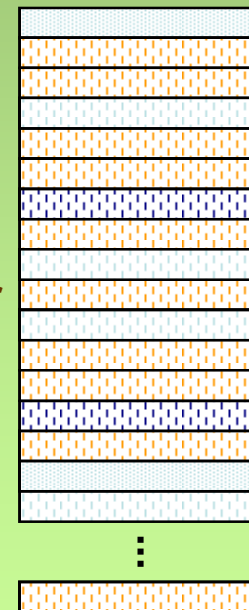
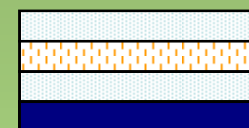


Mutación
Paramétrica

Mutación
Estructural



Generación $i+1$



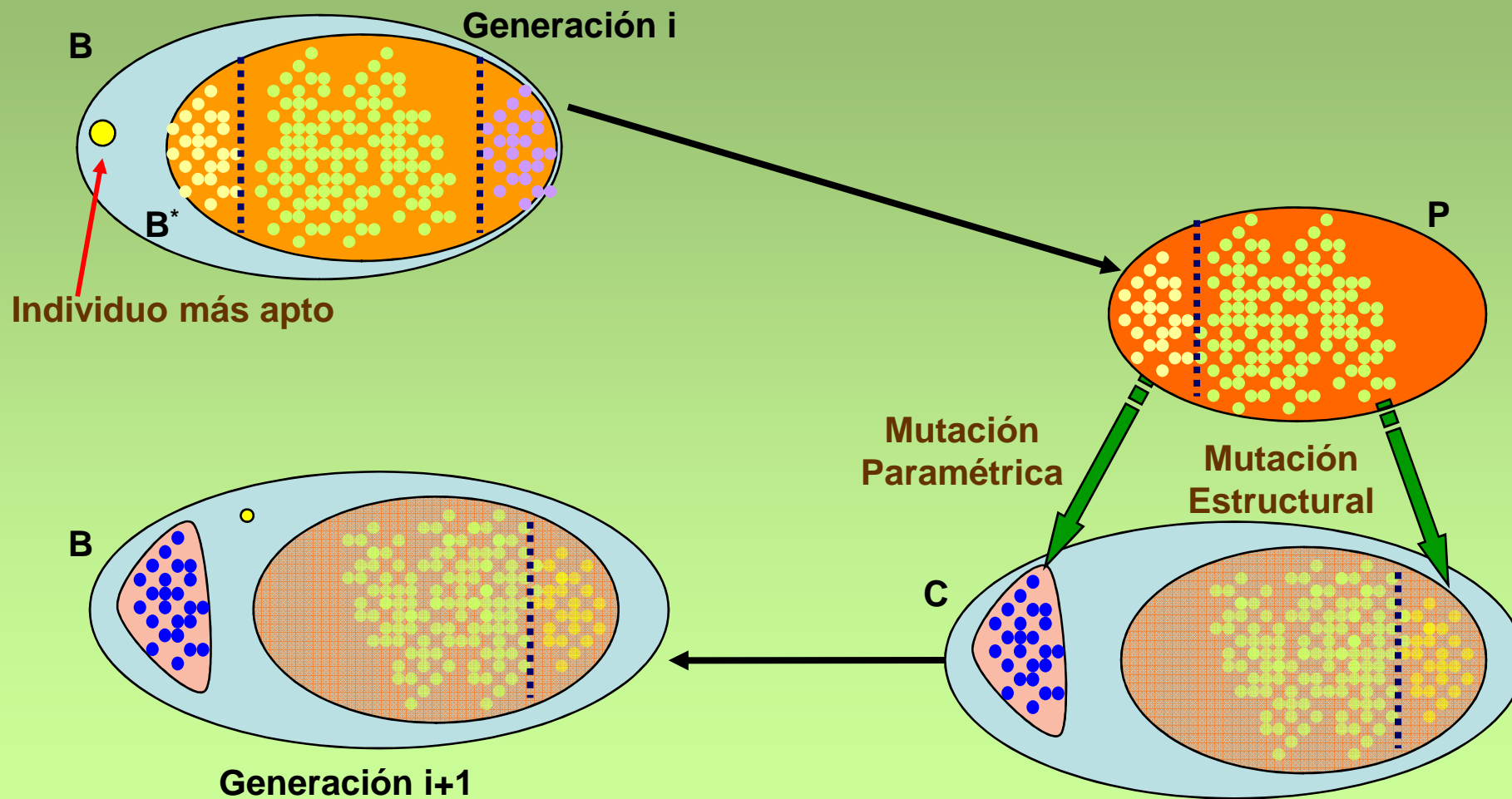
Ordenar

Población de modelos
(ordenada por
su aptitud)

Generación población



Generación de la población





❑ Función de Error

$$MSE(g) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

❑ Término de Regularización

$$S_{estruct}(g) = \gamma \left(1 - \frac{n_n(g)}{m} \right) + (1 - \gamma) \left(1 - \frac{n_c(g)}{n_{cMax}} \right), \quad \gamma \in [0, 1]$$

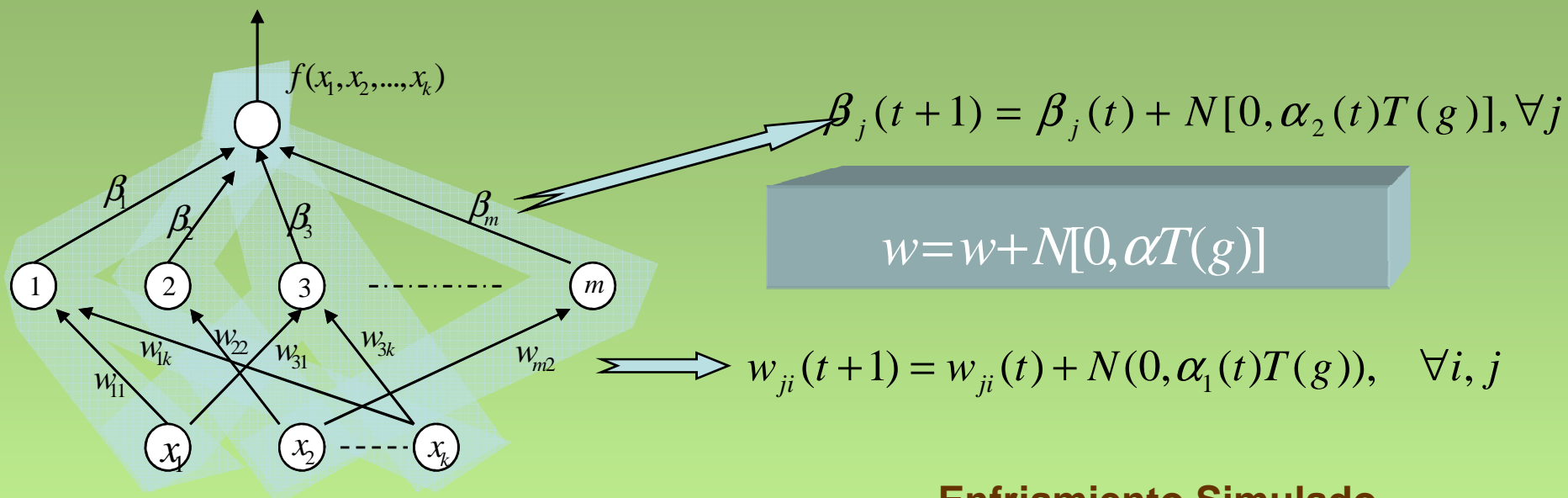
❑ Función de Aptitud

$$A(g) = \alpha \left(\frac{1}{1 + MSE(g)} \right) + (1 - \alpha) S_{estruct}(g), \quad \alpha \in [0, 1]$$

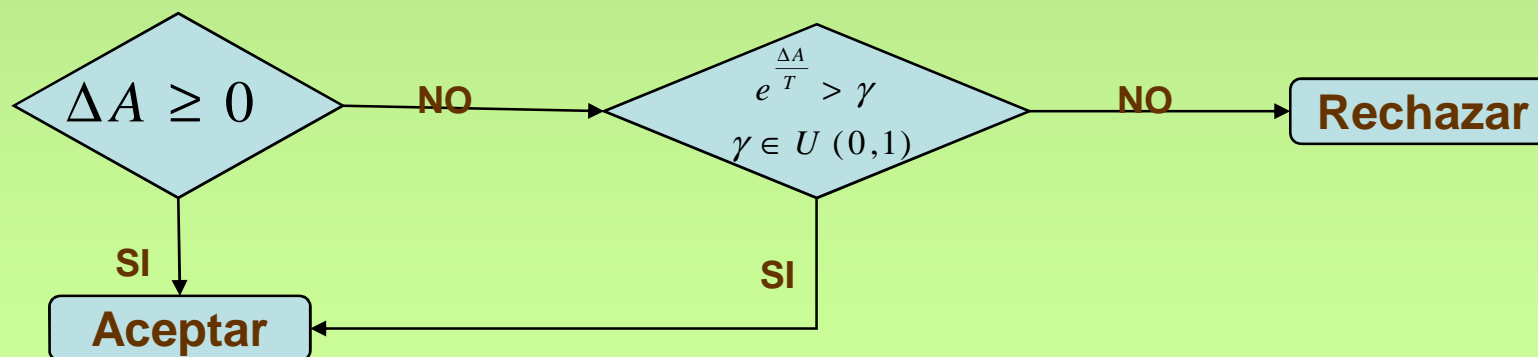
❑ Temperatura $T(g) = 1 - A(g)$, $0 \leq T(g) < 1$



Mutación paramétrica



Enfriamiento Simulado





Evolución de α_1, α_2



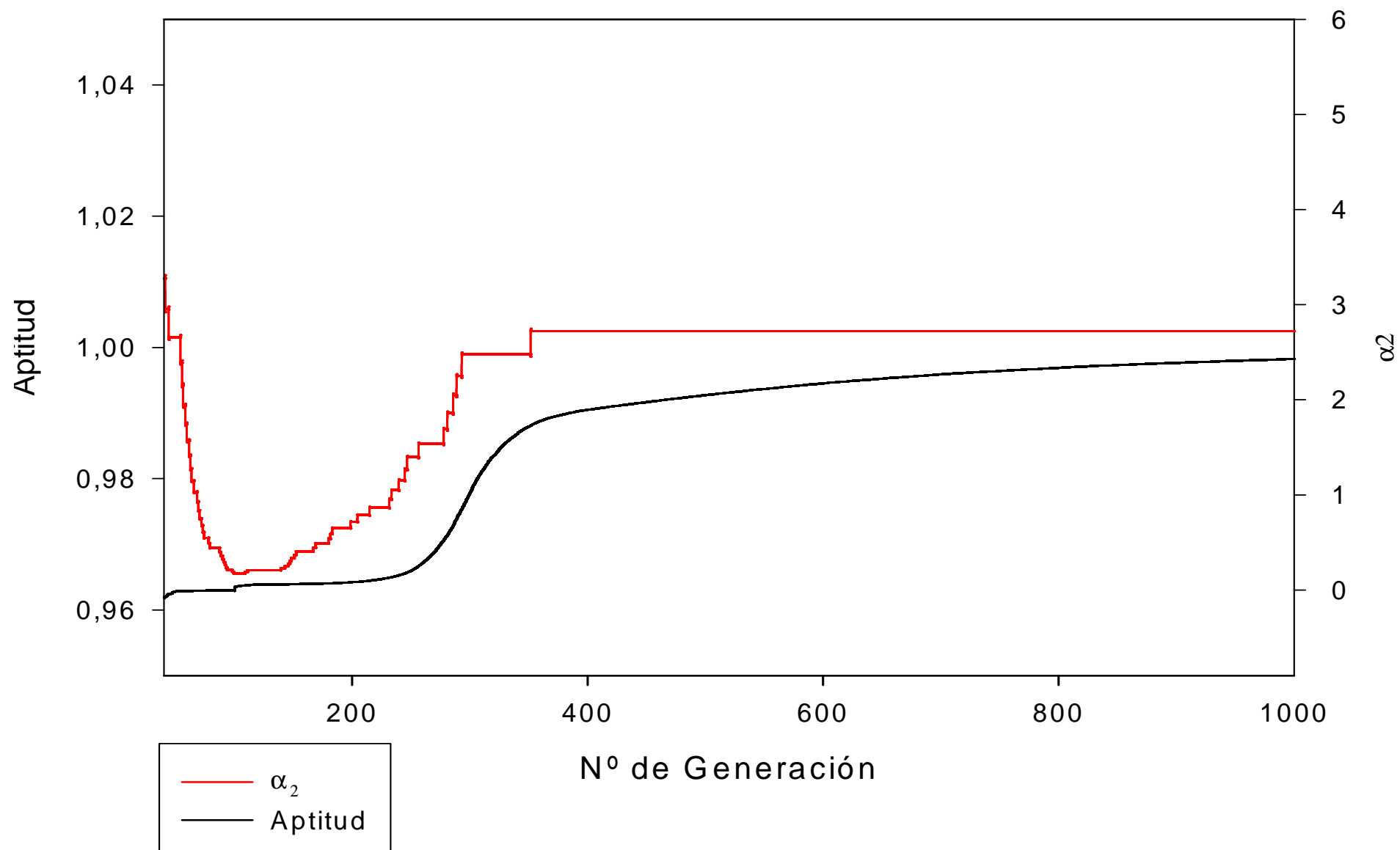
- Si el mejor individuo mejora durante 10 generaciones seguidas se aumenta el valor de $\alpha_1(t), \alpha_2(t)$
- Si el mejor individuo no mejora durante 10 generaciones seguidas se disminuye el valor de $\alpha_1(t), \alpha_2(t)$

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

- Zona Prometedora => Se aumenta el grado de mutación
- Zona no Prometedora => Se disminuye el grado de mutación



Evolución de α_1, α_2



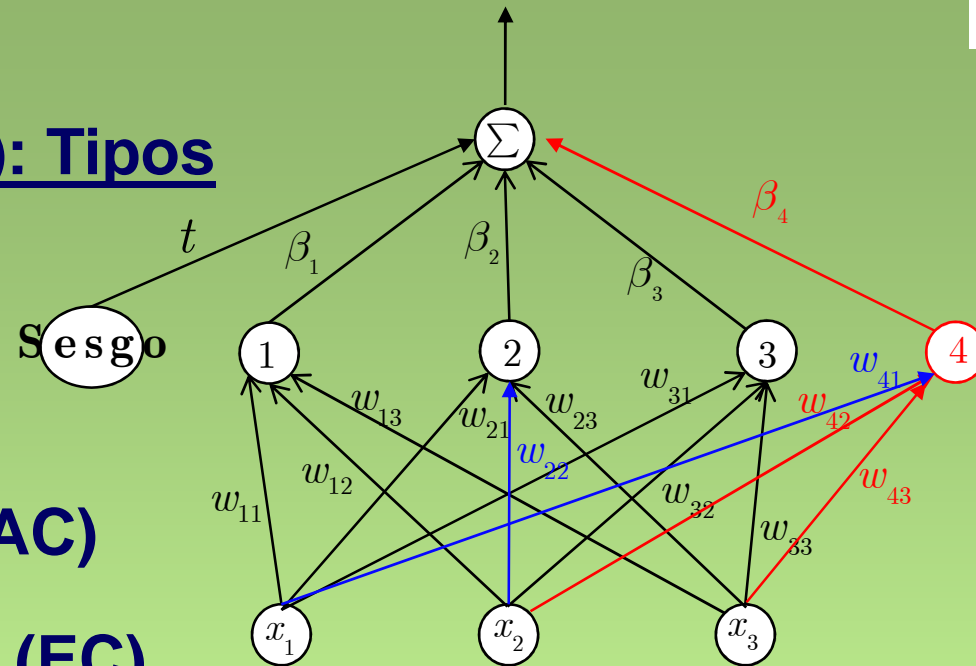


Mutación estructural



(Ejecución secuencial): Tipos

- Añadir nodos (AN)
- Eliminar nodos (EN)
- Añadir conexiones (AC)
- Eliminar conexiones (EC)
- Unir nodos (UN)



$$\Delta_{MIN} + \lfloor u T(g)(\Delta_{MAX} - \Delta_{MIN}) \rfloor$$

siendo u una v.a. $\in U(0, 1)$

	AN	EN	AC	EC
Δ_{MIN}	1	1	1	1
Δ_{MAX}	2	3	?	?



Condiciones de parada



- ❑ Cuando no mejora ni el mejor individuo de la población ni la media de los 10% mejores durante 20 generaciones consecutivas.
- ❑ Cuando los valores de α_1, α_2 estén cercanos a cero.



Primer Experimento: Datos Simulados



- ❑ Datos simulados del conjunto de entrenamiento.
- ❑ Se introduce un 1% o un 2% de ruido Gaussiano a los patrones
- ❑ Datos igualmente simulados para el conjunto de generalización

Nº de patrones: 100

Rango de los Exponentes: $[-5,5]$

Rango de los Coeficientes: $[-10,10]$

Nº de redes en la población : 1000



Resultados. Una Variable

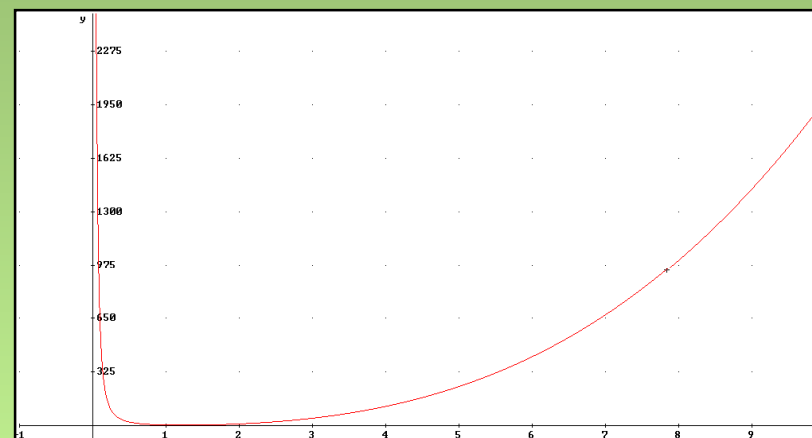
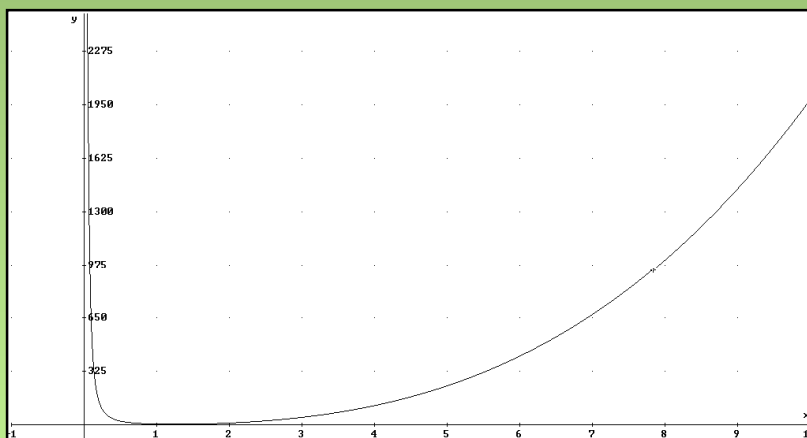


Objetivo

$$f(x) = 2x^3 + x\sqrt{x} - 5x + \frac{6}{x^2}$$

Resultado

$$f_p(x) = 2.0433x^{2.9933} - \frac{4.0219}{x^{2.2494}} - 3.8345x^{0.9159} + \frac{9.7519}{x^{2.1309}}$$

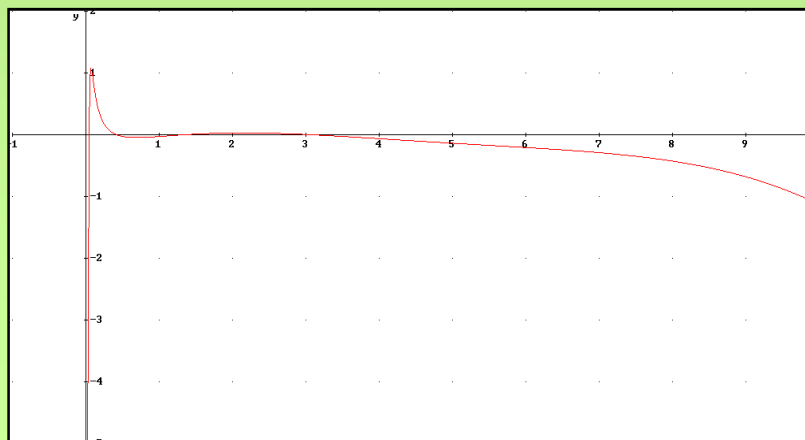


Nº de Generaciones: 2181

R²=0,9998

Error de los datos= 1%

Suma de Residuos=679



$f(x) - f_p(x)$

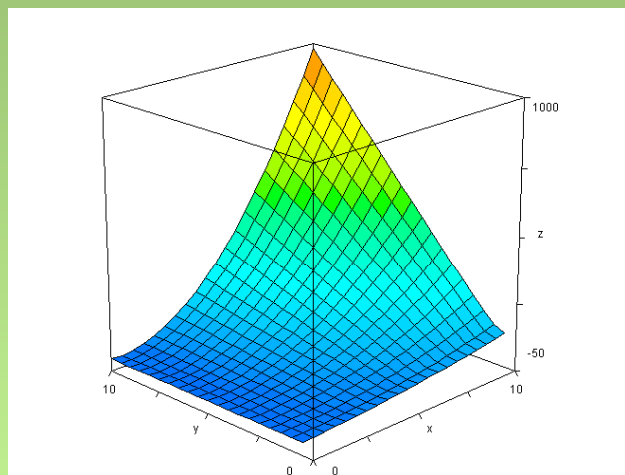


Resultados. Dos Variables



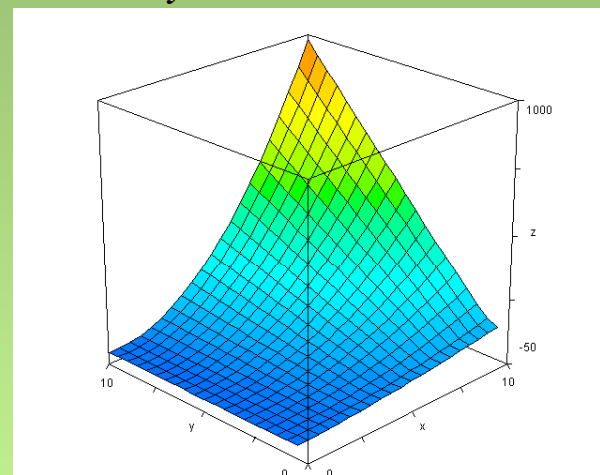
Objetivo

$$f(x, y) = -2\sqrt{xy} + x^2y + \frac{x}{y^2}$$

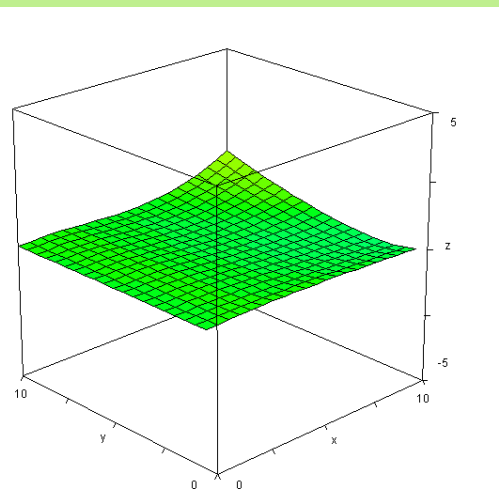


Resultado

$$f_p(x, y) = -2,0011x^{0,521}y^{0,5099} + 1,0161x^{1,9954}y^{0,9977} \\ + 0,9852x^{1,0048}y^{-2,0033}$$



Nº de Generaciones: 2000
R²=0,9995
Error de los datos= 1%
Suma de Residuos=50,09





Resultados. Tres Variables



Objetivo

$$f(x, y, z) = \frac{x^2 y}{z^2} - 3 \frac{zy^3}{x^2} + 2\sqrt{xyz} \quad f_p(x, y, z) = \frac{1,3313x^{1,9385}y^{0,9782}}{z^{2,0195}} - \frac{0,3884x^{1,6294}y^{0,8114}}{z^{2,1265}} - \frac{2,9462y^{3,0123}z^{0,9954}}{x^{2,0037}} + 1,9054x^{0,5513}y^{0,4816}z^{0,4765}$$

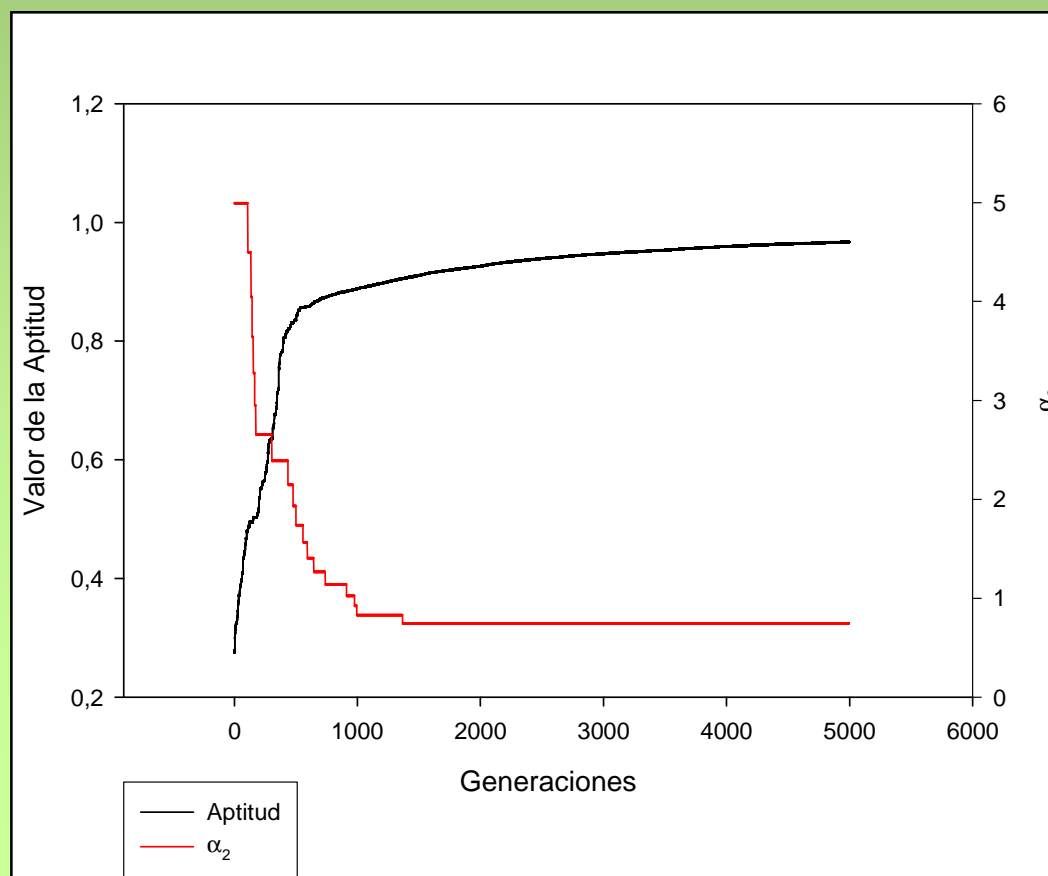
Resultado

Nº de Generaciones: 5000

$R^2=0,999$

Error de los datos= 1%

Suma de Residuos=138699





Aplicaciones.

**Modelos de Crecimiento
Microbiano**



Microbiología Predictiva



Conocimiento detallado de la respuesta de crecimiento de los microorganismos en los alimentos frente a los factores ambientales que les afectan y a partir de los datos resultantes, predecir lo que sucederá durante el almacenamiento, distribución, etc.



Estimar la vida
comercial y la
seguridad
microbiológica



Implantar medidas
para limitar el
desarrollo microbiano

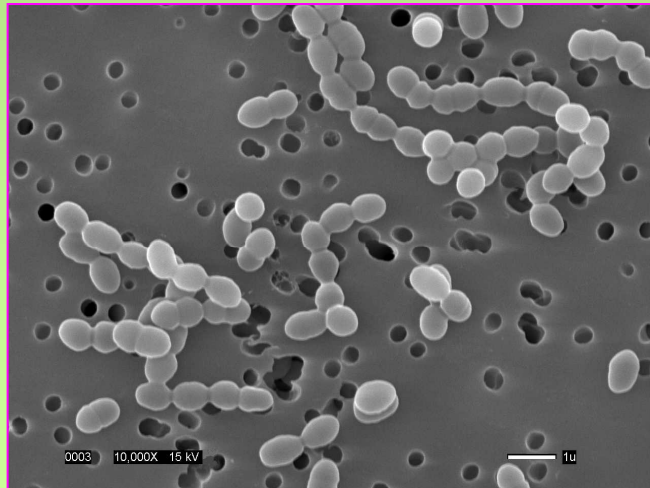


Microbiología Predictiva



Leuconostoc mesenteroides

- ➔ Microorganismo responsable de alteración de diferentes alimentos
- ➔ Ampliamente distribuido: plantas, lácteos y productos alimenticios
- ➔ Incluido en Bacterias ácido lácticas (BAL)



- Gram +
- Coccoide (agar)/ Bacilar (caldo)
- Anaerobio facultativo
- No móvil
- No esporas
- Rango de T^a: 10-37°C



Variables independientes

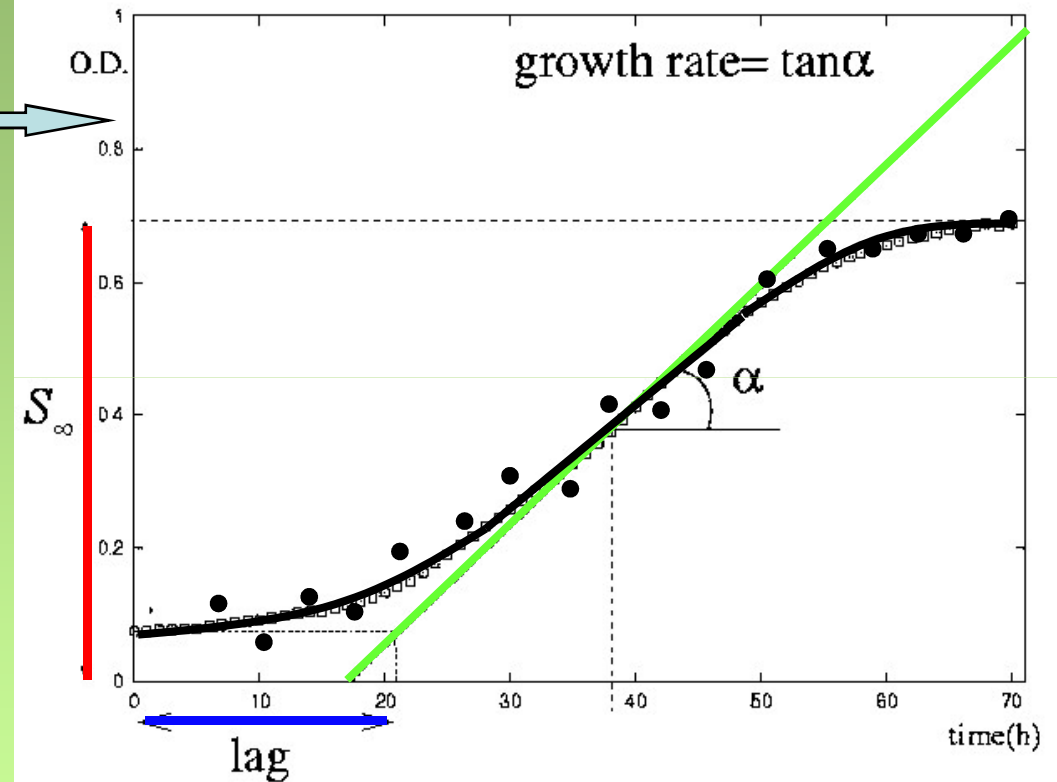
- Temperatura
- Concentración NaNO_2
- Concentración de NaCl
- pH

Modelo
Primario

Modelo
Secundario

Variables dependientes

- Lag
- Growth rate
- Y_{end}



Ecuación Baranyi y Roberts

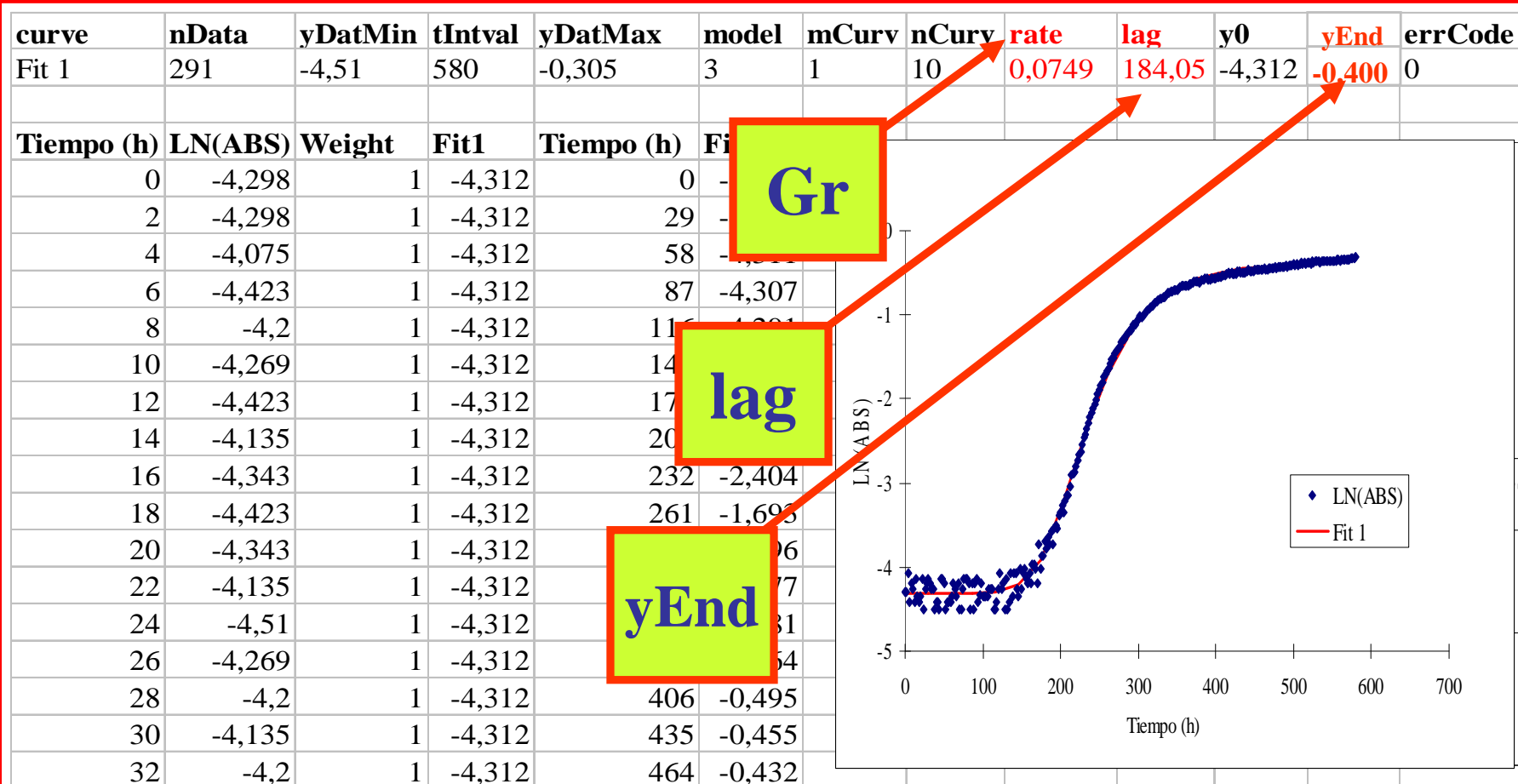


DMFit

Baranyi y Roberts, 1994

LN (ABS) vs tiempo

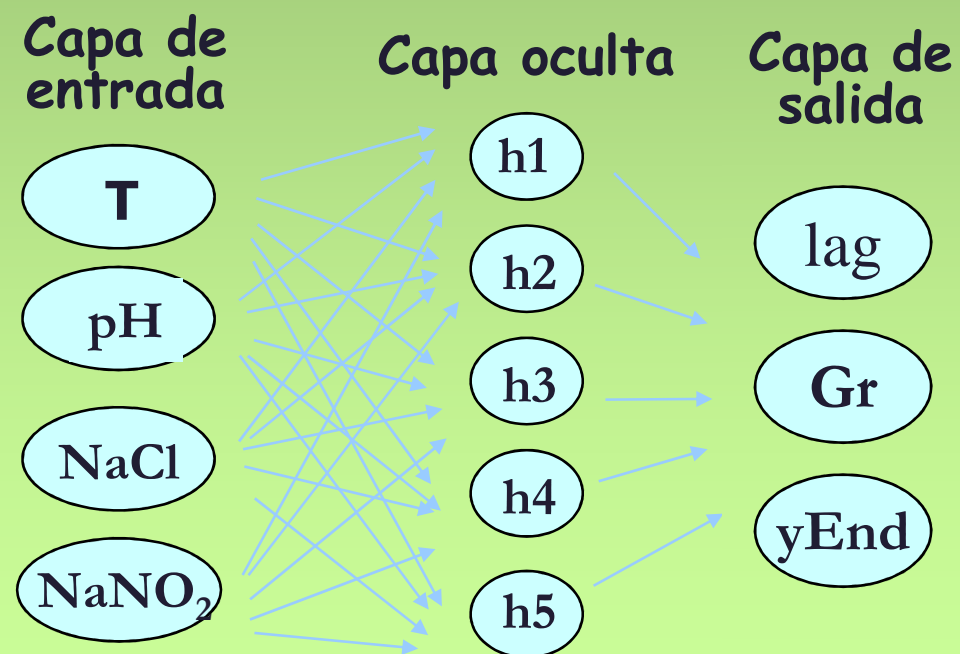
MODELO PRIMARIO





MODELOS SECUNDARIOS

Modelo de Redes Neuronales Artificiales



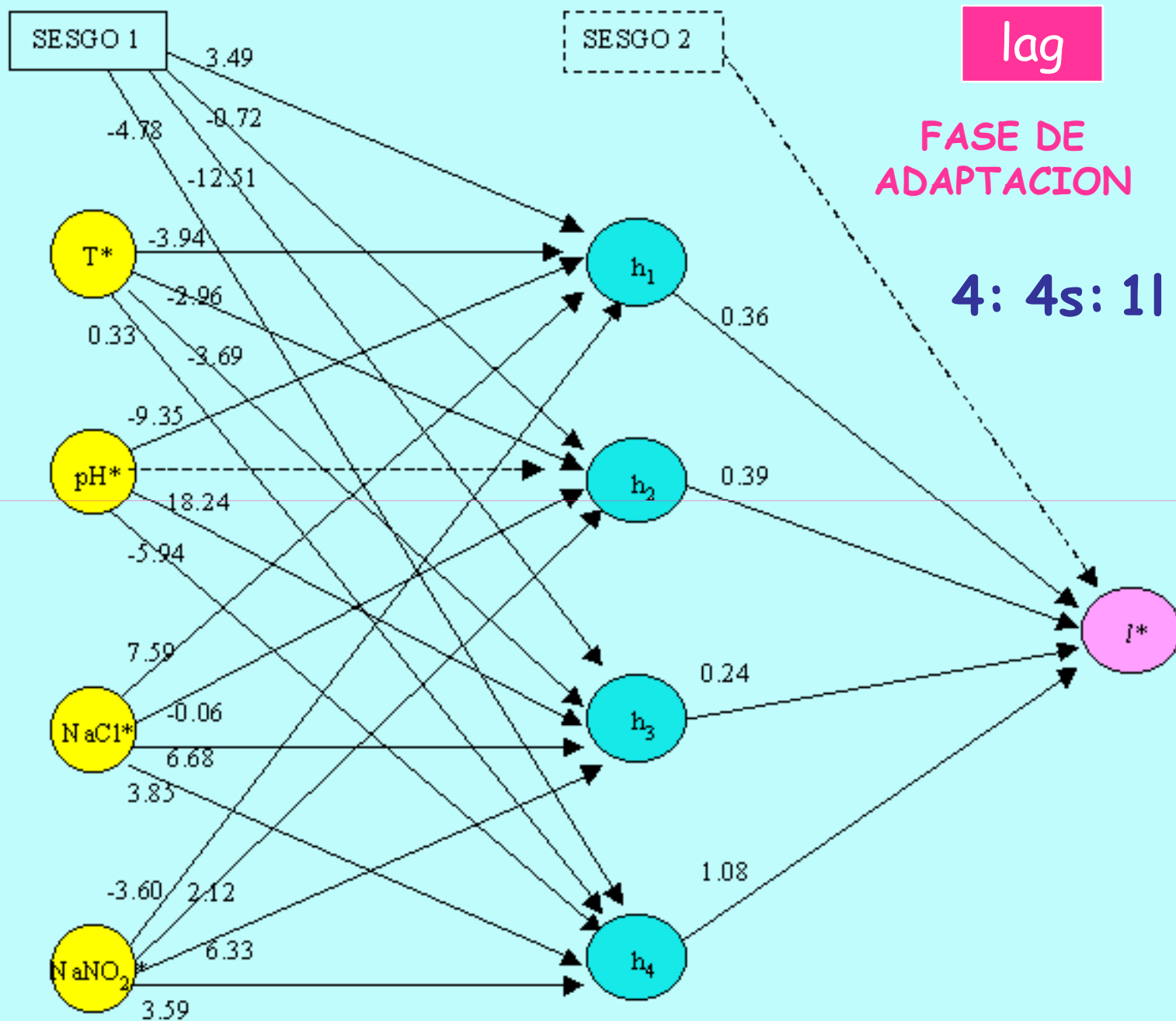


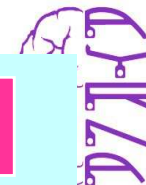
Metodología



- ❑ Se utilizaron 150 curvas para entrenar y 60 para generalizar
- ❑ Las variables independientes se normalizan en el intervalo [0.1,1.1]
- ❑ Las variables dependientes se normalizan en el intervalo [1,2]
- ❑ Comparación con algoritmo genético con redes de unidades sigmoide
- ❑ Misma topología en redes UP y US 4:4:1
- ❑ Se ejecutó el algoritmo 30 veces consecutivas
- ❑ Medida de comparación. SEP

$$SEP = \frac{100}{|\bar{y}|} \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}}$$

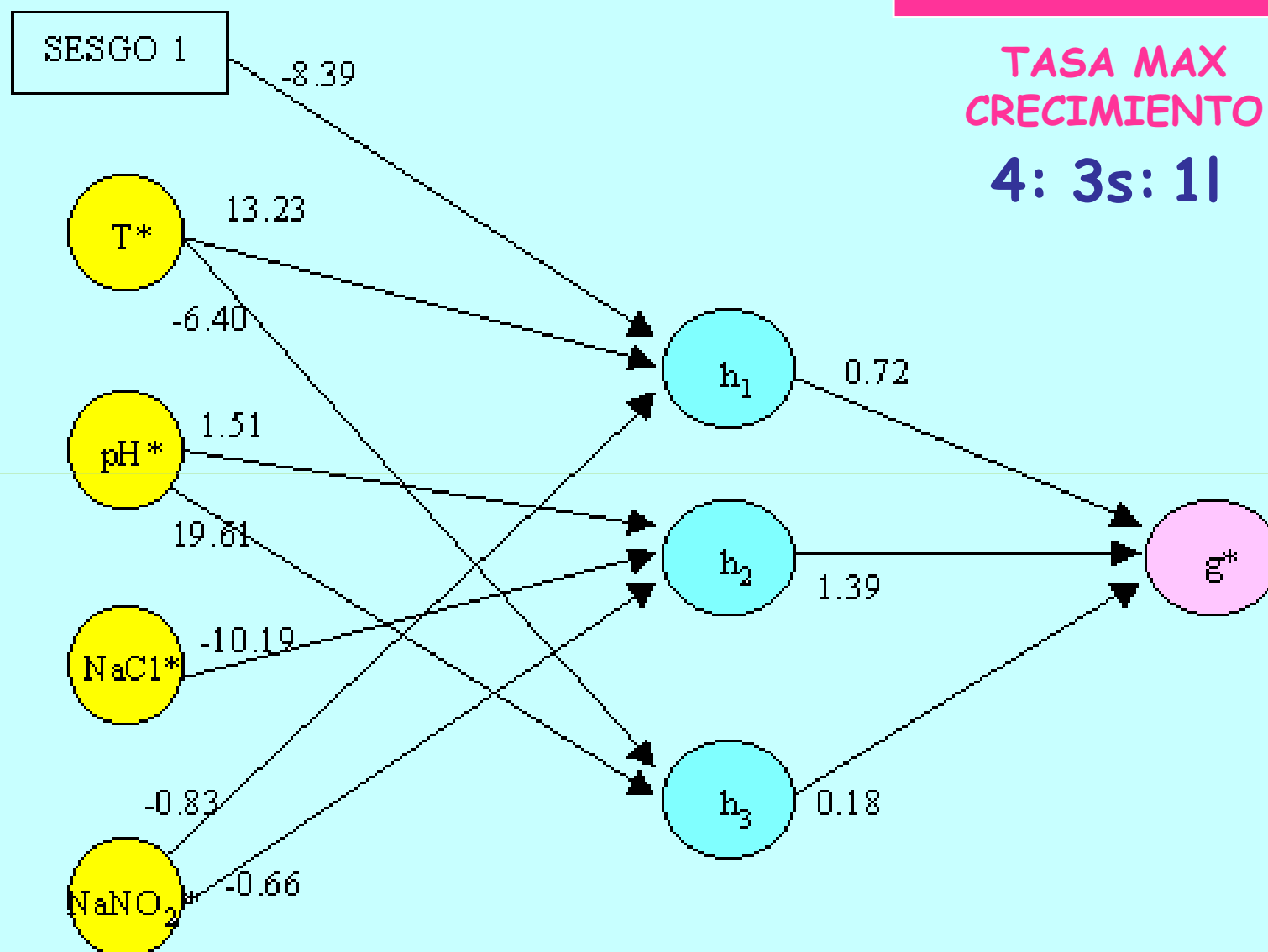


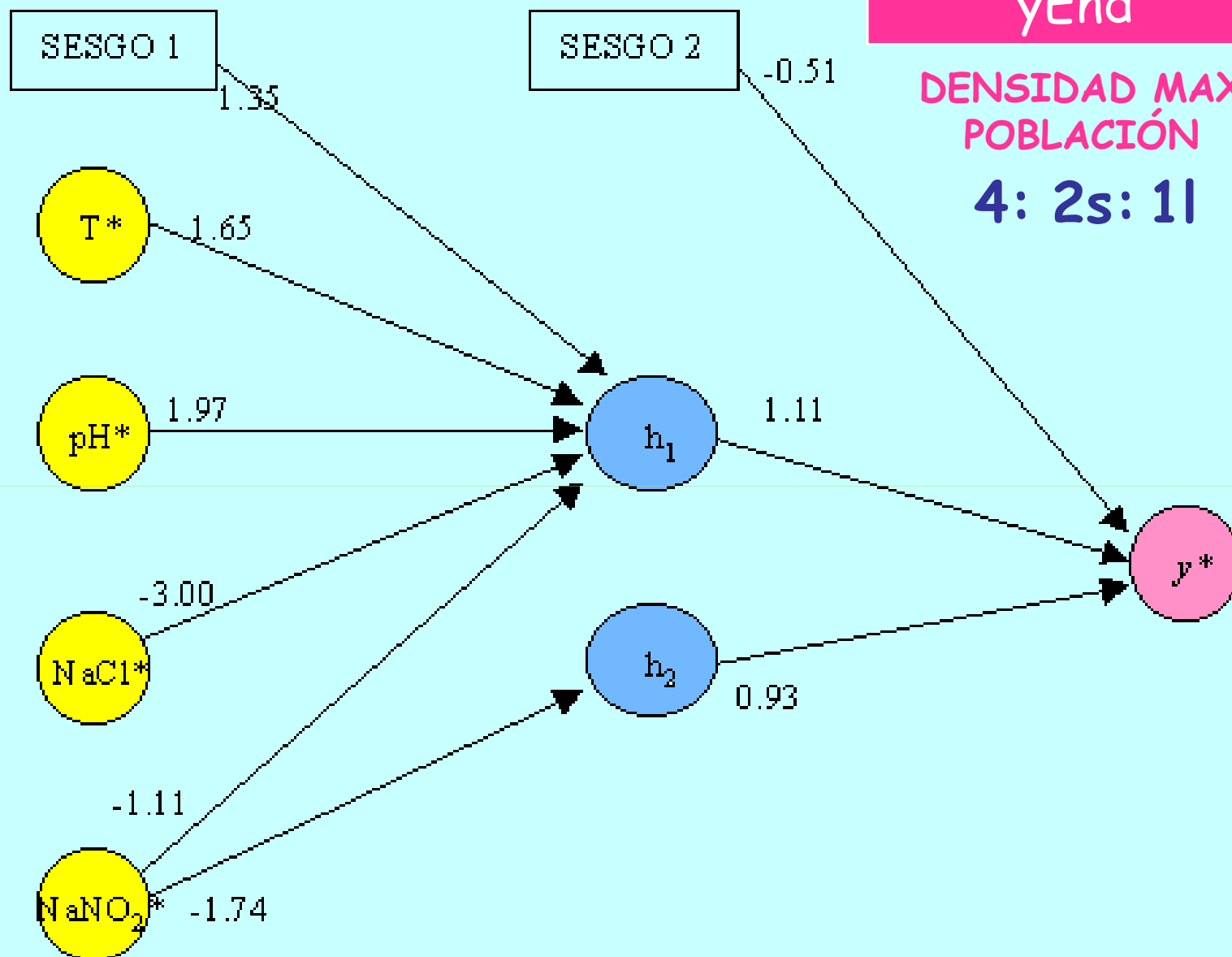
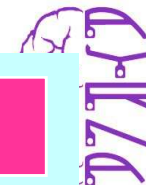


Gr

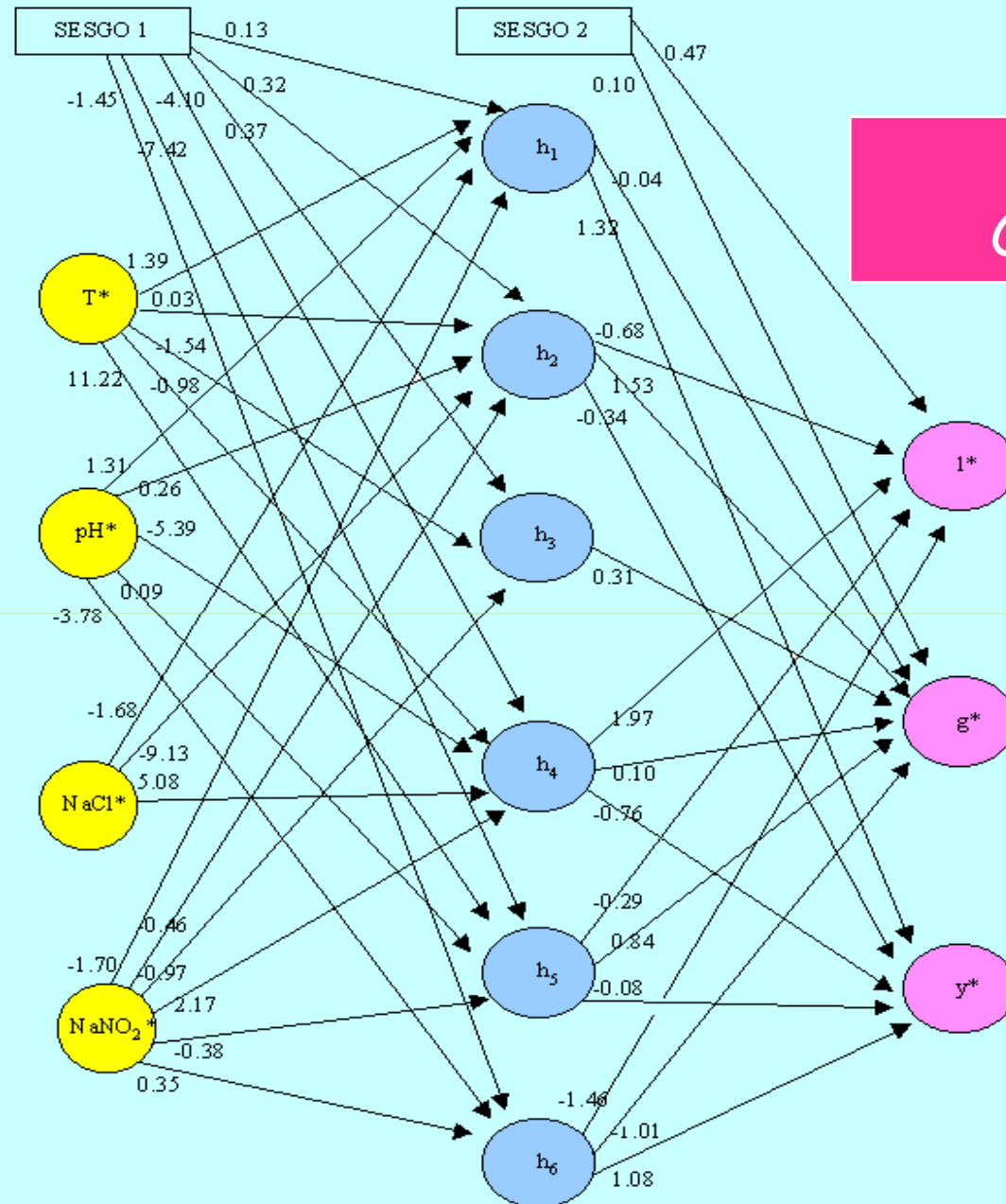
TASA MAX
CRECIMIENTO

4: 3s: 1l





MODELO CONJUNTO

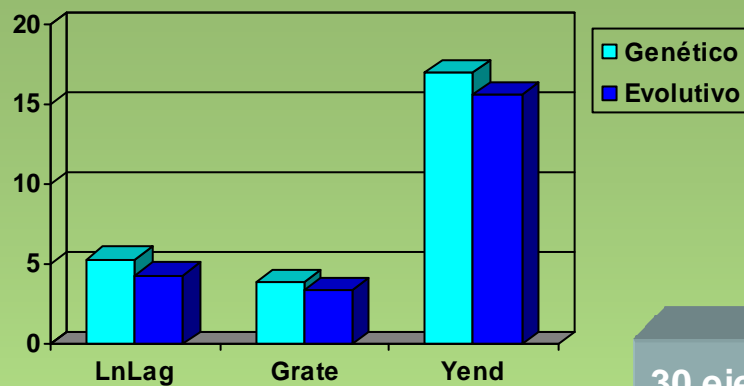




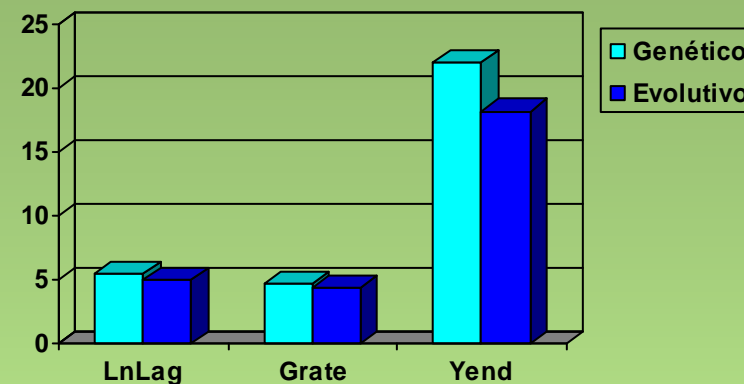
Crecimiento Microbiano. Resultados (I)



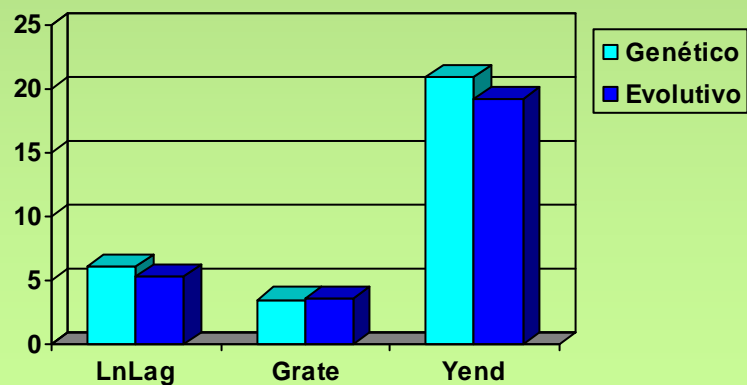
SEP de Entrenamiento Mejor Modelo



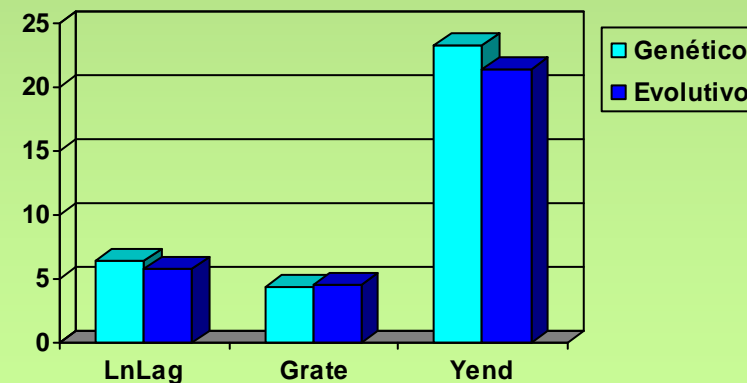
SEP de Entrenamiento Medio



SEP de Generalización Mejor Modelo



SEP de Generalización Medio





Crecimiento Microbiano. Ejemplo de interpretación de modelos



$$Grate^* = 2.9118(T^*)^{1.912}$$

~~$$+ 6.0509(T^*)^{6.548}(NaCl^*)^{5.246}$$~~

~~$$+ 1.8178(T^*)^{0.167}(pH^*)^{0.127}(NaCl^*)^{0.112}$$~~

~~$$- 0.2551(T^*)^{0.107}(pH^*)^{1.548}(NaCl^*)^{0.398}$$~~

~~$$- 4.3718(T^*)^{1.817}(pH^*)^{0.555}(NaNO_2^*)^{0.042}$$~~

$$pH^* \leq 0.1$$



$$Grate^* = 2.9118(T^*)^{1.912}$$

$$+ 6.0509(T^*)^{6.548}(NaCl^*)^{5.246}$$

$$+ 1.8178(T^*)^{0.167}(pH^*)^{0.127}(NaCl^*)^{0.112}$$

$$- 4.3718(T^*)^{1.817}(pH^*)^{0.555}(NaNO_2^*)^{0.042}$$

Término Básico

$$T^* \leq 0.35 \wedge NaCl^* \leq 0.35$$



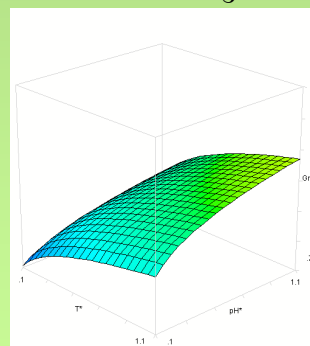
$$Grate^* = 2.9118(T^*)^{1.912}$$

$$+ 1.8178(T^*)^{0.167}(pH^*)^{0.127}(NaCl^*)^{0.112}$$

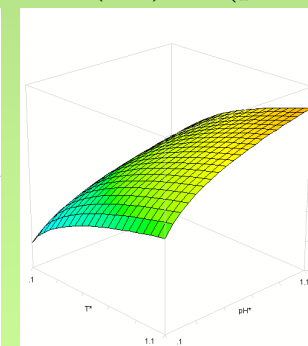
$$- 0.2551(T^*)^{0.107}(pH^*)^{1.548}(NaCl^*)^{0.398}$$

$$- 4.3718(T^*)^{1.817}(pH^*)^{0.555}(NaNO_2^*)^{0.042}$$

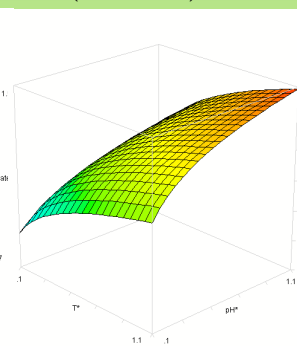
$$S_3 = 1.8178(T^*)^{0.167}(pH^*)^{0.127}(NaCl^*)^{0.112}$$



(NaCl*=0.1)



(NaCl*=0.6)



(NaCl*=1.1)



Bibliografía



(Bishop; 1995) Bishop, M., 1995. Neural networks for pattern recognition. Oxford University Press, UK.

(Durbin and Rumelhart; 1989) Durbin, R., Rumelhart, D., 1989. Products Units: A computationally powerful and biologically plausible extension to backpropagation networks. Neural Computation 1(1), 133-142.



Redes Neuronales Evolutivas: Aplicaciones

GRACIAS POR SU ATENCIÓN

**César Hervás-Martínez
Pedro A. Gutiérrez Peña
Grupo de Investigación AYRNA**

**Departamento de Informática y Análisis
Numérico
Universidad de Córdoba
Campus de Rabanales. Edificio Einstein.
Email: chervas@uco.es
pagutierrez@uco.es**

2021-2022

41