



# **ENTRENAMIENTO DE REDES NEURONALES BASADO EN ALGORITMOS EVOLUTIVOS**

**TESIS DE GRADO EN INGENIERÍA INFORMÁTICA**

**FACULTAD DE INGENIERÍA  
UNIVERSIDAD DE BUENOS AIRES**

**TESISTA: Sr. Luis Federico Bertona**

**DIRECTORES: Prof. Dr. Ramón García Martínez**

**Prof. M. Ing. Paola Britos**

**Laboratorio de Sistemas Inteligentes**

**NOVIEMBRE 2005**



*ENTRENAMIENTO DE REDES NEURONALES  
BASADO EN ALGORITMOS EVOLUTIVOS*

**TESIS DE GRADO EN INGENIERÍA INFORMÁTICA**

Laboratorio de Sistemas Inteligentes  
FACULTAD DE INGENIERÍA  
UNIVERSIDAD DE BUENOS AIRES

**Sr. Luis Federico Bertona**

**Tesista**

**Dr. Ramón García Martínez**

**M. Ing. Paola Britos**

**Directores**

**NOVIEMBRE 2005**



## Resumen

En este trabajo se estudia un método alternativo para el entrenamiento de redes neuronales con conexión hacia delante. Una vez determinada la topología de la red neuronal se utiliza un algoritmo genético para ajustar los pesos de la red neuronal. Se evalúan diferentes variantes de los operadores genéticos para el entrenamiento de las redes neuronales. Los resultados obtenidos por el algoritmo genético son contrastados con los resultados obtenidos por el algoritmo de retropropagación de errores.

**Palabras claves:** Computación evolutiva, redes neuronales, algoritmos genéticos, entrenamiento, retropropagación.

## Abstract

In this work an alternative method to train feedforward neural networks is studied. Once the topology of the neural network is determined, a genetic algorithm is used to adjust the neural network's weights. Different variations of the genetic operators to train neural networks are evaluated. The results reached by the genetic algorithm are contrasted with the results reached by the Backpropagation Algorithm.

**Keywords:** Evolutionary computation, neural networks, genetic algorithms, train, Backpropagation.



## Índice de contenidos

Capítulo 1: Introducción .....	1
1.2. Organización de esta tesis.....	2
Capítulo 2: Redes neuronales .....	5
2.1. El modelo biológico.....	5
2.2. Estructura de un sistema neuronal artificial .....	6
2.3. Modelo de neurona artificial.....	7
2.3.1. Entradas y salidas .....	8
2.3.2. Pesos sinápticos .....	8
2.3.3. Regla de propagación .....	8
2.3.4. Función de activación.....	9
2.3.5. Función de salida .....	9
2.4. Arquitectura de una red neuronal .....	10
2.5. Aprendizaje.....	10
2.5.1. Fase de entrenamiento. ....	10
2.5.2. Fase de operación.....	11
2.6. Redes neuronales con conexión hacia delante .....	11
2.6.1. Perceptrón.....	12
2.6.2. ADALINE / MADALINE.....	12
2.6.3. Perceptrón multicapa .....	13
Capítulo 3: Entrenamiento de redes neuronales .....	15
3.1. Métodos de gradiente descendente.....	15
3.2. El algoritmo Backpropagation .....	16
3.2.1. Deducción del algoritmo Backpropagation .....	16
3.2.2. Modos de entrenamiento.....	19
3.2.3. Aceleración del aprendizaje .....	20
3.3. Generalización.....	22
Capítulo 4: Algoritmos genéticos.....	25
4.1. Introducción .....	25
4.2. El cromosoma.....	26
4.3. Evaluación y aptitud.....	27
4.4. Población .....	27
4.5. Variantes de los operadores básicos .....	28
4.5.1. Selección.....	28
4.5.2. Cruza.....	30
4.5.3. Mutación.....	31
Capítulo 5: Descripción del problema.....	33
5.1. Deficiencias del algoritmo .....	33
5.2. Objetivo y marco de trabajo .....	35
Capítulo 6: Solución propuesta .....	37
6.1. Algoritmo genético para entrenamiento de redes neuronales.....	38
6.1.1. Codificación de parámetros.....	38
6.1.2. Del genotipo al fenotipo.....	38
6.1.3. Evaluación y aptitud .....	38
6.1.4. Operadores de selección .....	39
6.1.5. Operadores de cruza.....	41
6.1.6. Operadores de mutación.....	43
6.1.7. Operadores de inicialización .....	45
6.2. Problemas del método propuesto .....	46

6.2.1. El problema de la convergencia prematura .....	46
6.2.2. El problema de la permutación.....	47
6.3. Cuestiones a resolver en esta tesis .....	48
Capítulo 7: Experimentación .....	49
7.1. Diseño experimental .....	49
7.1.1. El algoritmo backpropagation .....	50
7.1.2. El algoritmo genético .....	51
7.1.3. El set de entrenamiento .....	51
7.1.4. Evaluación del método .....	52
7.1.5. Parámetros de los algoritmos.....	53
7.1.6. Tipos de gráficos .....	54
7.2. Experimentos .....	55
7.2.1. Red Neuronal .....	55
7.2.2. Algoritmos genéticos .....	60
7.2.3. Comparación de ambos métodos.....	72
Capítulo 8: Conclusiones y limitaciones .....	77
Referencias .....	83
Apéndice A: Datos Experimentales .....	89
Apéndice B: Metodología de software .....	197
B.1. Requerimientos .....	197
B.2. Análisis .....	197
B.2.1. Seleccionar topología .....	197
B.2.2. Seleccionar tipo de entrenamiento.....	198
B.2.3. Configurar algoritmo de entrenamiento.....	198
B.2.4. Cargar ejemplos de entrenamiento.....	198
B.2.5. Cargar ejemplos de evaluación .....	199
B.2.6. Entrenar red neuronal .....	199
B.3. Diseño .....	200
B.3.1. Plataforma .....	200
B.3.2. Archivos y bloques constitutivos del sistema .....	201
B.3.3. Estructuras de datos .....	210
B.3.4. Formato de los archivos de estadísticas .....	212
B.4. Implementación .....	213
B.4.1. Pantallas .....	213
B.4.2. Funciones implementadas .....	214
B.4.3. Pruebas .....	225
B.5. Mantenimiento .....	239
B.5.1. Gestión de configuración .....	239



## Capítulo 1: Introducción

La Inteligencia Artificial es la disciplina que estudia la forma de diseñar procesos que exhiban características que comúnmente se asocian con el comportamiento humano inteligente **[García Martínez, 1997]**. La Inteligencia Artificial sintetiza y automatiza tareas intelectuales y es, por lo tanto, potencialmente relevante para cualquier ámbito de la actividad intelectual humana **[Russell y Norving, 2004]**. Actualmente esta ciencia está comprendida por varios subcampos que van desde áreas de propósito general, como el aprendizaje y la percepción, a otras más específicas como la demostración de teoremas matemáticos, el diagnóstico de enfermedades, etc.

Uno de los modelos que ha surgido para emular el proceso de aprendizaje es la red neuronal artificial. Las redes neuronales son modelos que intentan reproducir el comportamiento del cerebro humano **[Hilera y Martínez, 1995]**. Una red neuronal consiste en un conjunto de elementos de procesamiento, llamados neuronas, los cuales se conectan entre sí **[Koehn, 1994]**. La organización y disposición de las neuronas dentro de una red neuronal se denomina topología, y viene dada por el número de capas, la cantidad de neuronas por capa, el grado de conectividad, y el tipo de conexión entre neuronas.

Una vez determinada la topología de la red neuronal es necesario entrenarla. En la etapa de entrenamiento la red es capaz de aprender relaciones complejas entre entradas y salidas mediante el ajuste de los pesos de las conexiones entre neuronas.

Widrow y Lehr **[Lehr y Widrow, 1990]** identifican una cantidad significativa de algoritmos de entrenamiento. La mayoría de éstos utilizan información del gradiente de una función de error para ajustar los pesos de las conexiones, y se los llaman algoritmos de gradiente descendente **[Porto, 1998]**.

Las redes neuronales artificiales han sido aplicadas con éxito en gran cantidad de problemas como por ejemplo reconocimiento de patrones, clasificación, visión, control, predicción, etc. **[Zilouchian, 2001]**.

Sin embargo, los algoritmos de gradiente descendente poseen dos problemas. Primero, suelen quedar atrapados en mínimos locales, generándose de esta manera estimaciones subóptimas de los pesos. Segundo, suelen ser muy lentos por utilizar pasos infinitesimales para alcanzar la solución.

Los algoritmos evolutivos, dentro de los cuales los algoritmos genéticos son los más conocidos, son una familia de modelos computacionales inspirados en la evolución y la supervivencia del más apto **[Bäch, et. al., 1991; Ömer, 1995; Whitley, 2001]**. Se utilizan fundamentalmente en la resolución de problemas de búsqueda y de optimización **[Holland, 1975]**. El verdadero poder de estos algoritmos radica en la búsqueda simultánea de la solución en un conjunto de posibles soluciones (individuos). Buscan una solución del problema reproduciendo genéticamente una población de individuos a lo largo de una serie de generaciones **[Koza, 1997]**.

En el contexto de los algoritmos evolutivos, el aprendizaje es formulado como un problema de optimización. Dada una topología fija, el entrenamiento de una red neuronal puede ser visto como un proceso de optimización cuyo

objetivo es encontrar un conjunto de pesos que minimice el error que produce la red sobre el conjunto de datos de entrenamiento. Esto convierte a los algoritmos evolutivos en una alternativa a los métodos de entrenamiento tradicionales, ya que aquellos son capaces de no quedar atrapados en mínimos locales y, al avanzar hacia la solución mediante operadores genéticos, pueden aumentar la velocidad de convergencia a la solución.

El objetivo de este proyecto es combinar dos técnicas utilizadas en la construcción de sistemas inteligentes, los algoritmos evolutivos y las redes neuronales, para crear un sistema donde, tras definir una topología de red neuronal, los primeros sean utilizados para entrenar a las segundas, en reemplazo de los tradicionales algoritmos de entrenamiento. Se contrastará el entrenamiento de redes neuronales basado en algoritmos genéticos con el entrenamiento basado en métodos convencionales actualmente en uso.

### **1.2. Organización de esta tesis**

El resto de este trabajo se encuentra organizado de la siguiente manera:

- Los capítulos 2 a 4 son capítulos introductorios, y se presentan las tecnologías utilizadas a lo largo de esta tesis. Luego, en los capítulos restantes, se realiza una presentación del problema, la solución propuesta y los resultados y conclusiones obtenidas a lo largo de los experimentos.
- En el Capítulo 2 se presentan las redes neuronales. Se introducen los conceptos básicos de ésta técnica y se realiza una descripción detallada del modo de funcionamiento de las mismas.
- El Capítulo 3 está dedicado al estudio de los métodos de aprendizaje de las redes neuronales. Se realiza una introducción a los métodos de entrenamiento y se desarrolla en detalle el algoritmo Backpropagation. Se exponen los distintos modos de entrenamiento y se enumeran las distintas técnicas para acelerar el entrenamiento de una red neuronal. Se introduce el concepto de generalización.
- En el Capítulo 4 se introducen los conceptos generales de los algoritmos genéticos. Se describe en detalle las características de esta técnica y se realiza una descripción pormenorizada de los operadores genéticos. Esta descripción sirve como base para entender las características de la solución propuesta.
- En el Capítulo 5 se enumeran las deficiencias que tiene el algoritmo de entrenamiento backpropagation y que llevan a la necesidad de buscar una forma alternativa y complementaria de entrenar redes neuronales. Se plantean los objetivos de diseño del método alternativo.
- En el Capítulo 6 se describen las características principales que convierten a los algoritmos genéticos como una alternativa interesante para entrenar redes neuronales. Se presentan las características de diseño del método alternativo y se proponen algunas variantes de los operadores genéticos para entrenar redes neuronales.

- El Capítulo 7 está dividido en dos partes. En la primera de ellas se realiza una presentación del diseño experimental. Se describen las características de los experimentos y la manera en que se analizarán los resultados. La segunda parte de dicho capítulo esta dedicada a la presentación de los resultados y al análisis de los mismos. Se muestran los principales resultados obtenidos a lo largo de cada experimento de manera gráfica y se realiza una descripción detallada de los mismos.
- En el Capítulo 8 se retoman todas las cuestiones surgidas a lo largo de este trabajo y se responden a las mismas en base a los resultados experimentales y/o los fundamentos teóricos introducidos. Contiene las conclusiones finales de este trabajo. También en este capítulo se identifican las limitaciones del trabajo y se plantean futuras líneas de investigación.
- En el Apéndice A se despliegan los datos numéricos obtenidos a lo largo de la etapa de experimentación.
- En el Apéndice B se introduce la metodología de software aplicada al desarrollo realizado en esta tesis. En el se presentan los conceptos fundamentales que se utilizaron para diseñar y desarrollar el sistema utilizado en esta tesis.



## Capítulo 2: Redes neuronales

Las redes neuronales artificiales (RNA) son modelos matemáticos que intentan reproducir el funcionamiento del sistema nervioso. Como todo modelo, realizan una simplificación del sistema real que simulan y toman las características principales del mismo para la resolución de una tarea determinada.

### 2.1. El modelo biológico

El cerebro es el elemento principal del sistema nervioso humano y está compuesto por un tipo especial de célula llamada neurona. Una neurona es una célula viva y como tal posee todos los elementos comunes de las células biológicas. A su vez, las neuronas tienen características propias que le permiten comunicarse entre ellas, lo que las diferencia del resto de las células biológicas. La figura 2.1 muestra la estructura típica de una neurona biológica.

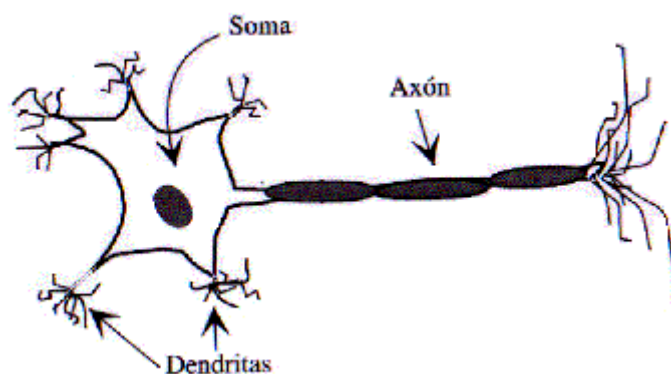


Figura 2.1. Neurona biológica

De la figura se observa que la neurona biológica está compuesta por un *cuerpo celular* o *soma*, del cual se desprende árbol de ramificaciones llamado *árbol dendrítico*, compuesto por las *dendritas*. Del soma también parte una fibra tubular, llamada *axón*, el cual suele ramificarse cerca de su extremo. Las dendritas actúan como un canal de entrada de señales provenientes desde el exterior hacia la neurona, mientras que el axón actúa como un canal de salida. El espacio entre dos neuronas vecinas se denomina *sinapsis*. En el córtex cerebral se observa una organización horizontal en capas, así como también una organización vertical en columnas de neuronas.

La intensidad de una sinapsis no es fija, sino que puede ser modificada en base a la información proveniente del medio. De esta manera la estructura del cerebro no permanece fija sino que se va modificando por la formación de nuevas conexiones, ya sean excitadoras o inhibitorias, la destrucción de conexiones, la modificación de la intensidad de la sinapsis, o incluso por muerte neuronal.

Desde un punto de vista funcional, las neuronas conforman un procesador de información sencillo. Constan de un subsistema de entrada (dendritas), un subsistema de procesamiento (el soma) y un subsistema de salida (axón)

Como se menciono antes, una de las características principales de las neuronas, y que la distinguen del resto de las células, es su capacidad de comunicarse. Las señales nerviosas pueden ser eléctricas o químicas. La transmisión química se da principalmente en la comunicación entre neuronas, mientras que la eléctrica se produce dentro de una neurona **[García Martínez, et al, 2003]**. En general, una neurona recibe información de cientos de neuronas vecinas y la transmite a otras tantas neuronas.

La comunicación entre neuronas se lleva a cabo de la siguiente manera: en el soma de las neuronas transmisoras o presinápticas se genera un pulso eléctrico llamado potencial de acción. El pulso eléctrico se propaga a través del axón en dirección a las sinapsis. La información se transmite a las neuronas vecinas utilizando un proceso químico, mediante la liberación de neurotransmisores. Estos neurotransmisores se transmiten a través de la sinapsis hacia la neurona receptora. La neurona receptora o postsináptica toma la señal enviada por cientos de neuronas a través de las dendritas y la transmite al cuerpo celular. Estas señales pueden ser excitadoras (positivas) o inhibitoras (negativas) **[Gurney, 1997]**. El soma es el encargado de integrar la información proveniente de las distintas neuronas. Si la señal resultante supera un determinado umbral (umbral de disparo) el soma emite un pulso que se transmite a lo largo del axón dando lugar a la transmisión eléctrica a lo largo de la neurona. Al llegar la señal al extremo del axón se liberan neurotransmisores que permiten transmitir la señal a las neuronas vecinas. **[Nascimento, 1994]**.

## ***2.2. Estructura de un sistema neuronal artificial***

Como se dijo anteriormente, las redes neuronales son modelos matemáticos que intentan reproducir el comportamiento del cerebro humano. El principal objetivo de este modelo es la construcción de sistemas capaces de presentar un cierto comportamiento inteligente. Esto implica la capacidad de aprender a realizar una determinada tarea.

Las características principales que reproducen las redes neuronales artificiales se pueden reducir a los siguientes tres conceptos: procesamiento paralelo, distribuido y adaptativo. **[Del Brio y Sanz Molina, 2002]**

El verdadero poder de este modelo radica en el procesamiento paralelo realizado por las neuronas artificiales. La neurona artificial es un elemento de procesamiento simple y constituye el elemento principal de un sistema neuronal artificial.

Estas neuronas artificiales se combinan en estructuras denominadas capas. Una red neuronal artificial esta un compuesta por un conjunto de capas. De esta manera, la información se encuentre distribuida a lo largo de las sinapsis de la red, dándole a este sistema cierta tolerancia a fallos.

A su vez, las redes neuronales artificiales son capaces de adaptar su funcionamiento a distintos entornos modificando sus conexiones entre neuronas. De esta manera pueden aprender de la experiencia y generalizar conceptos.

Por último, un conjunto de redes neuronales, junto con las interfases de entrada y salida, y los módulos lógicos adicionales conforman un sistema neuronal artificial.

### 2.3. Modelo de neurona artificial

La neurona artificial es un elemento de procesamiento simple que a partir de un vector de entradas produce una única salida. En general podemos encontrar tres tipos de neuronas artificiales, donde cada una de las cuales tiene su contraparte en el sistema nervioso:

1. Las que reciben información directamente desde el exterior, a las cuales se las denomina neuronas de entrada.
2. Las que reciben información desde otras neuronas artificiales, a las cuales se las denomina neuronas ocultas. Es en estas neuronas, en particular en sus sinapsis, donde se realiza la representación de la información almacenada.
3. Las que reciben la información procesada y las devuelven al exterior. A estas neuronas se las denomina neuronas de salida.

La figura 2.2 muestra los elementos que componen una neurona artificial:

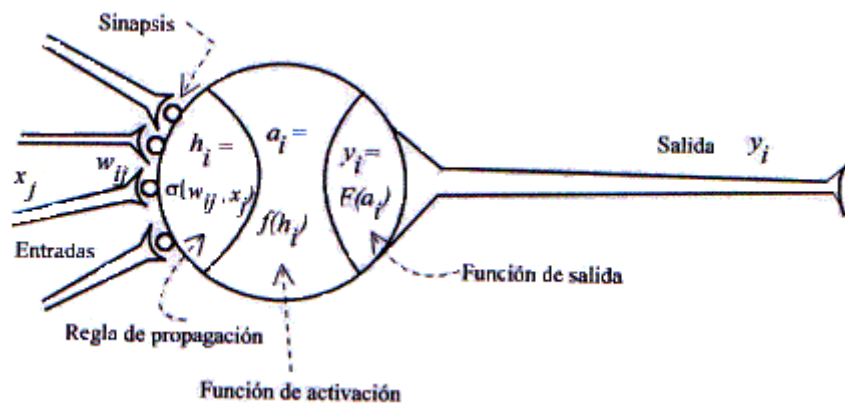


Figura 2.2. Neurona artificial

- **Conjunto de entradas,  $x_j(t)$ .** Estas pueden ser provenientes del exterior o de otras neuronas artificiales.
- **Peso sinápticos,  $w_{ij}$ .** Representan el grado de comunicación entre la neurona artificial  $j$  y la neurona artificial  $i$ . Pueden ser excitadores o inhibidores
- **Regla de propagación,  $\sigma(w_{ij}, x_j(t))$ .** Integra la información proveniente de las distintas neuronas artificiales y proporciona el valor del potencial postsináptico de la neurona  $i$ .
- **Función de activación,  $f_i(a_i(t-1), h_i(t))$ .** Provee el estado de activación actual de la neurona  $i$ .

- **Función de salida,  $F_i(a_i(t))$ .** Representa la salida actual de la neurona  $i$ .

De esta forma, la salida producida por una neurona  $i$ , para un determinado instante de tiempo  $t$  puede ser escrita en forma general de la siguiente manera

$$y_i(t) = F_i(f_i[a_i(t-1), \sigma_i(w_{ij}, x_j(t))]) \quad (2.1)$$

A continuación se estudian cada uno de los puntos introducidos anteriormente.

### 2.3.1. Entradas y salidas

Las entradas y salidas de una neurona pueden ser clasificadas en dos grandes grupos, binarias o continuas. Las neuronas binarias (digitales) sólo admiten dos valores posibles. En general en este tipo de neurona se utilizan los siguientes dos alfabetos  $\{0,1\}$  o  $\{-1,1\}$ . Por su parte, las neuronas continuas (analógicas) admiten valores dentro de un determinado rango, que en general suele definirse como  $[-1, 1]$ .

La selección del tipo de neurona a utilizar depende de la aplicación y del modelo a construir.

### 2.3.2. Pesos sinápticos

El peso sináptico  $w_{ij}$  define la fuerza de una conexión sináptica entre dos neuronas, la neurona presináptica  $i$  y la neurona postsináptica  $j$ . Los pesos sinápticos pueden tomar valores positivos, negativos o cero. En caso de una entrada positiva, un peso positivo actúa como excitador, mientras que un peso negativo actúa como inhibidor. En caso de que el peso sea cero, no existe comunicación entre el par de neuronas.

Mediante el ajuste de los pesos sinápticos la red es capaz de adaptarse a cualquier entorno y realizar una determinada tarea.

### 2.3.3. Regla de propagación

La regla de propagación determina el potencial resultante de la interacción de la neurona  $i$  con las  $N$  neuronas vecinas. El potencial resultante  $h_i$  se puede expresar de la siguiente manera:

$$h_i(t) = \sigma_i(w_{ij}, x_j(t)) \quad (2.2)$$

La regla de propagación más simple y utilizada consiste en realizar una suma de las entradas ponderadas con sus pesos sinápticos correspondientes:



$$h_i(t) = \sum_j w_{ij} * x_j(t) \quad (2.3)$$

### 2.3.4. Función de activación

La función de activación determina el estado de activación actual de la neurona en base al potencial resultante  $h_i$  y al estado de activación anterior de la neurona  $a_i(t-1)$ . El estado de activación de la neurona para un determinado instante de tiempo  $t$  puede ser expresado de la siguiente manera:

$$a_i(t) = f_i(a_i(t-1), h_i(t)) \quad (2.4)$$

Sin embargo, en la mayoría de los modelos se suele ignorar el estado anterior de la neurona, definiéndose el estado de activación en función del potencial resultante  $h_i$ :

$$a_i(t) = f_i(h_i(t)) \quad (2.5)$$

La tabla 2.1 muestra un listado de algunas de las funciones de activación más utilizadas en los distintos modelos de redes neuronales artificiales.

Función	Formula	Rango
<b>Identidad</b>	$y = x$	$[-\infty, \infty]$
<b>Escalón</b>	$y = \begin{cases} +1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$	$[0, 1]$
	$y = \begin{cases} +1 & \text{si } x \geq 0 \\ -1 & \text{si } x < 0 \end{cases}$	$[-1, 1]$
<b>Lineal a tramos</b>	$y = \begin{cases} x & \text{si } -1 \leq x \leq 1 \\ +1 & \text{si } x > 1 \\ -1 & \text{si } x < -1 \end{cases}$	$[-1, 1]$
<b>Sigmoidea</b>	$y = \frac{1}{1 + e^{-x}}$	$[0, 1]$
	$y = \tanh(x)$	$[-1, 1]$
<b>Sinusoidal</b>	$y = \text{Sen}(\omega.x + \varphi)$	$[-1, 1]$

Tabla 2.1. Funciones de activación

### 2.3.5. Función de salida

La función de salida proporciona el valor de salida de la neurona, en base al estado de activación de la neurona. En general se utiliza la función identidad, es decir:

$$y_i(t) = F_i(a_i(t)) = a_i(t) \quad (2.6)$$

## **2.4. Arquitectura de una red neuronal**

Una vez definida el tipo de neurona que se utilizará en un modelo de redes neuronales artificiales es necesario definir la topología de la misma.

La organización y disposición de las neuronas dentro de una red neuronal se denomina topología, y viene dada por el número de capas, la cantidad de neuronas por capa, el grado de conectividad, y el tipo de conexión entre neuronas.

Las neuronas suelen agruparse en unidades funcionales denominadas capas. Se denomina capa de entrada a aquella que esta compuesta por neuronas de entradas y por lo tanto recibe información procedente desde el exterior. Análogamente, se denomina capa oculta y capa de salida a aquellas capas que están compuestas por neuronas ocultas y de salida respectivamente. Una red neuronal artificial esta compuesta por una o más capas, las cuales se encuentran interconectadas entre sí.

Entre un par de neuronas de la red neuronal artificial pueden existir conexiones. Estas conexiones son las sinapsis, tienen asociadas un peso sináptico, y son direccionales.

Cuando la conexión se establece entre dos neuronas de una misma capa hablamos de conexiones laterales o conexiones intra-capas. Por el contrario, si la conexión se establece entre neuronas de distintas capas se la denomina conexión inter-capas. Si la conexión se produce en el sentido inverso al de entrada-salida la conexión se llama recurrente o realimentada.

Una red puede estar formada por una única capa de neuronas. En este caso hablamos de redes monocapa, y las neuronas que conforman dicha capa cumplen la función de neuronas de entrada y salida simultáneamente. Cuando la red esta compuesta por dos o más capas hablamos de redes multicapa.

A su vez, hablamos de redes neuronales con conexión hacia delante (redes feedforward) cuando las conexiones entre las distintas neuronas de la red siguen un único sentido, desde la entrada de la red hacia la salida de la misma. Cuando las conexiones pueden ser tanto hacia delante como hacia atrás hablamos de redes recurrentes (redes feedback).

## **2.5. Aprendizaje**

Durante la operatoria de una red neuronal podemos distinguir claramente dos fases o modos de operación: la fase de aprendizaje o entrenamiento, y la fase de operación o ejecución.

Durante la primera fase, la fase de aprendizaje, la red es entrenada para realizar un determinado tipo de procesamiento. Una vez alcanzado un nivel de entrenamiento adecuado, se pasa a la fase de operación, donde la red es utilizada para llevar a cabo la tarea para la cual fue entrenada.

### **2.5.1. Fase de entrenamiento.**

Una vez seleccionada el tipo de neurona artificial que se utilizará en una red neuronal y determinada su topología es necesario entrenarla para que la red

pueda ser utilizada. Partiendo de un conjunto de pesos sinápticos aleatorio, el proceso de aprendizaje busca un conjunto de pesos que permitan a la red desarrollar correctamente una determinada tarea. Durante el proceso de aprendizaje se va refinando iterativamente la solución hasta alcanzar un nivel de operación suficientemente bueno.

El proceso de aprendizaje se puede dividir en tres grandes grupos de acuerdo a sus características [Isasi Viñuela y Galván León, 2004], [Yao, 1999]:

- Aprendizaje supervisado. Se presenta a la red un conjunto de patrones de entrada junto con la salida esperada. Los pesos se van modificando de manera proporcional al error que se produce entre la salida real de la red y la salida esperada.
- Aprendizaje no supervisado. Se presenta a la red un conjunto de patrones de entrada. No hay información disponible sobre la salida esperada. El proceso de entrenamiento en este caso deberá ajustar sus pesos en base a la correlación existente entre los datos de entrada.
- Aprendizaje por refuerzo. Este tipo de aprendizaje se ubica entre medio de los dos anteriores. Se le presenta a la red un conjunto de patrones de entrada y se le indica a la red si la salida obtenida es o no correcta. Sin embargo, no se le proporciona el valor de la salida esperada. Este tipo de aprendizaje es muy útil en aquellos casos en que se desconoce cual es la salida exacta que debe proporcionar la red.

### **2.5.2. Fase de operación.**

Una vez finalizada la fase de aprendizaje, la red puede ser utilizada para realizar la tarea para la que fue entrenada. Una de las principales ventajas que posee este modelo es que la red aprende la relación existente entre los datos, adquiriendo la capacidad de generalizar conceptos. De esta manera, una red neuronal puede tratar con información que no le fue presentada durante de la fase de entrenamiento.

## **2.6. Redes neuronales con conexión hacia delante**

Las redes neuronales artificiales con conexión hacia delante son el tema central de esta tesis. Este tipo de red, que se caracteriza por su organización en capas y conexiones estrictamente hacia delante, utiliza algoritmos de entrenamiento del tipo supervisado.

Este grupo de red es el más utilizado en aplicaciones prácticas que utilicen redes neuronales, obteniéndose muy buenos resultados fundamentalmente como clasificadores de patrones y estimadores de funciones.

Dentro de este grupo de redes neuronales encontramos al perceptrón, la red ADALINE/MADALINE, y al perceptrón multicapa.

### 2.6.1. Perceptrón.

Este modelo tiene gran importancia histórica ya que fue el primer modelo en poseer un mecanismo de entrenamiento que permite determinar automáticamente los pesos sinápticos que clasifican correctamente a un conjunto de patrones a partir de un conjunto de ejemplos.

La arquitectura del perceptrón está compuesta por dos capas de neuronas, una de entrada y una de salida. La capa de entrada es la que recibe la información proveniente del exterior y la transmite a las neuronas sin realizar ningún tipo de operación sobre la señal de entrada. En general la información entrante es binaria. La función de activación de las neuronas de un perceptrón es del tipo escalón, dando de esta manera sólo salidas binarias. Cada neurona de salida del perceptrón representa a una clase. Una neurona de salida responde con 1 si el vector de entrada pertenece a la clase a la que representa y responde con 0 en caso contrario.

La operación de un perceptrón con  $n$  neuronas de entrada y  $m$  neuronas de salidas puede ser resumida de la siguiente manera:

$$y_i(t) = f\left(\sum_{j=1}^n w_{ij}x_j - \theta_i\right) \quad \forall i, 1 \leq i \leq m \quad (2.7)$$

El algoritmo de entrenamiento del perceptrón se encuentra dentro de los denominados algoritmos por corrección de errores. Este tipo de algoritmos ajustan los pesos de manera proporcional a la diferencia entre la salida actual proporcionada por la red y la salida objetivo, con el fin de minimizar el error producido por la red.

Se puede demostrar que este método de entrenamiento converge siempre en un tiempo finito y con independencia de los pesos de partida, siempre que la función a representar sea linealmente separable. El principal problema de este método de entrenamiento es que cuando la función a representar no es linealmente separable el proceso de entrenamiento oscilará y nunca alcanzará la solución. Las funciones no separables linealmente no pueden ser representadas por un perceptrón.

### 2.6.2. ADALINE / MADALINE

Otro de los modelos que tienen gran importancia es la red neuronal ADALINE. La topología de la red ADALINE es similar a la del perceptrón sólo que en este caso la función de salida de las neuronas es lineal. Dado que las señales de entrada pueden ser continuas, la red ADALINE es un dispositivo de entrada/salida analógica (continua) a diferencia del perceptrón que de acuerdo a lo dicho anteriormente es un dispositivo entrada/salida digital (binaria)

La operación de una red ADALINE con  $n$  neuronas de entrada y  $m$  neuronas de salidas puede ser resumida de la siguiente manera:

$$y_i(t) = \sum_{j=1}^n w_{ij}x_j - \theta_i \quad \forall i, 1 \leq i \leq m \quad (2.8)$$

Sin embargo, la principal diferencia entre la red ADALINE y el perceptrón consiste en la regla de aprendizaje que utilizan. En el caso de la red ADALINE implementa como método de aprendizaje la regla de Widrow-Hoff, también conocida como regla LMS (Least Mean Squares, mínimos cuadrados), que realiza una actualización continua de los pesos sinápticos de acuerdo a la contribución de cada neurona sobre el error total de la red.

Este método produce un conjunto de pesos sinápticos óptimos desde el punto de vista de los mínimos cuadrados (un conjunto de pesos que minimiza el error cuadrático que comete la red), y en caso de que los vectores de entrada sean linealmente independientes produce una asociación perfecta entre entradas-salidas.

Existe una versión multicapa de la ADALINE denominada MADALINE (Multiple ADALINE, múltiples Adalides) que consiste en una red neuronal con neuronas similares a las de la ADALINE pero que contiene capas de neuronas ocultas.

### 2.6.3. Perceptrón multicapa

El perceptrón multicapa es una extensión del perceptrón simple. La topología de un perceptrón multicapa está definida por un conjunto de capas ocultas, una capa de entrada y una de salida. No existen restricciones sobre la función de activación aunque en general se suelen utilizar funciones sigmoideas.

La operación de un perceptrón multicapa con una única capa oculta puede ser resumida de la siguiente manera:

$$z_k = \sum_j w'_{kj} y_i - \theta'_i = \sum_j w'_{kj} f(\sum_i w_{ji} x_i - \theta_i) - \theta'_i \quad (2.9)$$

Este modelo es el más utilizado en la actualidad. El espectro de aplicaciones del perceptrón multicapa es muy amplio lo que hace muy difícil enumerar sus aplicaciones más relevantes. Sin embargo, podemos mencionar algunas áreas de aplicación:

- Codificación de información
- Traducción de texto en lenguaje hablado
- Reconocimiento óptico de caracteres (OCR)

La popularidad de este modelo de redes neuronales no se debe únicamente al éxito obtenido en aplicaciones prácticas del mismo. Existen demostraciones teóricas que permiten explicar el éxito de dichas aplicaciones. En **[Funahashi, 1989]** se demuestra que un perceptrón multicapa cuya función de activación sea no constante, acotada y monótona creciente es un aproximador universal de funciones. En **[Hornik et al., 1989]** se llega a un resultado similar utilizando funciones de activación sigmoideas, no necesariamente continuas.



## Capítulo 3: Entrenamiento de redes neuronales

En el contexto de las redes neuronales el aprendizaje puede ser visto como el proceso de ajuste de los parámetros libres de la red [Yao, 1995]. Partiendo de un conjunto de pesos sinápticos aleatorio, el proceso de aprendizaje busca un conjunto de pesos que permitan a la red desarrollar correctamente una determinada tarea. El proceso de aprendizaje es un proceso iterativo, en el cual se va refinando la solución hasta alcanzar un nivel de operación suficientemente bueno.

La mayoría de los métodos de entrenamiento utilizados en las redes neuronales con conexión hacia delante consisten en proponer una función de error que mida el rendimiento actual de la red en función de los pesos sinápticos. El objetivo del método de entrenamiento es encontrar el conjunto de pesos sinápticos que minimizan (o maximizan) la función. El método de optimización proporciona una regla de actualización de los pesos que en función de los patrones de entrada modifica iterativamente los pesos hasta alcanzar el punto óptimo de la red neuronal.

### 3.1. Métodos de gradiente descendente

El método de entrenamiento más utilizado es el método del gradiente descendente. Este método define una función  $E(W)$  que proporciona el error que comete la red en función del conjunto de pesos sinápticos  $W$ . El objetivo del aprendizaje será encontrar la configuración de pesos que corresponda al mínimo global de la función de error, aunque en muchos casos es suficiente encontrar un mínimo local lo suficientemente bueno [Cauwenberghs, 1993].

El principio general del método es el siguiente: dado un conjunto de pesos  $W(0)$  para el instante de tiempo  $t=0$ , se calcula la dirección de máxima variación del error. La dirección de máximo crecimiento de la función  $E(W)$  en  $W(0)$  viene dado por el gradiente  $\nabla E(W)$ . Luego, se actualizan los pesos siguiendo el sentido contrario al indicado por el gradiente  $\nabla E(W)$ , dirección que indica el sentido de máximo decrecimiento [Defalco, 1997]. De este modo se va produciendo un descenso por la superficie de error hasta alcanzar un mínimo local.

$$W(t+1) = W(t) - \alpha \nabla E(W) \quad (3.1)$$

donde  $\alpha$  indica el tamaño del paso tomado en cada iteración, pudiendo ser diferente para cada peso e idealmente debería ser infinitesimal. El tamaño del paso es un factor importante a la hora de diseñar un método de estas características. Si se toma un paso muy chico el proceso de entrenamiento resulta muy lento, mientras que si el tamaño del paso es muy grande se producen oscilaciones en torno al punto mínimo.

### 3.2. El algoritmo Backpropagation

El algoritmo backpropagation es el método de entrenamiento más utilizado en redes con conexión hacia delante. Es un método de aprendizaje supervisado de gradiente descendente, en el que se distinguen claramente dos fases: primero se aplica un patrón de entrada, el cual se propaga por las distintas capas que componen la red hasta producir la salida de la misma. Esta salida se compara con la salida deseada y se calcula el error cometido por cada neurona de salida. Estos errores se transmiten hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de las capas intermedias [Fritsch, 1996]. Cada neurona recibe un error que es proporcional a su contribución sobre el error total de la red. Basándose en el error recibido, se ajustan los errores de los pesos sinápticos de cada neurona.

#### 3.2.1. Deducción del algoritmo Backpropagation

El algoritmo propone una actualización iterativa de los pesos de la siguiente manera:

$$W(t+1) = W(t) + \Delta W(t) \quad (3.2)$$

Si tomamos una variación proporcional al gradiente de una función de error  $E(W)$  tenemos que:

$$W(t+1) = W(t) - \alpha \nabla E[W(t)] \quad (3.3)$$

Como se explico anteriormente el primer paso de este algoritmo consiste en propagar hacia delante un patrón de entrada  $X_p$  y obtener la salida de la red  $Y_p$ .

La salida de la neurona  $i$  viene dada según su estado de activación. Si consideramos la función de salida identidad tenemos que

$$y_i(t) = F_i(a_i(t)) = a_i(t) \quad (3.4)$$

Siendo

$$a_i(t) = f_i(h_i(t)) \quad (3.5)$$

La regla de propagación más simple y utilizada consiste en realizar una suma de las entradas ponderadas con sus pesos sinápticos correspondientes.

$$h_i(t) = \sum_j w_{ij} * x_j(t) \quad (3.6)$$

Se compara la salida obtenida  $Y_p$  con la salida deseada  $D_p$ , obteniéndose un error que viene dado por



$$e_p = \frac{1}{2} \sum_{k=1}^M (d_{pk} - y_{pk})^2 \quad (3.7)$$

donde k es el índice de neurona para las neuronas de la última capa, y M el total de neuronas de la misma.

El error total de la red está dado por

$$e = \frac{\sum_{p=1}^P e_p}{P} \quad (3.8)$$

siendo p el índice de ejemplo, y P el numero total de ejemplos.

De acuerdo a la ecuación (3.3) la variación de los pesos sinápticos será proporcional al gradiente de la función de error:

$$\Delta w_{ji} = -\alpha \frac{\partial e_p}{\partial w_{ji}} \quad (3.9)$$

Si aplicamos la regla de la cadena a (3.9) obtenemos que

$$\frac{\partial e_p}{\partial w_{ji}} = \frac{\partial e_p}{\partial h_j} \frac{\partial h_j}{\partial w_{ji}} \quad (3.10)$$

La ecuación (3.10) expresa la derivada del error en función de dos derivadas. La derivada del error respecto al potencial resultante  $h_j$  indica como varia el error al variar la entrada de la neurona j, mientras que la derivada con respecto al peso sináptico  $w_{ji}$  indica como varia la entrada de la neurona j al variar el peso de la conexión que va desde la neurona i hasta la neurona j.

El segundo termino de la expresión (3.10) lo podemos expresar a partir de la ecuación (3.6) de la siguiente manera

$$\frac{\partial h_j}{\partial w_{ji}} = \frac{\partial \sum_i w_{ji} y_{pi}}{\partial w_{ji}} = y_{pi} \quad (3.11)$$

Si escribimos al primer término de la ecuación (3.10) como

$$\frac{\partial e_p}{\partial h_j} = -\delta_{pj} \quad (3.12)$$

tenemos que

$$\frac{\partial e_p}{\partial w_{ji}} = -\delta_{pj} y_{pi} \quad (3.13)$$

y por lo tanto la ecuación (3.9) queda expresada de la siguiente manera

$$\Delta w_{ji} = -\alpha \delta_{pj} y_{pi} \quad (3.14)$$

Para calcular el valor de delta se vuelve a aplicar la regla de la cadena

$$\delta_{pj} = -\frac{\partial e_p}{\partial h_j} = -\left( \frac{\partial e_p}{\partial y_{pj}} \frac{\partial y_{pj}}{\partial h_j} \right) \quad (3.15)$$

El cálculo del segundo término de la ecuación (3.15) es simple si observamos las ecuaciones (3.4) y (3.5)

$$\frac{\partial y_{pj}}{\partial h_j} = \frac{\partial f_j(h_j)}{\partial h_j} = f'_j(h_j) \quad (3.16)$$

sin embargo, para el cálculo del primer término de la ecuación (3.15) es necesario distinguir entre dos casos diferentes.

- **La neurona j es una neurona de salida**

En este caso podemos obtener el segundo término a partir de la ecuación (3.7) ya que el subíndice j es igual al subíndice k

$$\frac{\partial e_p}{\partial y_{pj}} = \frac{\partial \frac{1}{2} \sum_{j=1}^M (d_{pj} - y_{pj})^2}{\partial y_{pj}} = -(d_{pj} - y_{pj}) \quad (3.17)$$

Así, la variación de los pesos de una conexión que va hacia la capa externa de la red se calcula como:

$$\Delta w_{ji} = \alpha (d_{pj} - y_{pj}) f'_j(h_j) y_{pi} \quad (3.18)$$

- **La neurona j es una neurona oculta**

En este caso es necesario aplicar nuevamente la regla de la cadena

$$\frac{\partial e_p}{\partial y_{pj}} = \sum_k \left( \frac{\partial e_p}{\partial h_k} \frac{\partial h_k}{\partial y_{pj}} \right) \quad (3.19)$$

Donde k es el subíndice de las neuronas que pertenecen a la próxima capa. La ecuación (3.19) la podemos reescribir utilizando la ecuación (3.6)

$$\frac{\partial e_p}{\partial y_{pj}} = \sum_k \left( \frac{\partial e_p}{\partial h_k} \frac{\partial \left( \sum_j w_{kj} y_{pj} \right)}{\partial y_{pj}} \right) = \sum_k \left( \frac{\partial e_p}{\partial h_k} w_{kj} \right) \quad (3.20)$$

y por la ecuación (3.12) tenemos que

$$\frac{\partial e_p}{\partial y_{pj}} = \sum_k -\delta_{pk} w_{kj} = -\sum_k \delta_{pj} w_{kj} \quad (3.21)$$

Así, la variación de los pesos de una conexión que va desde una capa hacia otra capa de la red que no sea la externa se calcula como:

$$\Delta w_{ji} = \alpha \sum_k (\delta_{pk} w_{kj}) f'_j(h_j) y_{pi} \quad (3.22)$$

En la implementación del algoritmo, se toma una amplitud de paso que viene dado por la tasa de aprendizaje ( $\alpha$ ). A mayor tasa de aprendizaje el proceso será más rápido. Sin embargo, si la tasa de aprendizaje es muy alta puede dar lugar a oscilaciones en torno a un mínimo local. Es posible disminuir el impacto de dichas oscilaciones mediante la adición de un momento ( $\beta$ ), quedando la expresión (3.14) expresada de la siguiente manera:

$$\Delta w_{ji}(t+1) = \alpha \delta_{pj} y_{pj} + \beta \Delta w_{ji}(t) \quad (3.23)$$

De esta manera el momento  $\beta$  determina el efecto en el instante  $t+1$  del cambio de los pesos realizado en el instante  $t$ .

Con este momento se consigue la convergencia de la red en menor numero de iteraciones, ya que si la modificación de los pesos en los instantes  $t$  y  $t+1$  es en la misma dirección, entonces el descenso por la superficie de error en  $t+1$  es mayor. En cambio, si la modificación en los pesos en los instantes  $t$  y  $t+1$  se produce en direcciones opuestas, el paso que se da en  $t+1$  es más pequeño, lo que es adecuado, ya que esto significa que se a pasado por un mínimo.

Resumiendo, el algoritmo backpropagation queda expresado de la siguiente manera:

$w_{ji}(t+1) = w_{ji}(t) + [\alpha \delta_{pj} y_{pj} + \beta \Delta w_{ji}(t)]$
$\text{siendo } \delta_{pj} = \begin{cases} (d_{pj} - y_{pj}) f'_j(h_j) & \text{si } j \text{ es una neurona de salida} \\ \left( \sum_k \delta_{pk} w_{kj} \right) f'_j(h_j) & \text{si } j \text{ es una neurona oculta} \end{cases}$

### 3.2.2. Modos de entrenamiento

Durante la aplicación del algoritmo backpropagation, el aprendizaje se produce mediante la presentación sucesiva de un set de entrenamiento. Cada presentación completa al perceptrón multicapa del set de entrenamiento se denomina epoch. Así, el proceso de aprendizaje se repite epoch tras epoch hasta que los pesos sinápticos se estabilizan y la performance de la red converge a un valor aceptable.

La forma en que se actualizan los pesos sinápticos da lugar a dos modos de entrenamientos distintos, cada uno con sus ventajas y desventajas.

- **Modo Secuencial**

En este modo de entrenamiento la actualización de los pesos sinápticos se produce tras la presentación de cada ejemplo de entrenamiento [Yao, 1993a], de allí que también es conocido como modo por patrón. Si un set de entrenamientos posee N ejemplos, el modo secuencial de entrenamiento tiene como resultado N correcciones de pesos sinápticos durante cada epoch.

- **Modo Batch**

En este modo de entrenamiento la actualización de los pesos sinápticos se produce una única vez, tras la presentación de todo el set de entrenamiento. Para cada epoch se calcula el error cuadrático medio (3.8) producido por la red. La variación de los pesos sinápticos, para un set de entrenamiento de N ejemplos, se puede calcular a partir de las ecuaciones (3.7), (3.8) y (3.9) como:

$$\Delta w_{ji} = -\alpha \frac{\partial e}{\partial w_{ji}} = -\frac{\alpha}{N} \sum_{n=1}^N e_j \frac{\partial e_j}{\partial w_{ji}} \quad (3.24)$$

Luego, la derivada podemos definirla de la misma manera que la definimos previamente.

Si los patrones de entrenamiento se presentan a la red de manera aleatoria, el modo de entrenamiento secuencial convierte a la búsqueda en el espacio de pesos en estocástica por naturaleza, y disminuye la probabilidad de que el algoritmo backpropagation quede atrapado en un mínimo local. Sin embargo, la naturaleza estocástica del modo de entrenamiento secuencial dificulta el establecimiento de condiciones teóricas para la convergencia del algoritmo.

Por su parte, el uso del modo de entrenamiento batch provee una estimación precisa del vector gradiente, garantizando de esta manera la convergencia hacia un mínimo local.

### 3.2.3. Aceleración del aprendizaje

Como se menciono previamente las redes el algoritmo backpropagation ha sido utilizado con éxito en gran cantidad de aplicaciones. Sin embargo, el éxito

y la velocidad de convergencia de este mecanismo de entrenamiento tienen un alto grado de dependencia de la configuración del mismo. Por ello se han realizado una serie de métodos que permiten mejorar significativamente la performance del algoritmo.

- **Modo de actualización.** La actualización secuencial es más rápida computacionalmente y demanda menos recursos que la actualización batch. Esto es especialmente cierto cuando el set de datos es grande y altamente redundante [Chinrungrueng, 1993].
- **Set de datos.** La calidad del set de datos es un factor muy importante a tener en cuenta. Cada ejemplo presentado a la red debe cumplir con las siguientes dos premisas:
  - Maximizar el error de entrenamiento
  - Maximizar la información
- **Presentación de los ejemplos.** El modo en que se presentan los ejemplos es otro factor importante a tener en cuenta. La aleatorización del orden en que se presentan los ejemplos en los distintos epochs evita que los resultados se vean distorsionados por el orden de los ejemplos
- **Función de activación.** El uso de una función de activación adecuada puede acelerar notoriamente el tiempo de entrenamiento. El uso de funciones antisimétricas en general produce mejores tiempos que el uso de funciones no simétricas.
- **Valores objetivos.** La selección de los valores objetivos debe hacerse de acuerdo a la función de activación seleccionada. Una técnica que permite acelerar el tiempo de aprendizaje es desplazar el valor objetivo del valor máximo de la función. El algoritmo backpropagation tiende a saturar las neuronas ocultas cuando el valor objetivo es igual al máximo de la función de activación. Cuando esto sucede se produce el efecto de parálisis, lo que produce un aumento en el tiempo total de entrenamiento. Desplazando el valor objetivo un offset e del valor máximo de la función se reduce el riesgo de que las neuronas ocultas se saturen.
- **Normalización de las entradas.** Si bien el algoritmo backpropagation no exige que los valores de entrada a la red se encuentren normalizados, esta es una buena técnica para acelerar los tiempos de entrenamiento [Bishop, 1996]. La normalización de las entradas debe realizarse de manera tal que el valor medio de la misma se encuentre cercano a cero.
- **Preprocesamiento de los ejemplos.** Se aplica en aquellos casos en que un atributo toma un conjunto discreto de valores. Si un atributo sólo puede tomar N valores diferentes, la entrada de la red puede

subdividirse en N entradas, cada una de las cuales representa a una clase. Cada una de estas entradas ahora puede tomar dos valores, verdadero o falso. Esta técnica puede ayudar a mejorar los tiempos de entrenamiento de la red neuronal

### **3.3. Generalización**

Una vez finalizada la fase de aprendizaje, la red puede ser utilizada para realizar la tarea para la que fue entrenada. Una de las principales ventajas que posee este modelo es que la red aprende la relación existente entre los datos, adquiriendo la capacidad de generalizar conceptos. De esta manera, una red neuronal puede tratar con información que no le fue presentada durante de la fase de entrenamiento **[Chinrungrueng, 1988]**.

Cuando se evalúa una red neuronal no sólo es importante evaluar si la red ha sido capaz de aprender los patrones de entrenamiento. Es imprescindible también evaluar el comportamiento de la red ante patrones nunca antes vistos. Esta característica de las redes neuronales se la conoce como capacidad de generalización y es adquirida durante la fase de entrenamiento **[Sanger, 1989]**. Es necesario que durante el proceso de aprendizaje la red extraiga las características de las muestras, para poder luego responder correctamente a nuevos patrones.

De lo dicho anteriormente surge la necesidad de evaluar durante la fase de entrenamiento dos tipos de errores. El error de aprendizaje, que indica la calidad de la respuesta de la red a los patrones de entrenamiento, y el error de generalización, que indica la calidad de la respuesta de la red a patrones nunca antes vistos. Para poder obtener una medida de ambos errores es necesario dividir el set de datos disponibles en dos, el set de datos de entrenamiento, y el set de datos de evaluación. El primero se utiliza durante la fase de entrenamiento para que la red pueda extraer las características de los mismos y, mediante el ajuste de sus pesos sinápticos, la red logre una representación interna de la función. El set de evaluación se utiliza para evaluar la capacidad de generalización de la red.

La causa más común de la pérdida de capacidad de generalización es el sobreaprendizaje. Esto sucede cuando la cantidad de ciclos de entrenamientos tiende a ser muy alta. Se observa que la respuesta de la red a los patrones de entrenamiento es muy buena mientras que la respuesta a nuevos patrones tiende a ser muy pobre. Al aumentar el número de ciclos la red tiende a sobreajustar la respuesta a los patrones de entrenamiento, a expensas de una menor capacidad de generalización. La figura 3.1 muestra una situación idealizada de lo dicho anteriormente. En la misma se observa que en un determinado punto la red comienza a perder capacidad de generalización como consecuencia del sobreaprendizaje de los patrones de entrenamiento.

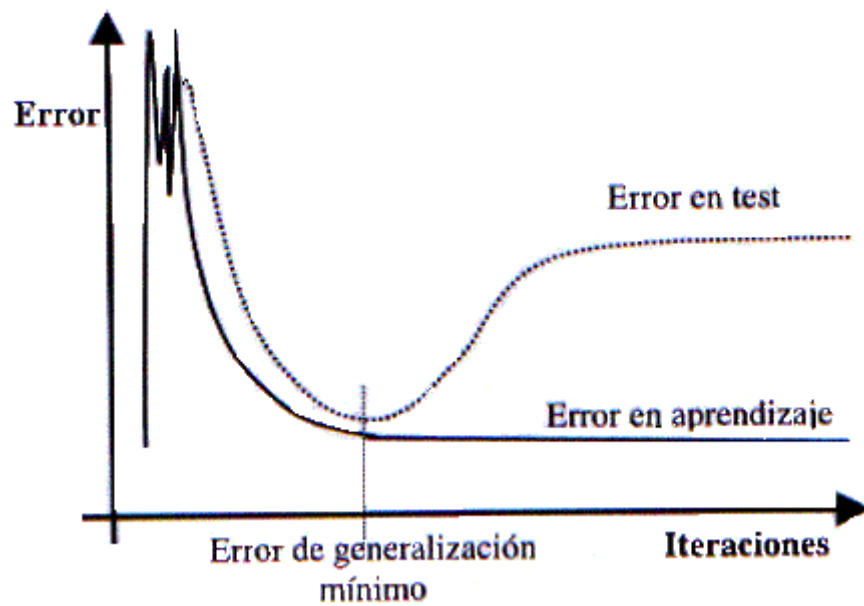


Figura 3.1. Generalización. Situación idealizada

En la figura 3.2 se muestra una situación más real del mismo caso. A medida que transcurre el proceso de aprendizaje se obtienen varios mínimos sobre el conjunto de evaluación. Existen diversas técnicas de parada temprana (early stopping) aunque en la mayoría de los casos se deja que el proceso de aprendizaje avance hasta alcanzar una cota de error razonable, guardando periódicamente las distintas configuraciones intermedias para luego seleccionar la de menor error de evaluación.

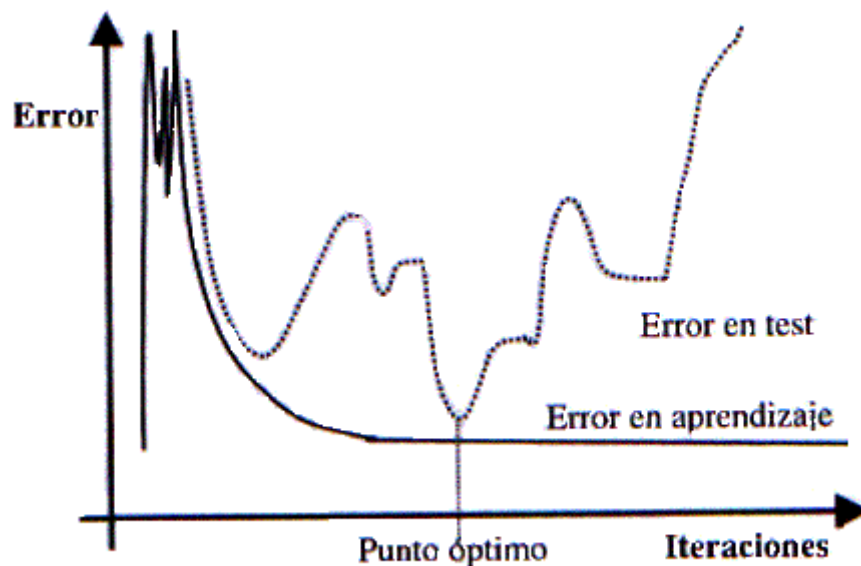


Figura 3.2. Generalización. Situación real

En ocasiones la pérdida de capacidad de generalización se produce por el uso excesivo de neuronas ocultas en la red neuronal. Esto hace que la red tienda a ajustar con mucha exactitud los patrones de entrenamiento, evitando que la red extraiga las características del conjunto. Este problema se ve agravado cuando los patrones de entrenamiento poseen ruido, ya que la red ajusta también el ruido de los mismos.



## Capítulo 4: Algoritmos genéticos

Los algoritmos genéticos son una técnica de resolución de problemas inspirada en la naturaleza. Están basados en el principio darwiniano de reproducción y supervivencia de los individuos más aptos [Beasley et al., 1993]

### 4.1. Introducción

La computación evolutiva es la rama de la inteligencia artificial que engloba a todas aquellas técnicas de resolución de problemas basadas en la evolución de las especies y la supervivencia del más apto [Joglekar y Tungare, 2001]. Dentro de ella encontramos a los algoritmos genéticos (genetic algorithms), las estrategias evolutivas (evolution strategies) y la programación evolutiva (evolutionary programming) entre otros [Yao, 1996].

Las técnicas evolutivas han sido aplicadas con éxito a distintos tipos de problemas como optimización de parámetros, planificación de tareas, diseño, etc. [Whitley, 2002]

Estos algoritmos codifican las posibles soluciones en estructuras llamadas cromosomas (o individuos). A un conjunto de individuos se lo conoce como población y representan un conjunto de soluciones posibles al problema [Cantú-Paz, 1997]. Mediante la aplicación de un conjunto de operadores genéticos sobre la población se va refinando gradualmente la solución hasta alcanzar un resultado que cumpla con las condiciones requeridas.

El primer paso en la aplicación de un algoritmo genético consiste en la generación de una población inicial. En general esta población se genera de manera aleatoria, y el tamaño de dicha población (la cantidad de individuos que la compone) es un parámetro que se define durante el diseño del algoritmo genético. Una vez generada esta población se debe evaluar la aptitud (fitness) de cada individuo [Deb, 2004].

El operador de selección es el encargado de decidir cuales individuos contribuirán en la formación de la próxima generación de individuos. Este mecanismo simula el proceso de selección natural, mediante el cual sólo los individuos más adaptados al ambiente se reproducen [Coello Coello, 2002]. El mecanismo de selección forma una población *intermedia*, que esta compuesta por los individuos con mayor aptitud de la generación actual.

La siguiente fase del algoritmo consiste en la aplicación de los operadores genéticos. El primero de ellos es la cruza, y su función es recombinar el material genético. Se toman aleatoriamente dos individuos que hayan sobrevivido al proceso de selección y se recombina su material genético creando uno o más descendientes, que pasan a la siguiente población. Este operador se aplica tantas veces como sea necesario para formar la nueva población.

El último paso consiste en la aplicación del operador de mutación. Este operador, que en general actúa con muy baja probabilidad, modifica algunos genes del cromosoma, posibilitando de esta manera la búsqueda de soluciones alternativas.

Una vez finalizado el proceso de selección, cruza y mutación se obtiene la siguiente generación del algoritmo, la cual será evaluada, repitiéndose el ciclo descrito previamente. Tras cada iteración la calidad de la solución generalmente va incrementándose, y los individuos representan mejores soluciones al problema [Forrest 1996].

Al algoritmo genético detallado anteriormente se lo conoce como algoritmo genético canónico y es la forma más utilizada. Sin embargo, en algunas implementaciones particulares de algoritmos genéticos se puede agregar nuevos operadores. En todos los casos, la forma en que se implementen los operadores variará de acuerdo a las características propias del problema.

## 4.2. El cromosoma

Los algoritmos genéticos trabajan manipulando cromosomas, que son estructuras que codifican las distintas soluciones de un determinado problema. La forma en que se codifican los cromosomas es dependiente de cada problema en particular, y suele variar de problema en problema.

Un cromosoma esta compuesto por un conjunto de genes. Cada **gen** representa una característica particular del individuo y ocupa una posición determinada en el cromosoma, llamada **locus**. A cada uno de los valores que puede tomar un gen se lo conoce como **alelo**. [Ilachinski, 1997]

Es importante destacar dos conceptos que muchas veces suelen confundirse: genotipo y fenotipo. El **genotipo** es el conjunto de genes de un individuo, la descripción genética del individuo. El **fenotipo** es la forma en que se expresa el genotipo, como resultado de la interacción con su entorno [Morrison, 1998]. La **morfogénesis** es el proceso de decodificar el genotipo para producir un fenotipo [Biondi y Michel, 1995].

Una cuestión a resolver cuando se diseñan algoritmos evolutivos es la forma de codificar los genes. Estos se pueden representar como cadenas binarias, números enteros, números reales, etc. [Whitley, 2001]. En general se utilizan las cadenas binarias por varios motivos:

- Cualquier parámetro se puede codificar como una cadena de bits. Una cadena de bits permite expresar valores booleanos, enteros, reales, etc.
- La codificación binaria se puede ajustar a cualquier problema. Por este motivo la mayor parte de los trabajos teóricos se basa en este esquema de codificación. El uso de cadenas de bits permite contar con gran cantidad de herramientas de análisis.
- Las cadenas de bits son fáciles de manipular. El desarrollo de operadores genéticos es simple y eficiente.

Sin embargo, las características propias del problema podrían llevar a la utilización de otro esquema de codificación. En dicho caso será necesario desarrollar operadores genéticos que se adapten al esquema seleccionado.

En lo que sigue de este capítulo se supone el uso de cadenas binarias, debido a que estas son más fáciles de comprender.

### 4.3. Evaluación y aptitud

El uso de los conceptos de función de evaluación y función de aptitud se suelen utilizar como sinónimos. Sin embargo, ambos conceptos son diferentes, y es conveniente hacer una distinción entre ellos.

La función de evaluación, o función objetivo, provee una medida de la performance de conjunto de parámetros. Indica que tan buena es la solución obtenida tras la aplicación de un conjunto de parámetros, independientemente de la aplicación de esta función sobre otro conjunto de parámetros [Whitley, 1994]. En general, esta medida se obtiene tras la decodificación de un cromosoma en el conjunto de parámetros que representa.

Por su parte, la función de adaptación siempre esta definida con respecto al resto de los individuos de la población. Indica que tan buena (o mala) es una solución comparada con el resto de las soluciones obtenidas hasta el momento. El valor obtenido por esta función se traduce, tras la aplicación del operador de selección, en oportunidades de reproducción. Aquellos que tengan mayor aptitud tendrán mayores posibilidades de reproducirse y transmitir su material genético a la próxima generación.

En el algoritmo genético canónico, la función de adaptación se define como

$$F_i = \frac{f_i}{\bar{f}} \quad (4.1)$$

$$\bar{f} = \frac{\sum_{i=1}^N f_i}{N} \quad (4.2)$$

donde  $f_i$  es el valor de la función de evaluación para el cromosoma  $i$

### 4.4. Población

Los algoritmos genéticos trabajan sobre una población de individuos [Bäck, 1992], donde cada uno de los cuales representa una posible solución. La población es un concepto muy importante en los algoritmos genéticos y existen dos cuestiones fundamentales a resolver: como generar la primera población, y cual debe ser el tamaño de la población.

La población inicial debe poseer la mayor diversidad posible. Idealmente, la población inicial debería contener todos los posibles valores (alelos) que pueda tomar un gen. De esta manera se aseguraría una exploración completa del espacio de búsqueda. En la práctica, esta situación no suele ser factible por lo que la primera población se genera comúnmente de manera azarosa, asignándole aleatoriamente a cada gen uno de los posibles alelos. De esta manera se asegura la diversidad de alelo por gen. En algunos casos se pueden

utilizar heurísticas para la generación de la población inicial, las que permiten obtener un punto de partida más próximo a la solución del problema. En estos casos, es importante que la heurística asegure la diversidad mencionada. De lo contrario el algoritmo genético no será capaz de realizar una búsqueda completa sobre el espacio de soluciones.

El tamaño de la población no es problema crítico. Si el tamaño de la población es muy grande se realizará una exploración del espacio de búsqueda más rápida en términos de generaciones. Sin embargo, el tiempo necesario para obtener una nueva generación a partir de la generación actual será mayor, dado que se estarán procesando individuos innecesarios. Si el tamaño de la población es muy chico podría llegar a darse el caso en que la población converja rápidamente a una solución que no sea lo suficientemente buena. En la práctica se suelen utilizar poblaciones de 20 individuos, pudiéndose aumentar el mismo en base a la complejidad del problema o a la calidad de la solución requerida. En algunos casos se utiliza un tamaño de población inicial, el cual se va incrementando a medida que la población tiende a ser homogénea.

## **4.5 Variantes de los operadores básicos**

El objetivo de cada uno de los tres operadores básicos está bien definido: elegir los individuos de la población que tengan mayor grado de adaptación (selección), recombinar el material genético de ellos para producir nuevos individuos (cruza) y alterar características de algunos de ellos para garantizar la diversidad (mutación). Siguiendo estos objetivos se han desarrollado gran cantidad de variantes para cada uno de los operadores. Cada variante tiene características particulares que afectarán el comportamiento del algoritmo genético. Una elección adecuada para cada operador puede influir decisivamente en la eficiencia del proceso de búsqueda. A continuación se realiza una mención sobre las variantes más utilizadas de cada uno de los operadores.

### **4.5.1. Selección**

#### **4.5.1.1. Selección por ruleta**

El primer paso para la aplicación de este operador consiste en obtener aleatoriamente un número  $R$  de la siguiente manera:

$$R = \text{Random}(0, \sum_{i=1}^N F_i) \quad (4.3)$$

Luego se selecciona al individuo  $j$  tal que:

$$\sum_{i=1}^j F_i \leq R < \sum_{i=1}^{j+1} F_i \quad (4.4)$$

Este individuo pasa a la siguiente generación, repitiéndose el procedimiento tantas veces como sea necesario hasta completar la siguiente generación. Al contener una componente aleatoria, la cantidad de copias de un individuo que efectivamente pasan a la siguiente generación puede variar de la cantidad de copias esperadas.

#### **4.5.1.2. Selección proporcional**

Este operador asigna un número de copias proporcional a la aptitud de un individuo. El número de copias del individuo  $i$  que pasarán a la siguiente población se calcula en base a la siguiente ecuación:

$$c_i = N * \frac{f_i}{\sum_{i=1}^N F_i} \quad (4.5)$$

#### **4.5.1.3. Selección por torneo**

Se seleccionan aleatoriamente dos individuos de la población, pasando a la siguiente generación aquel individuo que tiene mayor aptitud. Este proceso se repite  $N$  veces, donde  $N$  es el tamaño de la población. La selección por torneo no asegura que el número de copias que pasan a la próxima generación sea igual al esperado. Este operador se puede implementar tomando cualquier número de individuos.

#### **4.5.1.4. Selección por ranking**

Este operador asigna una probabilidad de selección proporcional a la aptitud del individuo. Sin embargo, la selección se realiza con un esquema análogo al de selección por ruleta. Se asigna a un individuo una probabilidad de selección igual a:

$$p_i = \frac{N - i}{\sum_{i=1}^N i} \quad (4.6)$$

Luego se selecciona un número aleatorio  $R$  de la siguiente manera:

$$R = \text{Random}(0, \sum_{i=1}^N p_i) \quad (4.7)$$

Por último se elige al individuo  $j$  tal que:

$$\sum_{i=1}^j p_i \leq R < \sum_{i=1}^{j+1} p_i \quad (4.8)$$

Este individuo pasa a la siguiente generación. El procedimiento se repite tantas veces como sea necesario hasta completar la siguiente generación. Nuevamente, la cantidad de copias que se pasan a la siguiente generación puede diferir de la cantidad esperada con un esquema proporcional.

## 4.5.2. Cruza

### 4.5.2.1. Cruza Uniforme

Este operador de cruza asigna aleatoriamente los pesos. Cada hijo tiene una probabilidad de 0.5 de recibir los genes de su padre y por ende de recibirlos de su madre.

### 4.5.2.2. Cruza Simple

En esta variante del operador de cruza se toma aleatoriamente un punto de cruza. Luego, a un hijo se le asignan todos los genes del padre ubicados a la izquierda del punto de cruza, y todos los genes de la madre ubicados a la derecha del punto de cruza. El segundo hijo es el complemento del primero. La figura 4.1 muestra un ejemplo de cruza simple en el cual se toma como punto de cruza el primero de ellos.

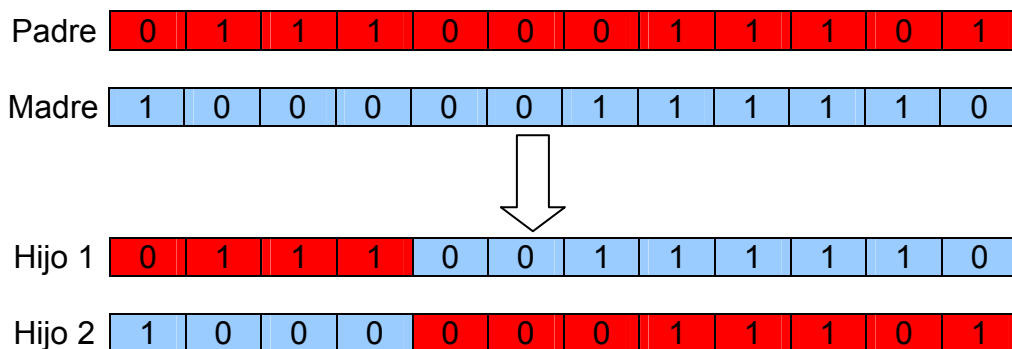


Figura 4.1. Cruza Simple

### 4.5.2.3. Cruza Multipunto

Esta variante del operador de cruza es similar al operador de cruza simple, sólo que la cantidad de puntos de cruza se determina aleatoriamente. Cada hijo recibe los genes entre dos puntos de cruza sucesivos de cada uno de los padres, de manera intercalada. Cabe destacar que la cruza simple es un caso particular de la cruza multipunto, con un único punto de cruza. La figura 4.2 muestra un ejemplo de cruza multipunto, tomando dos puntos de cruza.

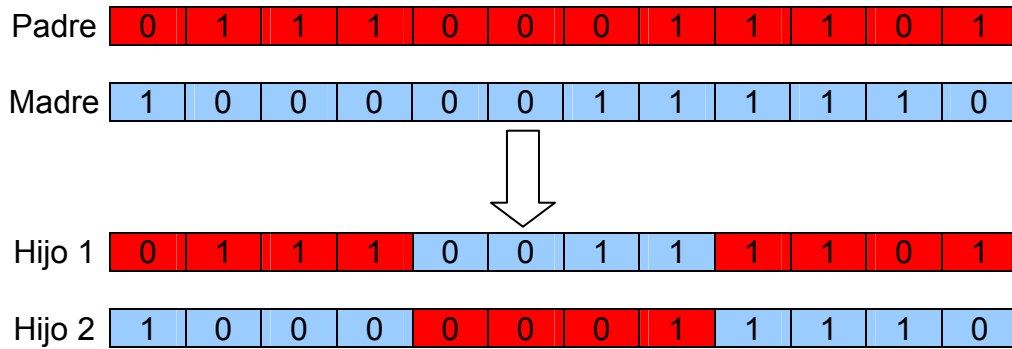


Figura 4.2. Cruza Multipunto

#### 4.5.2.4. Cruza Binomial

Este operador es similar al operador de cruza uniforme, sólo que las probabilidades se definen en función de la aptitud de los padres de la siguiente manera:

$$\begin{cases} p_{padre} = \frac{f_{padre}}{f_{padre} + f_{madre}} \\ p_{madre} = 1 - p_{padre} \end{cases} \quad (4.9)$$

#### 4.5.3. Mutación

La mutación binaria es el tipo de mutación tradicionalmente utilizado en el algoritmo genético canónico, y consiste en invertir un gen aleatoriamente. Dada la representación binaria del gen, se invierte el bit que lo representa con una determinada probabilidad, como lo muestra la figura 4.3.

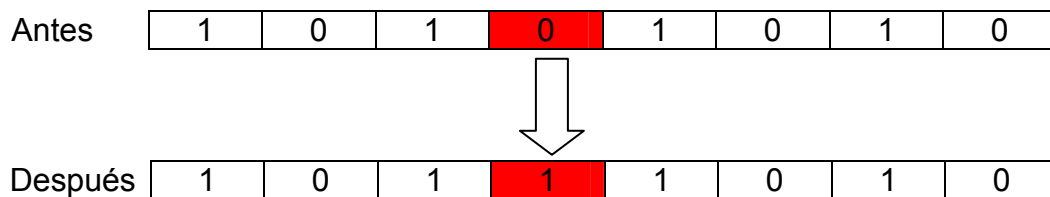


Figura 4.3. Mutación Binaria

##### 4.5.3.1. Mutación simple

En este caso la probabilidad de mutación permanece constante a lo largo de las distintas generaciones, como lo muestra la ecuación 4.10.

$$P_m(t) = P_m(0) = P_m^{\max} = P_m^{\min} \quad (4.10)$$

##### 4.5.3.2. Mutación adaptativa por temperatura ascendente

La probabilidad de mutación se va incrementando a medida que transcurren las generaciones. El objetivo de este aumento es mantener la

diversidad de individuos en la población, que tiende a hacerse homogénea con el transcurso de las generaciones. La probabilidad de mutación para una generación esta dada por la ecuación 4.11.

$$P_m(t) = P_m^{\min} + \frac{(P_m^{\max} - P_m^{\min})}{T} * t \quad (4.11)$$

### **4.5.3.3. Mutación adaptativa por temperatura descendente**

La probabilidad de mutación va decreciendo a medida que transcurren las generaciones. De esta manera se asegura una alta diversidad de individuos en las generaciones iniciales. La probabilidad mínima debe ser mayor a cero para permitir continuar la exploración del espacio de búsqueda a medida que las generaciones avanzan. La ecuación 4.12 muestra como se calcula la probabilidad de mutación para una determinada generación.

$$P_m(t) = P_m^{\max} - \frac{(P_m^{\max} - P_m^{\min})}{T} * t \quad (4.12)$$



## Capítulo 5: Descripción del problema

Las redes neuronales con conexión hacia delante en general son un importante método de aproximación de funciones [Kim, 1992]. Como se mencionó en el capítulo 2, el perceptrón multicapa es un aproximador universal de funciones.

Pese a todo, el algoritmo backpropagation suele tener algunas dificultades para encontrar un conjunto de pesos que permita utilizar la red como aproximador de funciones. Todos estos problemas deben ser tenidos en cuenta por el diseñador de una red neuronal, quien debe actuar en función de ellos. Para algunos de estos problemas existen técnicas que permiten disminuir su efecto, pero en ningún caso puede ser eliminado completamente.

El objetivo de este capítulo es hacer una síntesis de los diferentes problemas que suelen encontrarse durante una sesión de entrenamiento de un perceptrón multicapa utilizando el algoritmo backpropagation, así como también mencionar las técnicas que se pueden utilizar para disminuir su efecto. Al final del capítulo se plantean también los objetivos y cuestiones a resolver mediante el método propuesto.

### 5.1. Deficiencias del algoritmo

A pesar del éxito del algoritmo para entrenar redes multicapa, este algoritmo posee una serie de deficiencias, las cuales se analizan a continuación.

- **Adaptabilidad**

El algoritmo tiene como premisa la utilización de una función de activación derivable [Walker, 1995]. Al hacer uso de la derivada de la función de activación, es condición necesaria para la aplicación del algoritmo que la misma sea continua y derivable en todo el dominio de aplicación [Wilson, 1994]. Esto impide la utilización del método en otras topología como por ejemplo en redes Adaline/Madaline, donde la función de activación presenta discontinuidades.

Este problema suele encontrarse en varios métodos de entrenamiento, los cuales son desarrollados para una determinada topología y sus resultados, en general, no son extensibles directamente a otras topologías. Es necesario adaptar los métodos para aplicarlos a otras topologías.

- **Dependencia de parámetros del algoritmo**

Los algoritmos de gradiente descendente hacen uso de una tasa de aprendizaje que idealmente debería ser infinitesimal. De esta manera, mediante pequeños ajustes de los pesos sinápticos el algoritmo converge hacia un mínimo. El uso de tasas de aprendizaje muy pequeñas hace que el algoritmo tenga una convergencia estable hacia un mínimo, aunque el tiempo necesario para alcanzarlo puede llegar a ser muy alto.

Como consecuencia de lo dicho anteriormente, y con el objetivo de disminuir el tiempo de convergencia del algoritmo, en la práctica se suelen utilizar tasas de aprendizajes mayores a las teóricas. El aumento de la tasa de aprendizaje disminuye el tiempo de convergencia, pero tiene un efecto contraproducente: el algoritmo comienza a oscilar en torno a un mínimo, disminuyendo la probabilidad de alcanzarlo. El efecto de oscilación puede reducirse mediante la adición de una tasa de momento, como se describió en el capítulo 3, pero no puede eliminarse.

El algoritmo backpropagation es muy dependiente de los parámetros mencionados previamente. Dependiendo de la selección de parámetros realizadas el resultado de la aplicación del algoritmo será exitosa o no [Liu *et al*, 2004]. Pequeñas variaciones sobre los parámetros del algoritmo pueden conducir a resultados diferentes. El principal problema es que no existe un método general que permita establecer el valor de estos parámetros [Branke, 1995]. Los parámetros que aseguran la convergencia para un determinado problema pueden no ser aplicables a otro problema. De esta manera, la selección de los parámetros del algoritmo se realiza en base a la expertiz del diseñador, y se realiza un refinamiento de los mismos mediante mecanismos de prueba y error. Esto produce un aumento en el tiempo total de diseño y entrenamiento de la red.

- **Mínimos locales**

La superficie que define la función de error  $E$  (ecuación 8) en base a los parámetros de la red neuronal es compleja y esta llena de valles y colinas. Debido a la utilización del gradiente para encontrar el mínimo de dicha función de error se corre el riesgo de que el proceso de entrenamiento quede atrapado en un mínimo local [Sutton, 1986]. Esta situación no es deseable, fundamentalmente si dicho mínimo esta localizado lejos del mínimo global.

Existen algunos mecanismos para evitar que esto suceda. Una posible solución para evitar que el entrenamiento quede atrapado en un mínimo local es aumentar el número de neuronas ocultas de la red. Este mecanismo puede ayudar en aquellos casos en los que la red tiene escaso poder de representación interna, y no es capaz de distinguir entre dos patrones diferentes, proporcionando una misma salida para ambos patrones. Al aumentar el número de neuronas ocultas la red posee mayor cantidad de parámetros libres y puede conseguir una mejor representación interna.

Otros mecanismos que ayudan a disminuir los efectos de este problema son la adición de una tasa de momento al proceso de entrenamiento, utilizar una tasa de aprendizaje decreciente a lo largo del proceso, partir de otras configuraciones iniciales de la red, añadir ruido al método de gradiente, etc.

- **Parálisis**

El fenómeno de parálisis, también conocido como saturación, se produce cuando la entrada total a una neurona de la red toma valores muy altos, ya sean positivos o negativos. Al utilizar funciones de activación sigmoidales, la función de activación posee dos asíntotas horizontales. Si la entrada de la neurona alcanza un valor alto, la función de activación se satura y alcanza un valor de activación máximo o mínimo.

Cuando la función de activación se satura su derivada tiende a hacerse nula, haciendo que los parámetros de la red permanezcan invariables y, como consecuencia, la suma de los errores locales permanece constante por un largo periodo de tiempo [Kröse y van der Smagt, 1993]. Aunque esta situación se suele confundir con un mínimo local, pues el error permanece invariable, en este caso es posible que después de un cierto tiempo el error comience nuevamente a decrecer.

El fenómeno de parálisis del perceptrón multicapa ocurre fundamentalmente cuando los parámetros de la red toman valores muy altos. Un mecanismo para evitar esto consiste en partir de valores iniciales bajos.

- **Condiciones iniciales**

El conjunto de pesos iniciales de la red neuronal generalmente se selecciona de manera aleatoria. Sin embargo, el algoritmo backpropagation es muy dependiente de las condiciones iniciales seleccionadas [Kolen, 1991]. Pequeñas variaciones realizadas sobre las condiciones iniciales pueden llevar a grandes diferencias en el tiempo de convergencia del algoritmo.

Todos estos problemas muestran la necesidad de buscar un método alternativo que permita entrenar redes neuronales con conexión hacia delante. Este método no reemplazará los métodos existentes sino que será una alternativa a tener en cuenta por el diseñador de una red neuronal en el momento de seleccionar el algoritmo de entrenamiento.

## **5.2. Objetivo y marco de trabajo**

El método de entrenamiento deberá ser diseñado con el objetivo de eliminar los problemas que sufre el algoritmo backpropagation, o en su defecto deberá disminuir el efecto producido por éstos. El diseño de dicho método de entrenamiento deberá realizarse siguiendo las siguientes premisas:

- **Adaptabilidad**

El método deberá ser aplicable a distintas topologías sin necesidad de realizar grandes modificaciones sobre el mismo. De esta manera se busca reducir el tiempo de desarrollo de métodos de entrenamiento.

- **Baja dependencia de parámetros**

El método debe ser capaz de encontrar un conjunto de pesos sinápticos lo suficientemente bueno como para ser utilizado en problemas prácticos, sin que el éxito del mismo este fuertemente asociado a la configuración del método.

- **Mínimos locales**

El método debe realizar una buena exploración del espacio de soluciones para evitar quedar atrapado en mínimos locales. El método no necesariamente tiene que ser capaz de encontrar el mínimo global, pero si debe ser capaz de encontrar un mínimo local lo suficientemente bueno.

- **Parálisis**

El proceso de aprendizaje debe ser un proceso continuo. El método debe asegurar la continuidad del aprendizaje en los casos en que una neurona se sature.

- **Generalización**

Una característica deseable del método es que sea capaz de obtener un alto grado de generalización.

Siguiendo los lineamientos mencionados previamente se diseñará un nuevo método de entrenamiento, el cual se comparará con el algoritmo backpropagation. Los experimentos deberán resolver las siguientes cuestiones:

Cuestión: *¿Como influyen los parámetros del algoritmo backpropagation en el aprendizaje de la red?*

Cuestión: *¿De qué manera influyen los valores de inicialización en el aprendizaje utilizando el algoritmo backpropagation?*

Cuestión: *¿Cuál es la capacidad de generalización de las redes entrenadas por el método propuesto?*

Cuestión: *¿Cómo responde el método al fenómeno de saturación y a los mínimos locales?*

Cuestión: *¿Cómo se adapta el método a diferentes topologías?*

## Capítulo 6: Solución propuesta

El objetivo de este capítulo es desarrollar un método alternativo de entrenamiento de redes neuronales con conexión hacia delante, siguiendo los lineamientos propuestos en el capítulo anterior. El método propuesto deberá tener la capacidad de sobreponerse a los siguientes problemas:

- Adaptabilidad
- Dependencia de parámetros
- Mínimos locales
- Parálisis

Teniendo en cuenta los problemas mencionados se estudiará en este capítulo la posibilidad de entrenar redes neuronales basándose en la aplicación de algoritmos genéticos. Los algoritmos genéticos son una alternativa interesante ya que presentan las siguientes características:

- El espacio de búsqueda es muy grande y multimodal. Diferentes redes neuronales son capaces de lograr la misma performance **[Yao, 1996]**.
- A medida que crece la cantidad de ejemplos, y aumenta la complejidad de la topología, el espacio de búsqueda se vuelve cada vez más complejo, y la función de error contiene cada vez más mínimos locales. Los algoritmos genéticos presentan una alta eficiencia en espacios de búsqueda grandes y complejos, pudiendo encontrar soluciones próximas al mínimo global **[Montana, 1995]**, **[Adamidis, 1994]**.
- Otra importante razón para estudiar los algoritmos genéticos como método alternativo de entrenamiento de redes neuronales es su generalidad. El algoritmo genético requiere muy poca información acerca de la estructura del problema **[Jules, 1994]** y por lo tanto puede ser utilizado para entrenar diferentes variedades de redes neuronales, como redes recurrentes, redes con funciones de activación discontinuas, etc. **[Yao, 1993b]**.
- El proceso evolutivo es un proceso continuo. Los algoritmos evolutivos trabajan en paralelo sobre una población de posibles soluciones, reduciéndose las posibilidades de que el proceso se detenga en un mínimo local subóptimo **[Potter, 1992]**.

Durante el resto del capítulo se estudiarán las modificaciones necesarias a realizar sobre el algoritmo genético canónico para poder aplicarlo al entrenamiento de redes neuronales.

## 6.1. Algoritmo genético para entrenamiento de redes neuronales

El método propuesto consiste en una adaptación del algoritmo genético canónico. Se estudiará la manera de adaptar distintas variantes de los operadores clásicos para poder utilizar a los algoritmos genéticos como método de entrenamiento de redes neuronales.

### 6.1.1. Codificación de parámetros

Cada individuo (cromosoma) de la población se definirá como una cadena binaria. La longitud de dicha cadena dependerá de la topología de la red. Dentro del cromosoma cada peso de la red neuronal estará representado por 32 bits consecutivos. Así, la cantidad de genes dentro del cromosoma será proporcional a la cantidad de pesos (y umbrales) que tenga la red.

A lo largo de este capítulo se utilizará indistintamente el término peso y gen para facilitar la explicación de los distintos operadores. Sin embargo, es importante destacar que 32 genes serán la representación en el genotipo de un peso sináptico, existiendo una función que transforma un conjunto de 32 bits en el peso sináptico que representa (y viceversa).

### 6.1.2. Del genotipo al fenotipo

La transformación del genotipo al fenotipo será directa. Los pesos sinápticos que parten desde una misma neurona serán representados por genes contiguos en el genotipo. La figura 6.1 muestra gráficamente la transformación del genotipo al fenotipo.

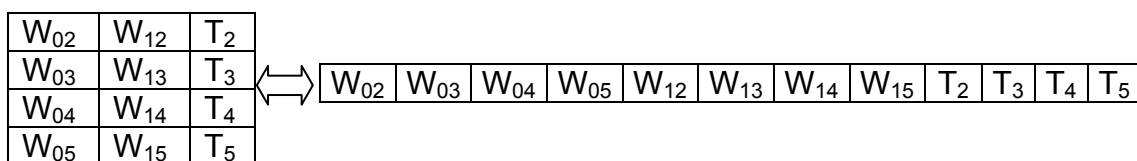


Figura 6.1. Morfogénesis

Este esquema de codificación del genotipo está directamente relacionado con el esquema de cruce que se utilizará, y que se describe en la sección 6.1.5.

### 6.1.3. Evaluación y aptitud

La función de evaluación que se utilizará en este método será la dada por la ecuación (6.1):

$$f_i = \frac{1}{e_i} \quad (6.1)$$

donde  $f_i$  es el valor de evaluación del individuo  $i$ , y  $e_i$  es el error cuadrático medio de la red neuronal que representa el individuo  $i$ . Esta función de evaluación aumenta cuando el error cuadrático medio disminuye. Así, se busca favorecer en mayor medida a los individuos que tienen menor error cuadrático medio sobre el set de entrenamiento.

Una vez evaluados todos los individuos de la población se procederá a calcular la aptitud de los mismos. En este caso se decidió no hacer uso de la función de aptitud clásica empleada por el algoritmo genético canónico, sino que se utilizará la función de aptitud dada por la ecuación (6.2):

$$F_i = 2 * \frac{f_{MIN}}{f_i} \quad (6.2)$$

Donde  $F_i$  es el valor de aptitud del individuo  $i$ ,  $f_i$  es el valor de evaluación del individuo  $i$ , y  $f_{MIN}$  es el valor de evaluación mínimo obtenido en las generaciones previas a la actual. De esta manera se busca asignarles mayores posibilidades de supervivencia a los individuos que superan la mejor performance de las generaciones previas.

#### 6.1.4. Operadores de selección

Se proponen para el estudio cuatro variantes del operador de selección. Todas estas variantes son operadores de selección clásicos, los cuales se utilizarán en combinación con un proceso de selección elitista. A continuación se realiza una descripción de cada uno de estos operadores:

##### 6.1.4.1. Selección elitista

Este operador se utilizará en combinación con el resto de los operadores propuestos. El objetivo de este esquema de selección es asegurar que los mejores individuos de una población pasen a la siguiente. El operador propuesto asigna una cantidad de copias igual a un porcentaje de la población, el cual varía de acuerdo a la posición del individuo:

Posición	Porcentaje
1	10%
2	5%
3	3%
4	2%
5	1,5%
6	1%

Tabla 6.1. Selección elitista

En todos los casos se asegura que al menos una copia de cada individuo pasará a la siguiente población, aun cuando el porcentaje corresponda a menos de una copia. El resto de los individuos de la población serán seleccionados por alguna otra variante del operador de selección.

#### **6.1.4.2. Selección proporcional**

Este operador asigna un número de copias proporcional a la aptitud de un individuo. El número de copias del individuo  $i$  que pasarán a la siguiente población se calcula en base a la siguiente ecuación:

$$c_i = N * \frac{F_i}{\sum_{i=1}^N F_i} \quad (6.3)$$

#### **6.1.4.3. Selección por ruleta**

El primer paso para la aplicación de este operador consiste en obtener aleatoriamente un número  $R$  de la siguiente manera:

$$R = \text{Random}(0, \sum_{i=1}^N F_i) \quad (6.4)$$

Luego se selecciona al individuo  $j$  tal que:

$$\sum_{i=1}^j F_i \leq R < \sum_{i=1}^{j+1} F_i \quad (6.5)$$

Este individuo pasa a la siguiente generación, repitiéndose el procedimiento tantas veces como sea necesario hasta completar la siguiente generación. Al contener una componente aleatoria, la cantidad de copias de un individuo que efectivamente pasan a la siguiente generación puede variar de la cantidad de copias esperadas.

#### **6.1.4.4. Selección por torneo**

Para este operador se definió un tamaño del torneo igual a dos. De esta manera se seleccionan aleatoriamente dos individuos de la población, pasando a la siguiente generación aquel individuo que tiene mayor aptitud. Este proceso se repite  $N$  veces, donde  $N$  es el tamaño de la población. La selección por torneo no asegura que el número de copias que pasan a la próxima generación sea igual al esperado.

#### **6.1.4.5. Selección por ranking**

Este operador asigna una probabilidad de selección proporcional a la aptitud del individuo. Sin embargo, la selección se realiza con un esquema



análogo al de selección por ruleta. Se asigna a un individuo una probabilidad de selección igual a:

$$p_i = \frac{N - i}{\sum_{i=1}^N i} \quad (6.6)$$

Luego se selecciona un número aleatorio R de la siguiente manera:

$$R = \text{Random}(0, \sum_{i=1}^N p_i) \quad (6.7)$$

Por último se elige al individuo j tal que:

$$\sum_{i=1}^j p_i \leq R < \sum_{i=1}^{j+1} p_i \quad (6.8)$$

Este individuo pasa a la siguiente generación. El procedimiento se repite tantas veces como sea necesario hasta completar la siguiente generación. Nuevamente, la cantidad de copias que se pasan a la siguiente generación puede diferir de la cantidad esperada con un esquema proporcional.

### 6.1.5. Operadores de cruza

Este operador se implementará directamente sobre el fenotipo, en contraste con los operadores de cruza clásicos utilizados en los algoritmos genéticos. En lugar de cruzar genes individuales se cruzarán neuronas completas. El objetivo de esta definición es mantener estructuras complejas, evitando que sean destruidas por el operador de cruza. Todos los pesos que parten de una misma neurona serán tratados como una unidad indivisible. La figura 6.2 muestra gráficamente los puntos de cruza para una red neuronal con dos neuronas de entrada y 4 neuronas de salida.

Neurona 0				Neurona 1				Umbrales			
W <sub>02</sub>	W <sub>03</sub>	W <sub>04</sub>	W <sub>05</sub>	W <sub>12</sub>	W <sub>13</sub>	W <sub>14</sub>	W <sub>15</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>

Figura 6.2. Puntos de cruza

Debido a esta definición será necesario darle gran importancia al operador de mutación, que será el encargado de explorar el espacio de soluciones, en búsqueda de mejores conjuntos de pesos sinápticos.

Todos los operadores de cruza que se estudiarán toman dos padres aleatoriamente de la población actual y generan dos hijos donde uno es el complemento del otro. A continuación se presentan las cuatro variantes a estudiar.

### **6.1.5.1. Cruza Uniforme**

Este operador de cruce asigna aleatoriamente los pesos entre puntos de cruce sucesivos. Cada hijo tiene una probabilidad de 0.5 de recibir los genes de su padre y por ende de recibirlos de su madre.

Para cada conjunto de pesos comprendido entre dos puntos de cruce sucesivos se selecciona aleatoriamente el origen de los genes del hijo 1 y se asignan a él. Luego, el hijo 2 tomará los genes del otro ancestro. La figura 6.3 muestra un ejemplo de cruce uniforme entre dos cromosomas. En el ejemplo se observa que el hijo 1 recibe dos conjuntos de pesos sucesivos del padre y el tercero de la madre. El segundo hijo se forma como complemento del primero.

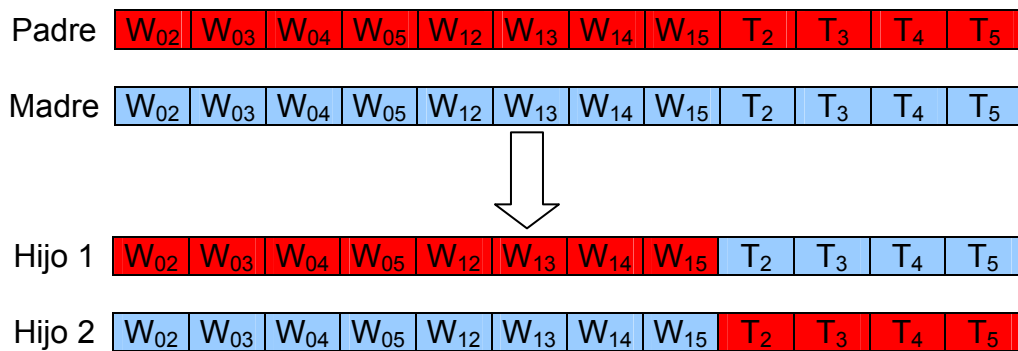


Figura 6.3. Cruza uniforme

### **6.1.5.2. Cruza Binomial**

Este operador es similar al operador de cruce uniforme, sólo que las probabilidades se definen en función de la aptitud de los padres de la siguiente manera:

$$\begin{cases} p_{padre} = \frac{f_{padre}}{f_{padre} + f_{madre}} \\ p_{madre} = 1 - p_{padre} \end{cases} \quad (6.9)$$

### **6.1.5.3. Cruza Simple**

En esta variante del operador de cruce se toma aleatoriamente un punto de cruce. Luego, a un hijo se le asignan todos los genes del padre ubicados a la izquierda del punto de cruce, y todos los genes de la madre ubicados a la derecha del punto de cruce. El segundo hijo es el complemento del primero. La figura 6.4 muestra un ejemplo de cruce simple en el cual se toma como punto de cruce el primero de ellos.

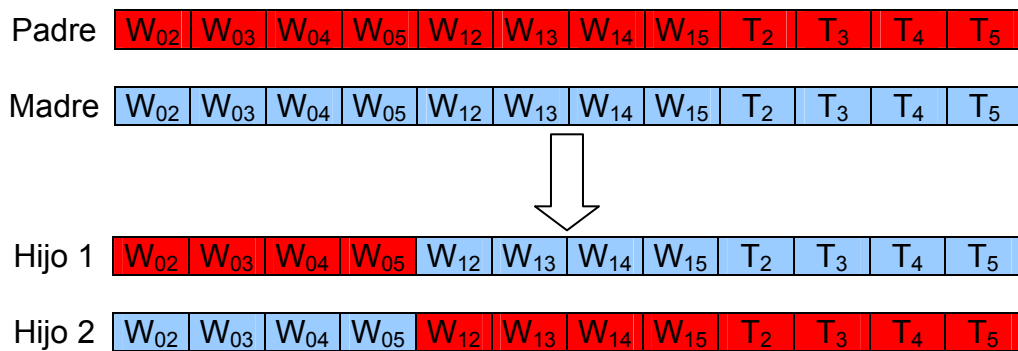


Figura 6.4. Cruza Simple

#### 6.1.5.4. Cruza Multipunto

Esta cuarta variante del operador de cruza es similar al operador de cruza simple, sólo que la cantidad de puntos de cruza se determina aleatoriamente. Cada hijo recibe los genes entre dos puntos de cruza sucesivos de cada uno de los padres, de manera intercalada. Cabe destacar que la cruza simple es un caso particular de la cruza multipunto, con un único punto de cruza. La figura 6.5 muestra un ejemplo de cruza multipunto, tomando dos puntos de cruza.

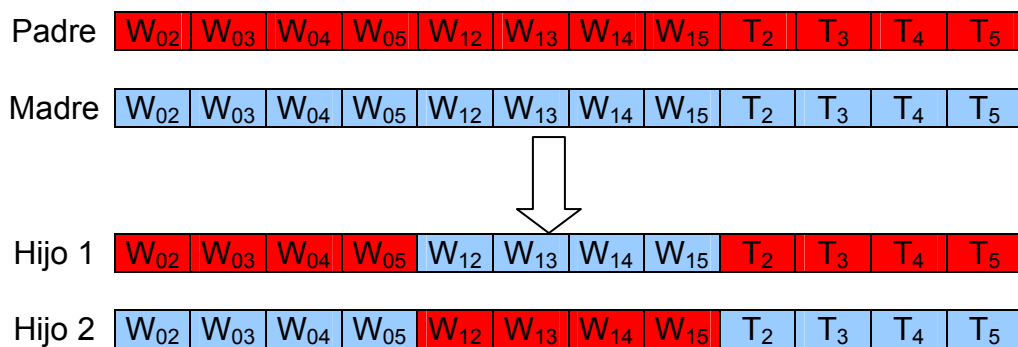


Figura 6.5. Cruza Multipunto

#### 6.1.6. Operadores de mutación

Para el operador de mutación se estudiarán dos variantes diferentes, la mutación binaria y la mutación numérica. En ambos casos, el operador de mutación actuará con una determinada probabilidad de mutación, la cual se explica más adelante.

La mutación binaria es el tipo de mutación tradicionalmente utilizado en el algoritmo genético canónico, y consiste en invertir un gen aleatoriamente. Dada la representación binaria del gen, se invierte el bit que lo representa con una determinada probabilidad, como lo muestra la figura 6.6.

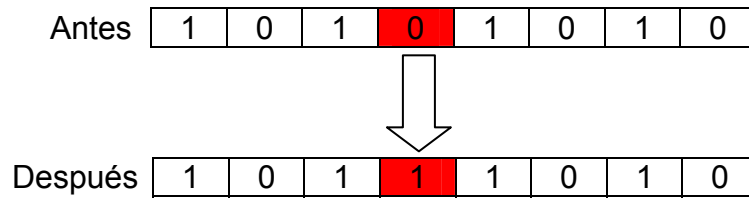


Figura 6.6. Mutación Binaria

La mutación numérica es otra variante del operador de mutación, implementada sobre el fenotipo. Se convierten los genes en el conjunto de pesos que representan y con una determinada probabilidad se les suman una pequeña cantidad aleatoria. La figura 6.7 muestra un ejemplo de mutación numérica.

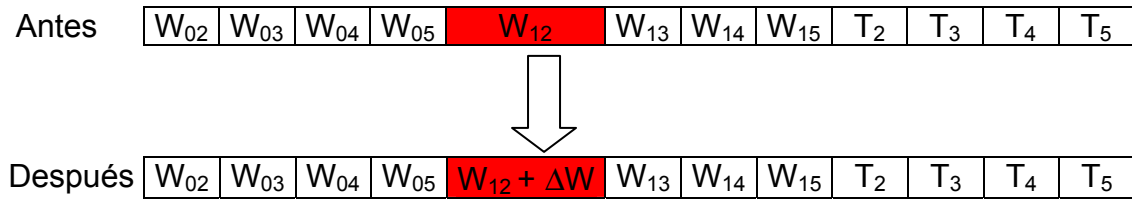


Figura 6.7. Mutación Numérica

Este tipo de mutación se desarrolló en base a observaciones realizadas sobre el algoritmo backpropagation. En dicho algoritmo cada peso es modificado proporcionalmente al gradiente de la función de error. De esta manera, a medida que la red se va acercando a la solución las modificaciones son menores. De acuerdo a lo dicho anteriormente se propone el esquema de variaciones de pesos dado por la ecuación 6.10.

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}(t) \quad (6.10)$$

donde

$$\Delta W_{ij}(t) = \text{Random}(-R, R) \quad (6.11)$$

y

$$|R| = \begin{cases} 0.6 - 0.0005 * t & \text{si } t \leq 1000 \\ 0.1 & \text{caso contrario} \end{cases} \quad (6.12)$$

En ambos casos el operador de selección será el encargado de determinar si el individuo resultante tiene el grado de adaptación suficiente para contribuir a la formación de la próxima generación.

Como se mencionó en el capítulo 4 el operador de mutación actúa con una probabilidad de mutación  $P_m$ . Cada uno de estos tipos de mutación se combinará con tres funciones de probabilidad de mutación distintas: uniforme,

adaptativa por temperatura ascendente, y adaptativa por temperatura descendente, y que se definen a continuación

#### **6.1.6.1. Mutación uniforme**

En este caso la probabilidad de mutación permanece constante a lo largo de las distintas generaciones, como lo muestra la ecuación 6.13.

$$P_m(t) = P_m(0) = P_m^{\max} = P_m^{\min} \quad (6.13)$$

#### **6.1.6.2. Mutación adaptativa por temperatura ascendente**

La probabilidad de mutación se va incrementando a medida que transcurren las generaciones. El objetivo de este aumento es mantener la diversidad de individuos en la población, que tiende a hacerse homogénea con el transcurso de las generaciones. La probabilidad de mutación para una generación esta dada por la ecuación 6.14.

$$P_m(t) = P_m^{\min} + \frac{(P_m^{\max} - P_m^{\min})}{T} * t \quad (6.14)$$

#### **6.1.6.3. Mutación adaptativa por temperatura descendente**

La probabilidad de mutación va decreciendo a medida que transcurren las generaciones. De esta manera se asegura una alta diversidad de individuos en las generaciones iniciales. La probabilidad mínima debe ser mayor a cero para permitir continuar la exploración del espacio de búsqueda a medida que las generaciones avanzan. La ecuación 6.15 muestra como se calcula la probabilidad de mutación para una determinada generación.

$$P_m(t) = P_m^{\max} - \frac{(P_m^{\max} - P_m^{\min})}{T} * t \quad (6.15)$$

#### **6.1.7. Operadores de inicialización**

La forma en que se crea la población inicial es una cuestión clave a resolver. Como se mencionó en el capítulo 4, la población inicial debería contener idealmente todos los alelos posibles que pueda tomar un gen. Dado que en la práctica esta situación no es factible, el método de inicialización debe asegurar la suficiente diversidad como para explorar todo el espacio de búsqueda. A continuación se presentan dos formas distintas de inicializar la población.

##### **6.1.7.1. Inicialización Random**

Este método de inicialización se implementa directamente sobre el fenotipo y consiste en generar aleatoriamente un conjunto de pesos. El valor de los

pesos esta dado por una distribución uniforme definida en un rango de valores que idealmente debería estar centrado en cero. Es decir,

$$W_{ij} = \text{Random}(\text{Ini}^{\min}, \text{Ini}^{\max}) \quad (6.16)$$

### **6.1.7.2. Inicialización Selectiva**

Este método es similar al anterior, sólo que una determinada cantidad de individuos es generada con valores previamente seleccionados.

Individuo	$W_{ij}$
1	0
2	$\text{Ini}^{\min}$
3	$\text{Ini}^{\max}$
4	$\text{Random}(0.5 * \text{Ini}^{\min} ; 0.5 * \text{Ini}^{\max})$
5	$\text{Random}(0.75 * \text{Ini}^{\min} ; 0.75 * \text{Ini}^{\max})$
6	$\text{Random}(0.25 * \text{Ini}^{\min} ; 0.25 * \text{Ini}^{\max})$
7	$0.5 * \text{Ini}^{\min}$
8	$0.5 * \text{Ini}^{\max}$

Tabla 6.2. Inicialización Selectiva

El resto de los individuos se genera con pesos dados por la ecuación 6.16

## **6.2. Problemas del método propuesto**

Durante la aplicación del método propuesto podrían aparecer algunos problemas, los cuales se describen a continuación. También se describe a continuación el tratamiento que se le dará a cada uno de ellos.

### **6.2.1. El problema de la convergencia prematura**

La convergencia prematura es un problema muy común en los algoritmos genéticos. Este problema se presenta cuando aparece en una generación un individuo con un nivel de adaptación muy superior al resto de la población [Korning, 1995]. En este caso, el súper-individuo domina rápidamente la población, antes de que se haya podido realizar una exploración suficientemente amplia del espacio de búsqueda. Esta característica se presenta fundamentalmente cuando se hace uso de operadores de selección proporcionales, donde el súper-individuo contribuye con gran cantidad de copias a la siguiente generación. Así, la población tiende a hacerse homogénea y el algoritmo genético se estabiliza en una solución que puede no ser lo suficientemente buena.

Para evitar este problema se estudiará el uso de operadores de selección alternativos a la selección proporcional, donde todos los individuos tengan posibilidades de pasar a la siguiente generación. También, para garantizar la

heterogeneidad de la población, se hará uso de probabilidades de mutación superiores a las utilizadas normalmente en algoritmos genéticos.

### 6.2.2. El problema de la permutación

El problema de la permutación es un tema tratado muy frecuentemente en trabajos sobre diseño de topologías de redes neuronales mediante algoritmos genéticos. El origen de este problema se encuentra en la disposición de las neuronas ocultas de una red neuronal [Liu y Yao, 1997]. Dos redes neuronales pueden ser funcionalmente equivalentes independientemente del orden de sus neuronas ocultas, como se muestra en la figura 6.8.

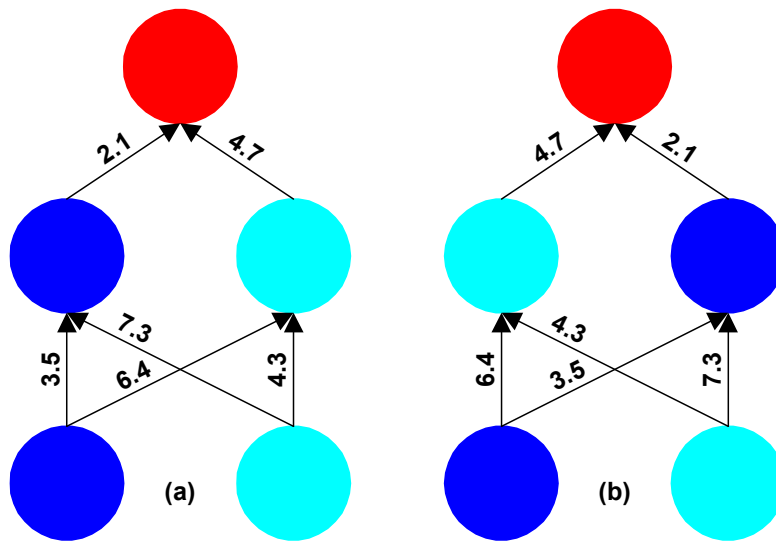


Figura 6.8. El problema de la permutación

Dado que una red neuronal puede tener varios genotipos que la representan, la probabilidad de que el operador de cruce produzca de ellos un individuo más adaptado es baja. Por esto, varios trabajos no hacen uso de este operador, a pesar de la importancia del mismo [Whitley, 1995].

Sin embargo, el efecto de este problema no es tan severo como se supone [Hancock, 1992]. Por ello, en este trabajo se tratará a los individuos que presentan este problema como sinónimos, dejando que el operador de selección sea el encargado de decidir que individuos contribuyen a la próxima generación. A su vez, el tamaño de la población se seleccionará de manera de minimizar los efectos de este problema en el operador de cruce.

En base a lo dicho a lo largo de este capítulo se deberán diseñar los experimentos de manera tal que respondan a las siguientes cuestiones:

Cuestión: *¿Cuál es el operador de selección adecuado para entrenar redes neuronales utilizando algoritmos genéticos?*

Cuestión: *¿Cuál es el operador de cruce adecuado para entrenar redes neuronales utilizando algoritmos genéticos?*

Cuestión: *¿Cual es la manera más efectiva de realizar la mutación para entrenar redes neuronales utilizando algoritmos genéticos? ¿Cuál es el operador de mutación adecuado para entrenar redes neuronales?*

Cuestión *¿De qué manera influyen los valores de inicialización en el aprendizaje utilizando el método propuesto? ¿Cuál es la ventaja de utilizar un esquema de inicialización selectivo?*

Cuestión: *¿Cuál es la ventaja del método propuesto en cuanto a los tiempos de entrenamiento?*

### **6.3. Cuestiones a resolver en esta tesis**

El propósito de esta tesis es establecer una medida de comparación entre los métodos tradicionales de entrenamiento y el método de entrenamiento propuesto en este capítulo. Dentro de este contexto, las cuestiones que se plantean a lo largo de esta tesis y se deberán resolver en base a los resultados experimentales son las siguientes:

Cuestión 1: *¿Como influyen los parámetros del algoritmo backpropagation en el aprendizaje de la red?*

Cuestión 2: *¿De qué manera influyen los valores de inicialización en el aprendizaje utilizando el algoritmo backpropagation?*

Cuestión 3: *¿Cuál es el operador de selección adecuado para entrenar redes neuronales utilizando algoritmos genéticos?*

Cuestión 4: *¿Cuál es el operador de cruza adecuado para entrenar redes neuronales utilizando algoritmos genéticos?*

Cuestión 5: *¿Cual es la manera más efectiva de realizar la mutación para entrenar redes neuronales utilizando algoritmos genéticos? ¿Cuál es el operador de mutación adecuado para entrenar redes neuronales?*

Cuestión 6: *¿De qué manera influyen los valores de inicialización en el aprendizaje utilizando el método propuesto? ¿Cuál es la ventaja de utilizar un esquema de inicialización selectivo?*

Cuestión 7: *¿Cuál es la ventaja del método propuesto en cuanto a los tiempos de entrenamiento?*

Cuestión 8: *¿Cuál es la capacidad de generalización de las redes entrenadas por el método propuesto?*

Cuestión 9: *¿Cómo responde el método al fenómeno de saturación y a los mínimos locales?*

Cuestión 10: *¿Cómo se adapta el método a diferentes topologías?*



## Capítulo 7: Experimentación

En este capítulo se presentan los resultados experimentales obtenidos por ambos métodos de entrenamiento. La sección 7.1 describe las características generales de los experimentos realizados, la manera en que se diseñaron los experimentos, y el objetivo perseguido por éstos. Luego, en la sección 7.2 se presentan los resultados obtenidos. Los resultados experimentales permitirán establecer una medida de comparación entre ambos métodos y determinar las características principales de cada uno de ellos.

### 7.1. *Diseño experimental*

Los experimentos realizados fueron diseñados con el objetivo de establecer las siguientes tres características:

1. Determinar la configuración óptima del algoritmo backpropagation (sección 7.2.1)
2. Determinar la configuración óptima del algoritmo de entrenamiento propuesto (sección 7.2.2)
3. Establecer una comparación directa entre ambos métodos (sección 7.2.3)

Para cada configuración en particular de un método de entrenamiento se realizaron un total de 50 experimentos. Se seleccionó esta cantidad de experimentos ya que permite determinar con precisión el tiempo promedio de entrenamiento de un método, así como también el grado de variación de los resultados con respecto a dicho valor promedio. Esta afirmación se hace extensiva al resto de las variables analizadas.

Los resultados obtenidos se analizaron de la siguiente manera: se tomó como valor del método para una variable en particular al valor promedio sobre el total de experimentos realizados (50 corridas). Se utilizó el desvío estándar para determinar el grado de variación de los valores de la variable con respecto del valor medio.

Los valores individuales de cada experimento también se analizaron con el objetivo de realizar comparaciones entre distintas configuraciones. En este caso se estudió cual es el mejor valor obtenido por cada configuración. Sin embargo, este valor no fue tomado como un valor general del método ya que en muchos casos se presentaron grandes variaciones entre los valores obtenidos. Sólo fueron considerados para tener una medida del “mejor caso”.

Los resultados presentados a lo largo de este capítulo fueron obtenidos tras más de 400 horas de entrenamiento. A continuación se presenta un resumen del tiempo total de entrenamiento realizado por cada método, de acuerdo al objetivo de entrenamiento perseguido por el experimento:

Tipo Algoritmo Entrenamiento	Objetivo Entrenamiento	Tiempo
Algoritmo genético	Evaluación de operadores del algoritmo genético - Selección	15.22
	Evaluación de operadores del algoritmo genético - Cruza	11.91
	Evaluación de operadores del algoritmo genético - Mutación	116.25
	Parámetros de inicialización del algoritmo genético	35.96
	Comparación de método	5.80
	<b>Total</b>	<b>185.13</b>
Red Neuronal	Parámetros óptimos de la red neuronal	179.12
	Parámetros de inicialización de la red neuronal	48.98
	Comparación de método	9.54
	<b>Total</b>	<b>237.63</b>
<b>Total</b>		<b>422.76</b>

**Tabla 7.1** Distribución de los tiempos de entrenamiento.

### 7.1.1. El algoritmo backpropagation

El algoritmo Backpropagation se utilizó como medida de comparación para constatar la calidad del método propuesto. En la primera serie de experimentos se estudió el conjunto de parámetros que minimizan los tiempos de entrenamiento de este método. Los parámetros de configuración estudiados durante esta serie de experimentos fueron:

- Tasa de aprendizaje
- Tasa de momento
- Rango de valores iniciales.

Los parámetros óptimos obtenidos durante esta primera etapa fueron utilizados luego para comparar los resultados obtenidos por este método con los resultados obtenidos por el algoritmo genético.

El primer paso en la aplicación de este método consiste en la generación aleatoria del conjunto de pesos de la red neuronal. El valor de inicialización esta dado por una distribución uniforme, y cuyo rango de valores óptimos es un parámetro en estudio.

Luego se procede a la aplicación del método utilizando las tasas de aprendizaje y de momento seleccionadas. La aplicación del algoritmo continuará hasta que la red alcance un nivel de error menor al umbral prefijado o cuando se hayan realizado una cantidad de ciclos determinada. Si la red alcanza un nivel de error menor al umbral prefijado se considera que el método fue capaz de entrenar la red. Caso contrario, se considera que la red no pudo ser entrenada.

### 7.1.2. El algoritmo genético

La segunda serie de experimentos tuvo como objetivo determinar cuales son los operadores genéticos y los parámetros de configuración que minimizan los tiempos de entrenamiento empleados por el algoritmo genético. Los parámetros y/o operadores estudiados en este caso fueron:

- Función de selección
- Función de cruza
- Tipo de mutación
- Función de mutación y probabilidad de mutación
- Función de inicialización y rango de valores iniciales

Dependiendo de los parámetros seleccionados la aplicación del método tendrá características propias. Sin embargo, una definición general del método es la siguiente:

1. Generar una población inicial.
2. Seleccionar los mejores individuos de la población
3. Cruzar los individuos de la población
4. Mutar aleatoriamente los individuos de la población
5. Evaluar a los individuos de la población resultante.
6. Si el nivel de error de la población es mayor al umbral prefijado, volver al paso 2

Los parámetros óptimos obtenidos durante esta etapa fueron utilizados luego para comparar los resultados obtenidos por este método con los resultados obtenidos por el algoritmo backpropagation.

### 7.1.3. El set de entrenamiento

El set de entrenamiento que se utilizará durante la evaluación del método propuesto debe presentar las siguientes características:

1. Debe poseer un elevado numero de ejemplos.
2. Los ejemplos deben ser variados, y los atributos multivaluados.
3. El problema a resolver debe ser una aplicación real, de compleja resolución.

Las características mencionadas previamente responden a la necesidad de evaluar el método propuesto sobre aplicaciones reales. Al poseer un elevado número de ejemplos se podrá evaluar también la capacidad de generalización de la red obtenida.

El set de datos utilizado a lo largo de los experimentos realizados consiste en 690 solicitudes de tarjetas de créditos. Este problema, conocido como Australian Credit Card Problem, es altamente utilizado para la evaluación de métodos de entrenamiento **[Blake y Merz, 1998]**.

El set de entrenamiento esta compuesto de 690 ejemplos, cada uno de los cuales es una solicitud. Cada ejemplo posee 14 atributos que describen las

características del solicitante, de los cuales 8 poseen valores discretos y 6 poseen valores numéricos. La salida de cada ejemplo puede pertenecer a una de dos clases, que indica la aprobación o no de la solicitud.

Para facilitar el entrenamiento de la red neuronal, y disminuir los tiempos de entrenamiento, se realizó un preprocesamiento de estos ejemplos. A los atributos que toman valores continuos se los normalizó dentro del intervalo [0,1]. A los atributos que toman valores discretos se los dividió en N atributos, donde N representa la cantidad de valores que puede tomar el atributo. De esta manera se obtiene un set de datos de 690 ejemplos, con 47 atributos entradas y una única salida.

Para poder evaluar las características de la red obtenida se particionó el set de datos en dos subsets: el subset de entrenamiento y el subset de evaluación. El primer subset se utilizó para realizar el entrenamiento de la red, mientras que el segundo subset se utilizó para evaluar la red obtenida. El particionamiento se realizó de manera tal que los subsets resultantes sean representativos, es decir, que posean las mismas características que el set original.

Para la primera y segunda serie de experimentos se utilizó el mismo particionamiento del set de datos. Se toman aleatoriamente el 65% de los ejemplos para entrenamiento y se dejan un 35% de ejemplos para evaluación.

Para realizar la comparación entre ambos métodos se particionó nuevamente el set de datos, tomando en este caso el 80 % de los ejemplos para entrenamiento. A continuación se presentan las características de cada set de datos:

	CANTIDADES			PORCENTAJES		
	Total	Clase 1	Clase 2	%	% C1	% C2
<b>Dataset</b>	690	307	383	100.0%	44.5%	55.5%
<b>Ent. 1</b>	448	199	249	64.9%	44.4%	55.6%
<b>Eval. 1</b>	242	108	134	35.1%	44.6%	55.4%
<b>Ent. 2</b>	552	246	306	80.0%	44.6%	55.4%
<b>Eval. 2</b>	138	61	77	20.0%	44.2%	55.8%

**Tabla 7.2** Características de los sets de datos

#### 7.1.4. Evaluación del método

La evaluación del método propuesto se realizó comparando su performance con la obtenida por el algoritmo backpropagation. Se tomó como medida de comparación el error cuadrático medio por dos motivos. En primer lugar, el error cuadrático medio es inversamente proporcional a la performance de la red. Una red con bajo error cuadrático medio tiene una buena performance. El segundo motivo por el que se selecciono esta unidad de medida es que puede ser calculada fácilmente en ambos métodos, sin introducir distorsiones sobre el valor obtenido.

Dado que ambos métodos trabajan con ciclos iterativos distintos (las redes neuronales lo hacen con epochs mientras que los algoritmos genéticos trabajan con generaciones) se seleccionó como unidad de medida para comparar tiempos entre ambos métodos al segundo. De esta manera, en todas las estadísticas se registran el tiempo en segundos desde el comienzo del

entrenamiento, permitiendo realizar una comparación directa entre ambos métodos.

Para poder establecer una medida real de comparación, al algoritmo backpropagation se le aplicarán el conjunto de técnicas para la aceleración del algoritmo descritas en el capítulo 3.

### 7.1.5. Parámetros de los algoritmos

#### 7.1.5.1. Parámetros fijos

Estos parámetros permanecieron invariables a lo largo de los experimentos:

- **Cantidad de generaciones.** Máxima cantidad de generaciones del algoritmo genético. Se tomó un valor máximo de 5000 generaciones
- **Cantidad de individuos por generación.** Cantidad de individuos que componen cada generación del algoritmo genético. Se tomó un valor de 30 individuos.
- **Probabilidad de cruza.** Determina la probabilidad de que dos individuos se crucen e intercambien su material genético. Este parámetro está directamente relacionado con el salto generacional, que es la cantidad de individuos nuevos que conforman la población. Se tomó un valor de 1, por lo que entre dos generaciones sucesivas se renuevan completamente los individuos que la componen.
- **Cantidad de ciclos o epochs.** Máxima cantidad de ciclos del algoritmo Backpropagation. Se tomó un valor máximo de 1000 epochs.
- **Error máximo admisible.** Error cuadrático medio a partir del cual una red puede considerarse entrenada. Se tomó valor de 0.05 para las primeras dos series de experimentos y de 0.07 para la tercer serie.
- **Umbral de acierto.** Es la diferencia máxima que puede existir entre el valor deseado y el valor obtenido por la red para que el ejemplo pueda ser considerado como acierto. Se tomó un valor máximo de 0.35.

#### 7.1.5.2. Variables independientes

Estas variables fueron modificadas de acuerdo al objetivo de cada experimento:

- **Función de selección.** Indica cual es la función de selección que se utiliza en el algoritmo genético.
- **Función de cruza.** Denota cual es la función de cruza que se utiliza en el algoritmo genético.
- **Tipo de mutación.** Determina la manera en que se aplica el operador mutación.
- **Función de mutación.** Indica cual es la función de mutación que se utiliza en el algoritmo genético.

- **Función de inicialización.** Define de que manera se conforma la generación inicial del algoritmo genético.
- **Probabilidad de mutación.** Determina cual es la probabilidad de aplicar el operador de mutación.
- **Valor de inicialización.** Define el rango de valores dentro del cual se generarán aleatoriamente los pesos iniciales de un cromosoma o una red neuronal.
- **Tasa de aprendizaje.** Indica el tamaño del paso tomado en cada iteración. Define como se realizará el ajuste de los pesos entre ciclos sucesivos.
- **Tasa de momento.** Permite disminuir el impacto de las oscilaciones en torno al punto mínimo cuando se utilizan tasas de aprendizaje altas.

#### **7.1.5.3. Variables dependientes**

El siguiente conjunto de variables depende de la configuración realizada sobre el algoritmo de entrenamiento. Se utilizan para medir la performance del método aplicado:

- **Tiempo de entrenamiento.** Tiempo que transcurre desde el inicio del entrenamiento hasta el fin del mismo.
- **Error.** Es el error cuadrático medio de una red neuronal. Para el caso del algoritmo genético se toma el mínimo de la generación.
- **Porcentaje de aciertos.** Cantidad de ejemplos bien clasificados sobre el total de ejemplos evaluados.

#### **7.1.6. Tipos de gráficos**

Para mostrar los resultados obtenidos a lo largo de los experimentos realizados se utilizarán dos tipos de gráficos.

- **Gráficos de tendencia del error de entrenamiento:** muestran, en función del tiempo, la tendencia logarítmica del error de entrenamiento. El error de entrenamiento se define como el error promedio obtenido para un instante de tiempo por las distintas corridas del método en análisis.
- **Gráficos de error de entrenamiento:** muestran, en función del tiempo, la evolución del error de entrenamiento promedio.
- **Gráficos de aciertos.** Muestran el porcentaje de ejemplos correctamente clasificados sobre el total de ejemplos evaluados. Se considera que un ejemplo es clasificado correctamente si la diferencia entre la salida deseada y la salida obtenida es menor al umbral de acierto.
- **Gráfico de tiempos individuales de entrenamiento:** Muestra para cada experimento en particular el tiempo total de entrenamiento.

## 7.2. Experimentos

### 7.2.1. Red Neuronal

El objetivo de esta serie de experimentos es determinar cuales son los parámetros óptimos que permitan entrenar una red neuronal para el problema en cuestión. Para este grupo de experimentos se utilizó el primer set de datos.

#### 7.2.1.1. Tasas de aprendizaje y momento óptimos

##### **Descripción:**

Las tasas de aprendizaje y de momento determinan de qué manera se realizan las actualizaciones de los pesos entre iteraciones sucesivas. De acuerdo a las características de la topología y de los datos estas constantes suelen variar de problema en problema.

##### **Variables:**

Para la realización de los experimentos se utilizaron los siguientes parámetros:

*Tasa de aprendizaje ( $\alpha$ )* = {0.01, 0.05, 0.1, 0.25}

*Taza de momento ( $\beta$ )* = {0, 0.01, 0.05, 0.1}

##### **Resultados:**

Alfa	Beta	Cant. Corridas	% Entrenamiento	Tiempo Prom. Ent.	Desvío Tiempo Ent.
0.01	0	50	100.0%	534	165
	0.01	50	100.0%	592	274
	0.05	50	100.0%	580	207
	0.1	50	100.0%	542	179

**Tabla 7.3** Tiempos de entrenamiento. Tasa de aprendizaje = 0.01

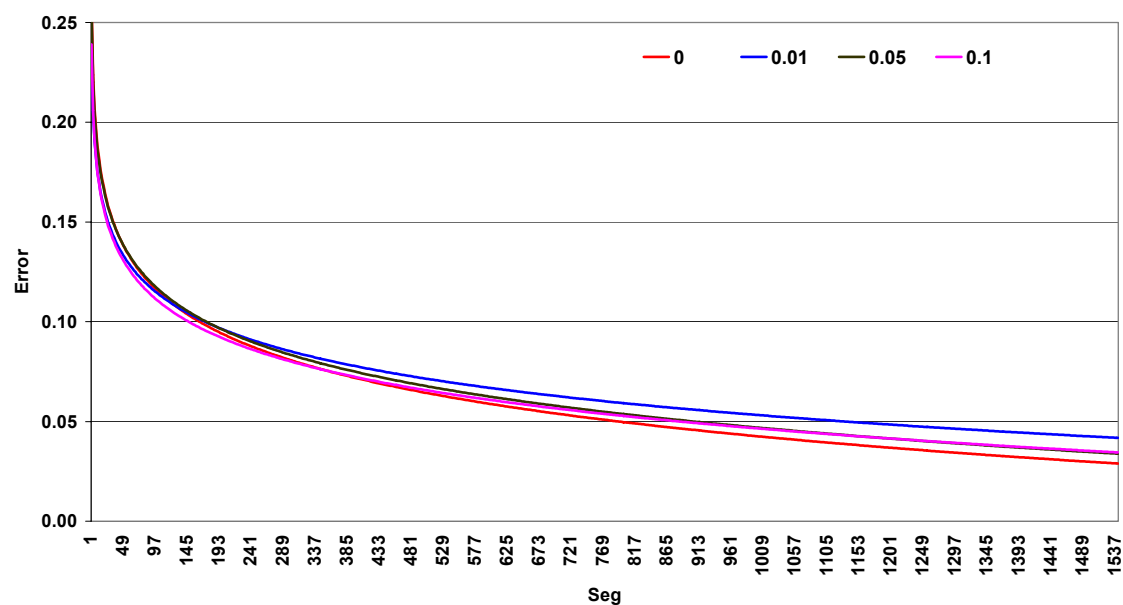


Gráfico 7.1 Tendencia del error de entrenamiento. Tasa de aprendizaje = 0.01

Alfa	Beta	Cant. Corridas	% Entrenamiento	Tiempo Prom. Ent.	Desvío Tiempo Ent.
0.05	0	50	100.0%	324	189
	0.01	50	100.0%	269	92
	0.05	50	98.0%	368	349
	0.1	50	100.0%	296	193

Tabla 7.4 Tiempos de entrenamiento. Tasa de aprendizaje = 0.05

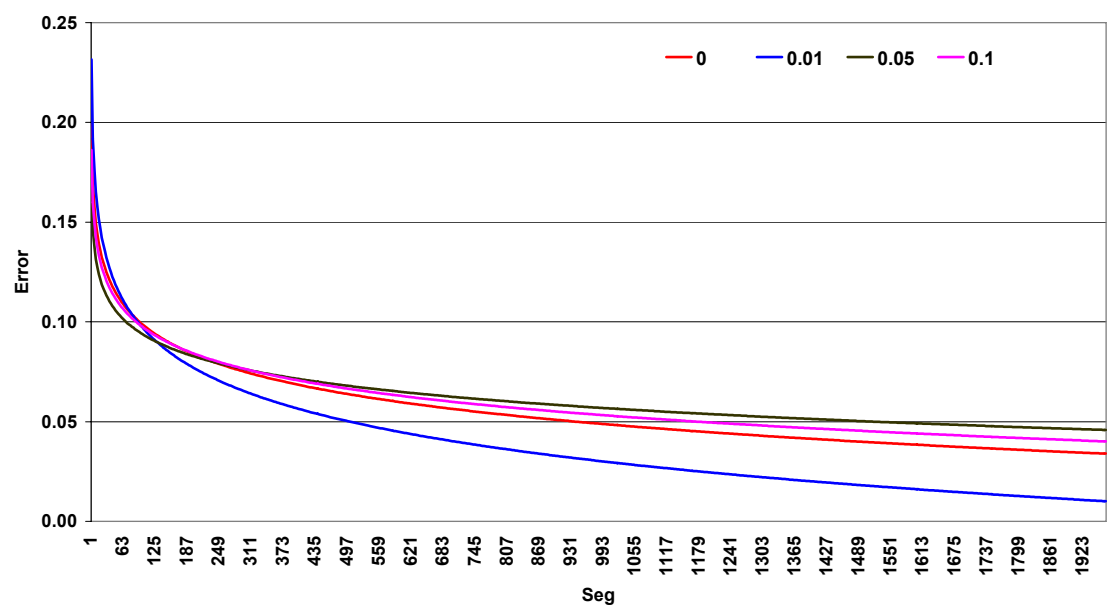
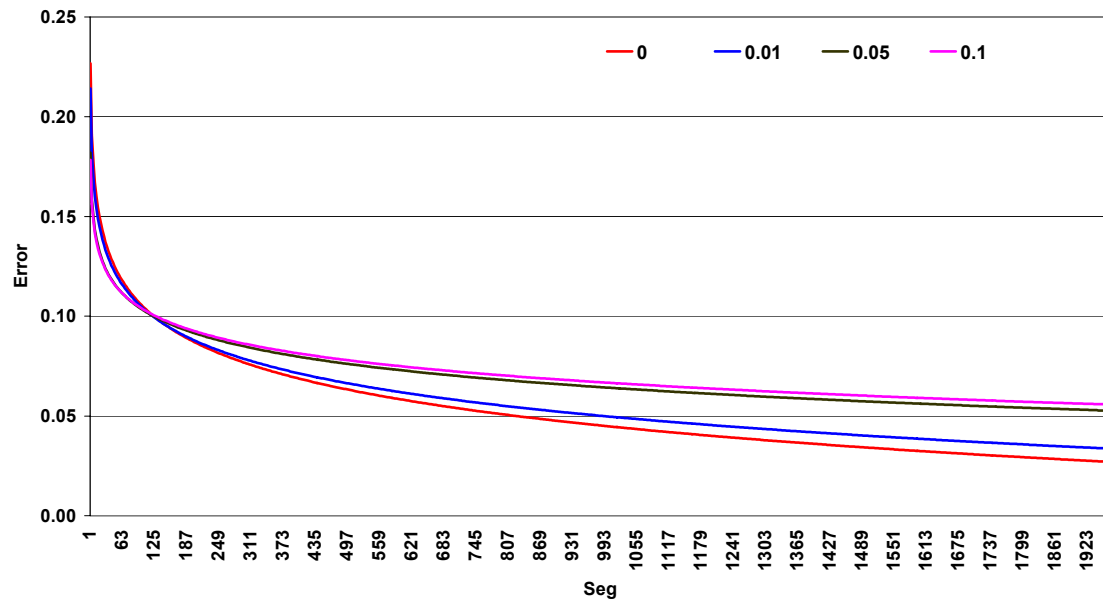


Gráfico 7.2 Tendencia del error de entrenamiento. Tasa de aprendizaje = 0.05

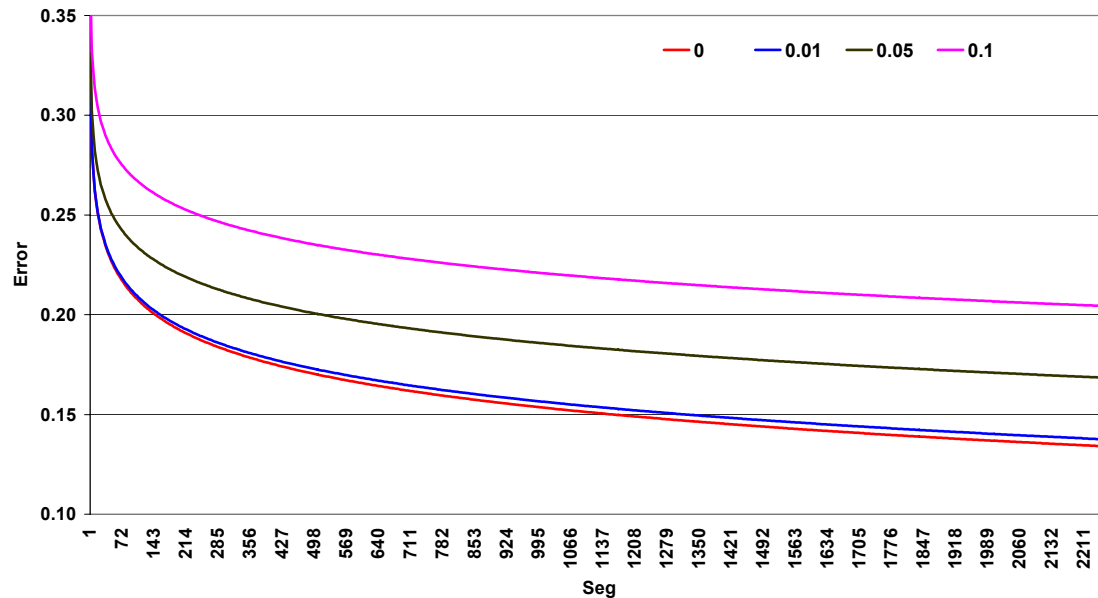


Alfa	Beta	Cant. Corridas	% Entrenamiento	Tiempo Prom. Ent.	Desvío Tiempo Ent.
0.1	0	50	100.0%	363	141
	0.01	50	100.0%	331	159
	0.05	50	98.0%	376	300
	0.1	50	98.0%	368	265

**Tabla 7.5** Tiempos de entrenamiento. Tasa de aprendizaje = 0.1

**Gráfico 7.3** Tendencia del error de entrenamiento. Tasa de aprendizaje = 0.1

Alfa	Beta	Cant. Corridas	% Entrenamiento	Tiempo Prom. Ent.	Desvío Tiempo Ent.
0.25	0	50	0.0%	1,980	35
	0.01	50	0.0%	1,989	52
	0.05	50	0.0%	1,990	25
	0.1	50	0.0%	1,999	41

**Tabla 7.6** Tiempos de entrenamiento. Tasa de aprendizaje = 0.25



**Gráfico 7.4** Tendencia del error de entrenamiento. Tasa de aprendizaje = 0.25

Alfa	Beta	Corrida	Tiempo Prom. Ent.
0.1	0.01	14	137
0.05	0	14	138
	0.05	17	139
	0.01	7	145
		26	148
	0.1	28	148
	0.01	25	153
0.1	0.1	32	155
0.05	0.05	50	156
	0.01	42	157
	0.1	29	157

**Tabla 7.7** Mejores tiempos individuales de entrenamiento.

#### Interpretación de los resultados:

- El uso de tasas de aprendizaje bajas lleva a una convergencia lenta del algoritmo. Sin embargo, permiten un descenso más estable sobre la superficie de error permitiendo localizar mejores mínimos. El porcentaje de convergencia a una solución es alto (Tabla 7.3, Gráfico 7.1).
- Al utilizar tasas de aprendizaje altas el algoritmo oscila alrededor de un mínimo, y no se estabiliza en una solución. El porcentaje de convergencia a una solución es bajo (Tabla 7.6, Gráfico 7.4).
- Al utilizar tasas de aprendizaje altas es preferible utilizar tasas de momentos bajas. Esto aumenta la estabilidad del algoritmo. Sin embargo los resultados obtenidos siguen siendo malos (Gráfico 7.4).
- Los mejores tiempos de entrenamiento individuales se obtuvieron utilizando las tasas de aprendizaje 0.05 y 0.1 (Tabla 7.6).

- Los mejores tiempos de entrenamiento promediados sobre el total de ejecuciones del algoritmo se obtuvieron utilizando una tasa de aprendizaje de 0.05 y una tasa de momento de 0.01 (Tabla 7.4).

### 7.2.1.2. Valores de inicialización óptimos

#### Descripción:

Los experimentos presentados en esta sección tienen como objetivo encontrar cual es el rango de valores de inicialización que disminuye los tiempos de entrenamiento.

#### Variables:

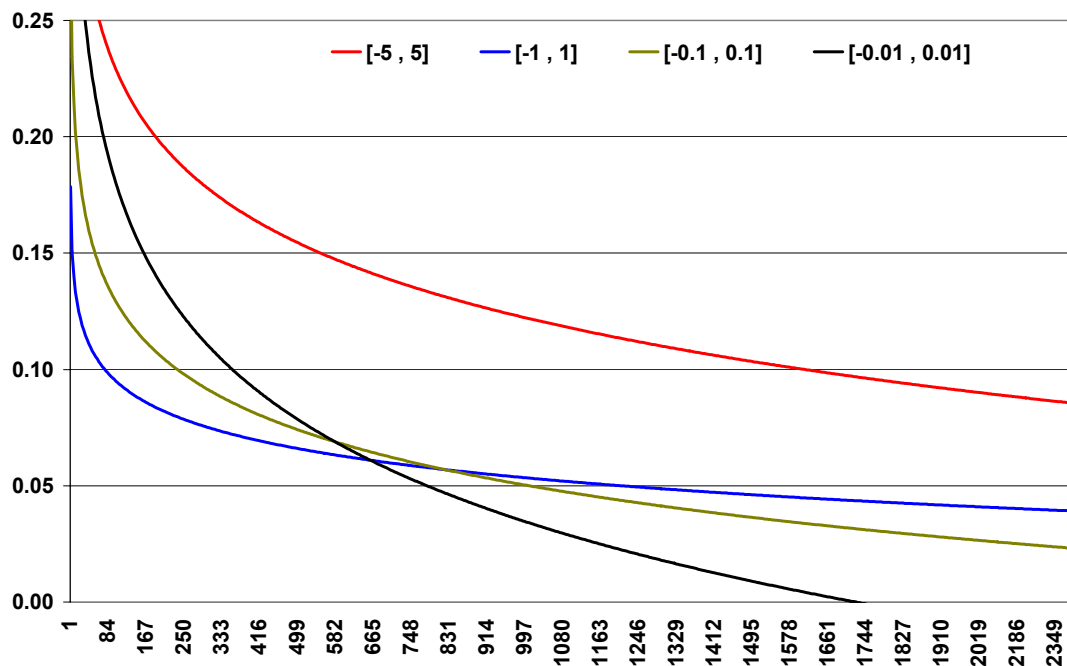
Los rangos estudiados fueron los siguientes:

Rangos =  $\{[-0.01, 0.01]; [-0.1, 0.1]; [-1, 1]; [-5, 5]\}$

#### Resultados:

Min Inicialización	Max Inicialización	Cant. Corridas	% Entrenamiento	Tiempo Prom. Ent.	Desvío Tiempo Ent.
-0.01	0.01	50	100.0%	682	128
-0.1	0.1	50	100.0%	559	111
-1	1	50	100.0%	306	211
-5	5	50	0.0%	1,980	66

**Tabla 7.8** Tiempos de entrenamiento. Valores de inicialización de la red neuronal



**Gráfico 7.5** Tendencia del error de entrenamiento. Valores de inicialización de la red neuronal

Min Inicialización	Max Inicialización	Corrida	Tiempo Prom. Ent.
-1	1	32	124
		41	136
		16	141
		3	152
		24	165
		10	167
		49	173
		7	187
		31	194
		38	194

**Tabla 7.9** Mejores tiempos individuales de entrenamiento.

**Interpretación de los resultados:**

- El uso de valores de inicialización muy altos evita que la red sea capaz de alcanzar un nivel de entrenamiento adecuado (Tabla 7.8, Gráfico 7.5).
- El uso de valores de inicialización muy bajos aumenta el tiempo necesario de entrenamiento (Tabla 7.8).
- Los mejores tiempos de entrenamiento promediados sobre el total de ejecuciones se obtuvieron para el rango de inicialización  $[-1,1]$ . Esta característica se observa también si se analizan cada una de las ejecuciones (Tabla 7.8, Tabla 7.9).

### 7.2.2. Algoritmos genéticos

El objetivo de esta serie de experimentos es determinar cuales son los parámetros de configuración del algoritmo genético óptimos que permitan entrenar una red neuronal para el set de datos analizado. Para este grupo de experimentos se utilizó el primer set de datos.

#### 7.2.2.1. Función de selección

**Descripción:**

El objetivo del operador de selección es elegir los individuos de la población que tengan mayor grado de adaptación.

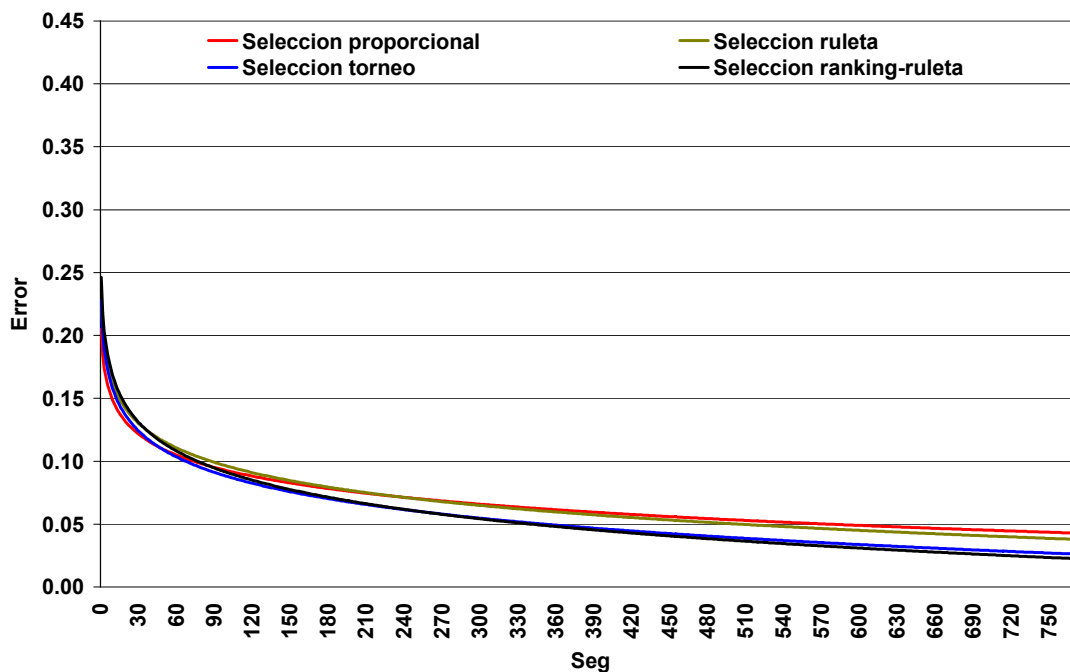
**Variables:**

A lo largo de esta sección se analizan cuatro funciones de selección diferentes:

$S = \{\text{Proporcional, Ruleta, Torneo, Ranking-Ruleta}\}$

**Resultados:**

Función Selección	Cant. Corridas	% Entrenamiento	Tiempo Prom. Ent.	Desvío Tiempo Ent.
Selección proporcional	50	96.0%	296	155
Selección ruleta	50	100.0%	325	112
Selección torneo	50	100.0%	208	79
Selección ranking-ruleta	50	100.0%	267	78

**Tabla 7.10** Tiempos de entrenamiento. Función de selección**Gráfico 7.6** Tendencia del error de entrenamiento. Función de selección

Función Selección	Corrida	Tiempo Prom. Ent.
Selección torneo	32	129
	46	134
Selección proporcional	26	136
	40	140
Selección torneo	33	144
Selección proporcional	48	147
Selección torneo	43	148
	16	149
Selección proporcional	20	150
Selección ranking-ruleta	4	153

**Tabla 7.11** Mejores tiempos individuales de entrenamiento.**Interpretación de los resultados:**

- Los tiempos promedio obtenidos por todas las funciones de selección analizadas son relativamente similares (Tabla 7.10, Gráfico 7.6).

- Los mejores tiempos individuales se obtuvieron utilizando los métodos de selección por torneo y selección proporcional (Tabla 7.10). Sin embargo esta última no fue capaz de alcanzar el nivel de entrenamiento deseado en 2 ejecuciones (Tabla 7.10).
- Los métodos de selección por torneo y ranking-ruleta muestran tiempos de convergencia bastantes similares a lo largo de las distintas ejecuciones. Esto se ve reflejado en el bajo desvío estándar que muestran sobre el total de corridas (Tabla 7.10).
- La función de selección por torneo obtuvo, en promedio, los mejores tiempos de entrenamiento (Tabla 7.10).
- Los resultados muestran que el método de entrenamiento es poco dependiente del operador de selección utilizado (Tabla 7.10, Gráfico 7.6).

#### **7.2.2.2. Función de cruza**

##### **Descripción:**

El objetivo del operador de cruza es recombinar el material genético de los individuos de una población para producir nuevos individuos.

##### **Variables:**

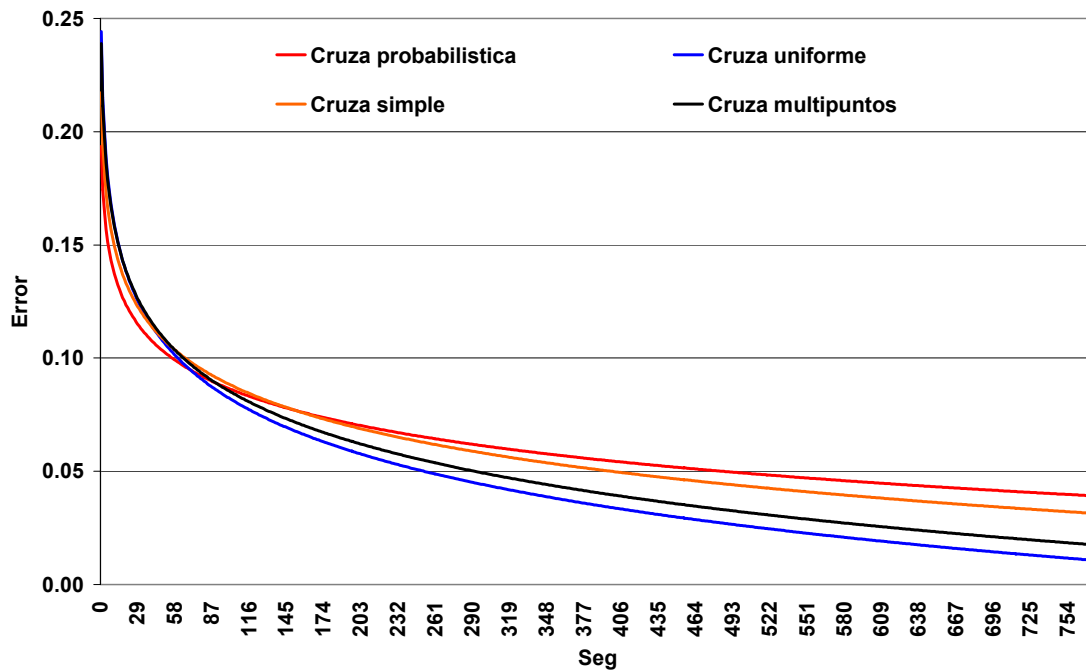
En esta sección se presentan los resultados obtenidos para cuatro operadores de cruza diferentes.

C = {Probabilística, Uniforme, Simple, Multipuntos}

##### **Resultados:**

<b>Tipo Función Cruza</b>	<b>Cant. Corridas</b>	<b>% Entrenamiento</b>	<b>Tiempo Prom. Ent.</b>	<b>Desvío Tiempo Ent.</b>
Cruza probabilística	50	98.0%	202	99
Cruza uniforme	50	100.0%	190	51
Cruza simple	50	100.0%	249	111
Cruza multipuntos	50	100.0%	216	67

**Tabla 7.12** Tiempos de entrenamiento. Función de cruza



**Gráfico 7.7** Tendencia del error de entrenamiento. Función de cruza

Tipo Función Cruza	Corrida	Tiempo Prom. Ent.
Cruza probabilística	27	124
Cruza simple	42	126
Cruza multipuntos	12	126
Cruza probabilística	40	127
	14	131
	32	131
Cruza multipuntos	42	133
Cruza probabilística	41	134
Cruza multipuntos	41	134
Cruza uniforme	12	135
	38	135
Cruza simple	10	135

**Tabla 7.13** Mejores tiempos individuales de entrenamiento.

#### Interpretación de los resultados:

- Los tiempos promedio obtenidos por todas las funciones de cruza analizadas son muy similares (Tabla 7.12).
- Se observa también que el desvío estándar del tiempo de entrenamiento es relativamente bajo (Tabla 7.12). Esta característica se debe a que los tiempos de entrenamiento permanecen similares entre las distintas ejecuciones del algoritmo.
- Los mejores tiempos individuales se reparten entre los cuatro operadores. Todos los operadores son capaces de alcanzar un nivel de entrenamiento adecuado en tiempos similares (Tabla 7.13).

- El algoritmo genético no pudo entrenar la red neuronal una única vez. Este caso se produjo utilizando la función de cruza probabilística (Tabla 7.12).
- Los tiempos promedio obtenidos por la cruza uniforme son mejores que los obtenidos por el resto de los operadores analizados (Tabla 7.13, Gráfico 7.7).
- Los resultados muestran que el método de entrenamiento es poco dependiente del operador de cruza utilizado.

### 7.2.2.3. Función de mutación

#### Descripción:

El objetivo del operador de mutación es alterar características de algunos individuos para garantizar la diversidad de la población.

#### Variables:

En esta sección se analizan los experimentos realizados para el estudio de tres operadores de mutación. Para cada uno de éstos se estudiaron también distintos rangos de variación de la probabilidad de mutación, y dos variantes para llevar a cabo la mutación:

$M = \{\text{Uniforme, Lineal Creciente, Lineal Decreciente}\}$

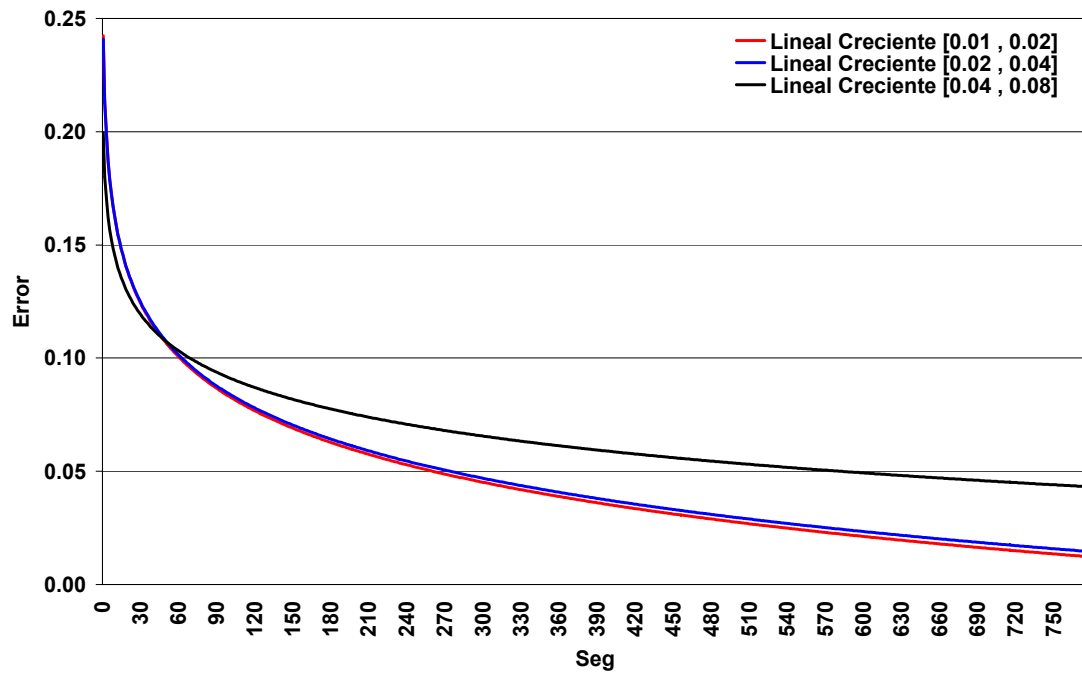
$\text{Tipo de Mutación} = \{\text{Numérica, Binaria}\}$

#### Resultados:

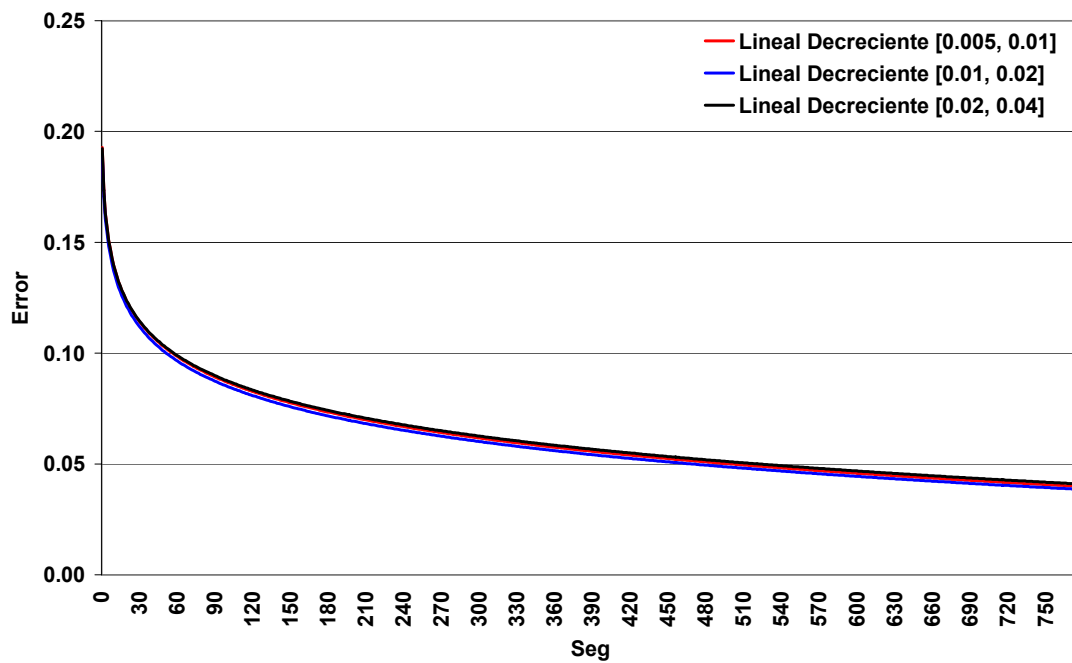
Función Mutación	Cant. Corridas	% Entrenamiento	Tiempo Prom. Ent.	Desvío Tiempo Ent.
Mutación lineal creciente [0.01,0.02]	50	100.0%	195	48
Mutación lineal creciente [0.02,0.04]	50	100.0%	191	58
Mutación lineal creciente [0.04,0.08]	50	94.0%	326	185
Mutación lineal decreciente [0.005,0.01]	50	80.0%	327	229
Mutación lineal decreciente [0.01,0.02]	50	86.0%	274	207
Mutación lineal decreciente [0.02,0.04]	50	86.0%	284	205
Mutación uniforme [0.01,0.01]	50	100.0%	214	54
Mutación uniforme [0.02,0.02]	50	100.0%	203	56
Mutación uniforme [0.04,0.04]	50	98.0%	267	167

**Tabla 7.14** Tiempos de entrenamiento. Función de mutación numérica

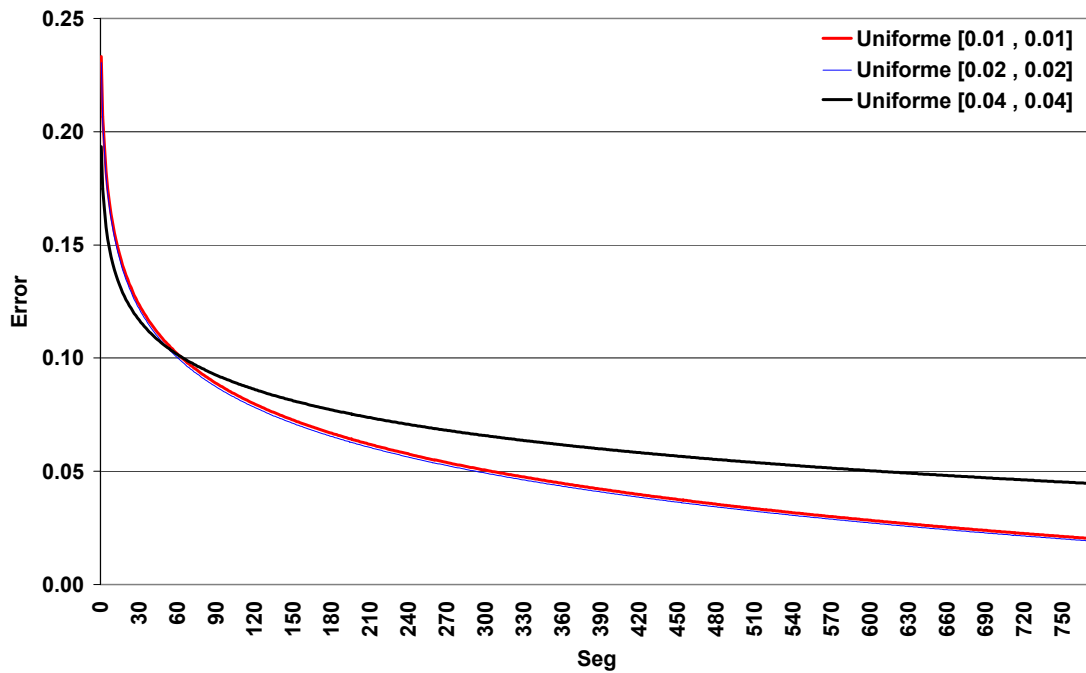




**Gráfico 7.8** Tendencia del error de entrenamiento. Función de mutación numérica lineal creciente



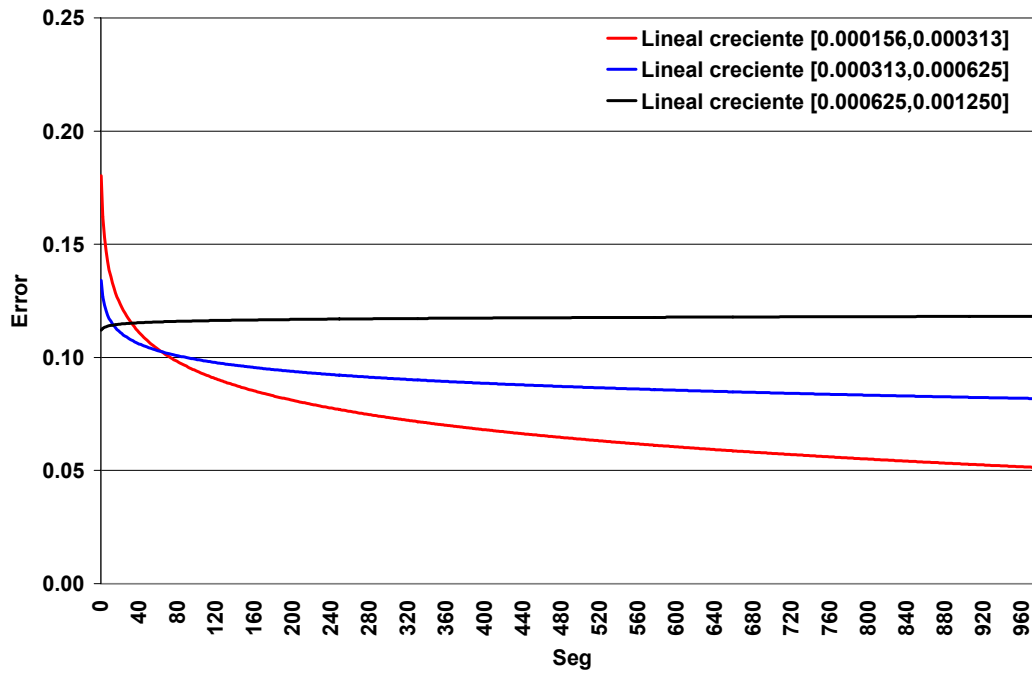
**Gráfico 7.9** Tendencia del error de entrenamiento. Función de mutación numérica lineal decreciente



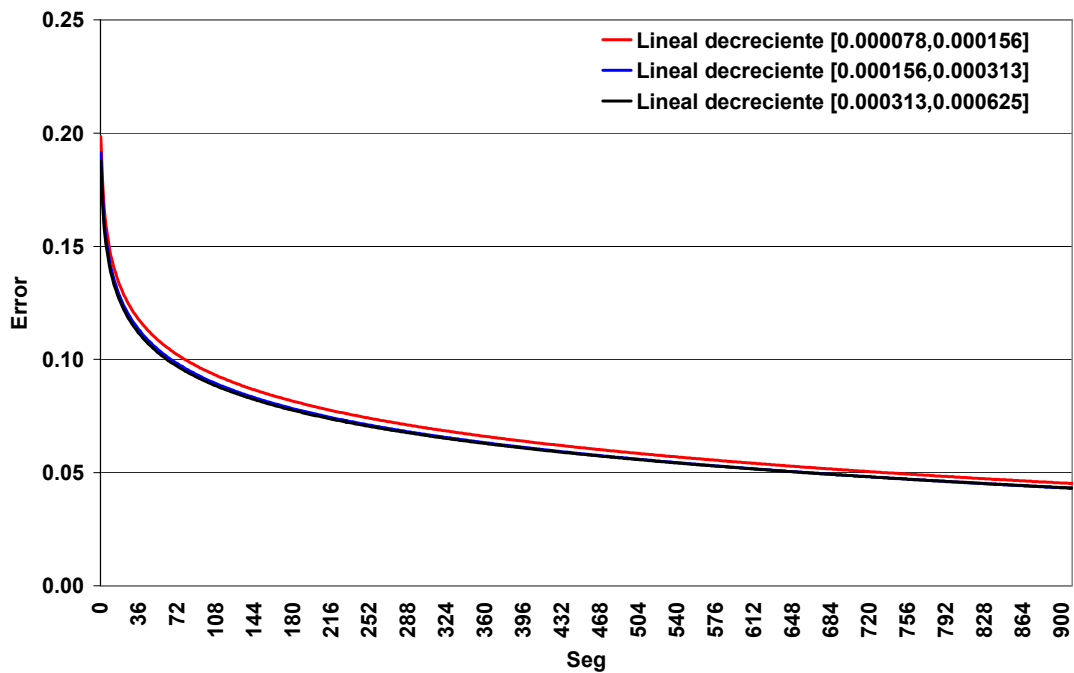
**Gráfico 7.10** Tendencia del error de entrenamiento. Función de mutación numérica uniforme

Función Mutación	Cant. Corridas	% Entrenamiento	Tiempo Prom. Ent.	Desvío Tiempo Ent.
Mutación lineal creciente [0.000156,0.000313]	50	40.0%	715	265
Mutación lineal creciente [0.000313,0.000625]	50	0.0%	914	10
Mutación lineal creciente [0.000625,0.001250]	50	0.0%	923	2
Mutación lineal decreciente [0.000078,0.000156]	50	10.0%	837	186
Mutación lineal decreciente [0.000156,0.000313]	50	40.0%	644	315
Mutación lineal decreciente [0.000313,0.000625]	50	58.0%	526	326
Mutación uniforme [0.000156,0.000156]	50	90.0%	485	214
Mutación uniforme [0.000313,0.000313]	50	88.0%	502	221
Mutación uniforme [0.000625,0.000625]	50	90.0%	543	176

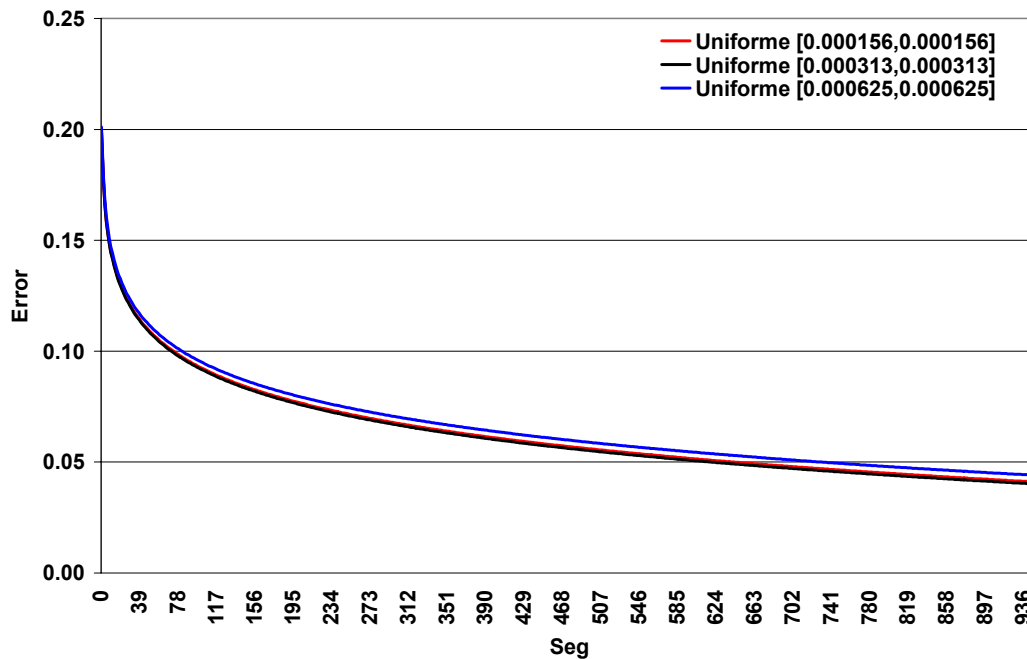
**Tabla 7.15** Tiempos de entrenamiento. Función de mutación binaria



**Gráfico 7.11** Tendencia del error de entrenamiento. Función de mutación binaria lineal creciente



**Gráfico 7.12** Tendencia del error de entrenamiento. Función de mutación binaria lineal decreciente



**Gráfico 7.13** Tendencia del error de entrenamiento. Función de mutación binaria uniforme

#### Interpretación de los resultados:

- La mutación numérica es ampliamente superior a la mutación binaria. Esto se observa claramente en los porcentajes de entrenamiento alcanzados por cada una de ellas (Tabla 7.14, Tabla 7.15).
- El uso de mutación binaria demostró no ser una buena opción. Ninguna de las implementaciones logró superar el 90 % de entrenamientos exitosos (Tabla 7.15).
- Para mutaciones binarias, el uso de tasas de mutación uniformes demuestra una amplia superioridad sobre los otros dos tipos de funciones de mutación estudiadas, cuyos resultados no superaron en ningún caso el 60 % de entrenamientos exitosos (Tabla 7.15).
- Dentro de las funciones de mutación numérica estudiadas se observa que disminuir la tasa de mutación a medida que avanzan las generaciones no es una buena opción. Los mejores porcentajes de entrenamiento se obtuvieron manteniendo una tasa de mutación constante a lo largo de las diferentes generaciones. A su vez, el uso de tasas de mutación crecientes produjo los mejores tiempos promedio de entrenamiento, con un porcentaje de convergencia levemente inferior a la mutación uniforme (Tabla 7.14).
- La función de mutación numérica lineal creciente, con probabilidades variando entre un 2% y un 4% mostró los mejores tiempos promediados sobre el total de ejecuciones. Sin embargo, otras implementaciones del operador de mutación muestran resultados comparables con los obtenidos por esta implementación (Tabla 7.14).
- El operador de mutación presenta mayor variación de resultados entre las distintas implementaciones que el resto de los operadores estudiados.

**7.2.2.4. Valores de inicialización****Descripción:**

Los experimentos presentados en esta sección tienen como objetivo encontrar cual es el rango de valores de inicialización que disminuye los tiempos de entrenamiento. Se estudian a su vez dos operadores de inicialización diferentes.

**Variables:**

Los operadores rangos estudiados fueron los siguientes:

*Tipo Inicialización* = {Random, Random Selectiva}

*Rangos* = {[ -0.01, 0.01]; [-0.1, 0.1]; [-1, 1]; [-5, 5]}

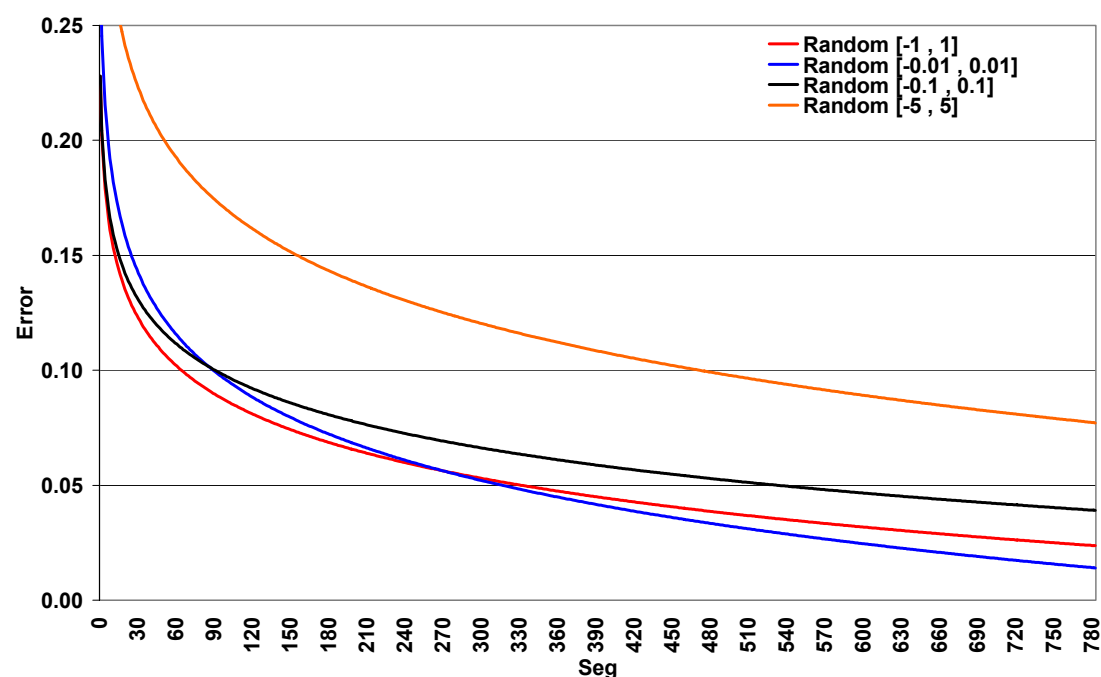
**Resultados:**

Tipo Función Inicialización	Min Ini.	Max Ini.	Cant. Corridas	% Entrenamiento	Tiempo Prom. Ent.	Desvío Tiempo Ent.
Inicialización Random	-0.01	0.01	50	100.0%	249	73
	-0.1	0.1	50	100.0%	239	86
	-1	1	50	100.0%	209	74
	-5	5	50	4.0%	770	26

**Tabla 7.16** Tiempos de entrenamiento. Inicialización random

Tipo Función Inicialización	Min Ini.	Max Ini.	Corrida	Tiempo Prom. Ent.
Inicialización Random	-1	1	32	121
			40	137
			30	138
			36	138
			44	144
			7	147
			45	150
	-0.01	0.01	25	150
	-1	1	31	151
			39	151
	-0.1	0.1	11	151

**Tabla 7.17** Mejores tiempos individuales de entrenamiento.



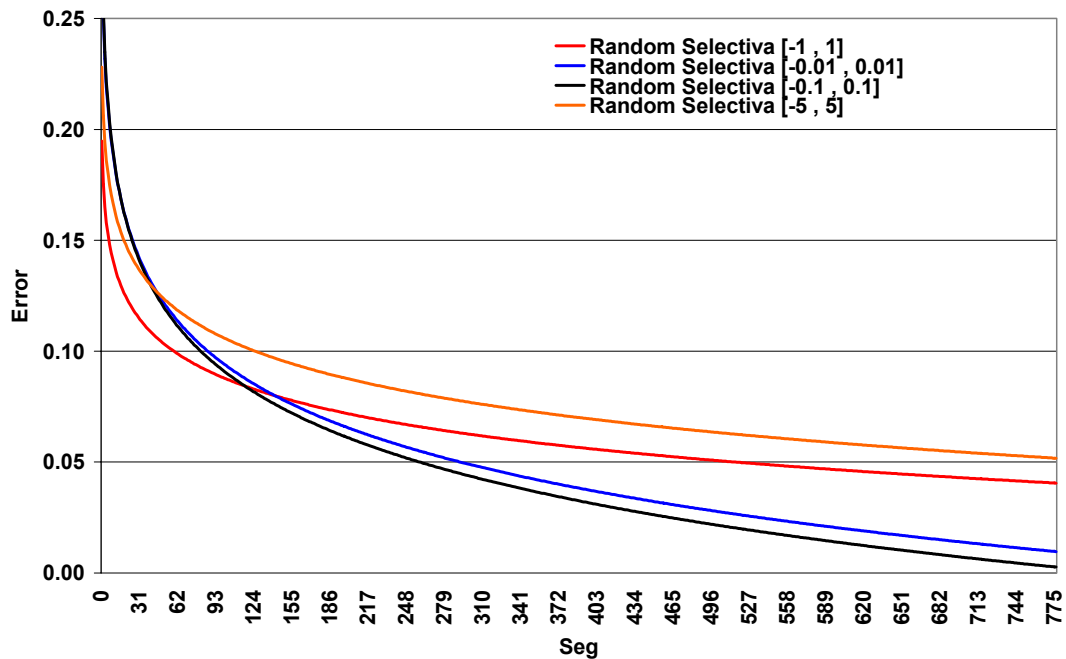
**Gráfico 7.14** Tendencia del error de entrenamiento. Inicialización random

Tipo Función Inicialización	Min Ini.	Max Ini.	Cant. Corridas	% Entrenamiento	Tiempo Prom. Ent.	Desvío Tiempo Ent.
Inicialización Random Selectiva	-0.01	0.01	50	100.0%	233	60
	-0.1	0.1	50	100.0%	222	52
	-1	1	50	100.0%	224	122
	-5	5	50	76.0%	442	235

**Tabla 7.18** Tiempos de entrenamiento. Inicialización random selectiva

Tipo Función Inicialización	Min Ini.	Max Ini.	Corrida	Tiempo Prom. Ent.
Inicialización Random Selectiva	-1	1	10	135
	-0.01	0.01	42	135
	-1	1	9	138
			42	139
	-0.1	0.1	17	139
	-1	1	19	142
	-5	5	27	145
	-1	1	24	145
	-0.1	0.1	13	145
	-1	1	46	147

**Tabla 7.19** Mejores tiempos individuales de entrenamiento.



**Gráfico 7.15** Tendencia del error de entrenamiento. Inicialización random selectiva

#### Interpretación de los resultados:

- El uso de ambos tipos de inicialización no muestra grandes diferencias. La ventaja de preseleccionar algunos valores de inicialización sólo se observa al utilizar grandes valores de inicialización (Tabla 7.16, Tabla 7.18).
- El uso de grandes valores de inicialización muestra resultados pobres. Por su parte, se obtuvieron buenos resultados al hacer uso de valores de inicialización medios y bajos (Tabla 7.16, Tabla 7.18, Gráfico 7.14, Gráfico 7.15).
- A diferencia de las redes neuronales, los algoritmos genéticos muestran mayor inmunidad a las condiciones de inicialización. Esto se observa en la paridad general en los tiempos de entrenamiento promedio obtenidos por los distintos métodos. El pequeño desvío Standard de los tiempos promedio de entrenamiento también es una buena medida de dicha inmunidad (Tabla 7.16, Tabla 7.18).
- Los mejores tiempos individuales de entrenamiento se obtuvieron utilizando valores iniciales dentro del rango  $[-1, 1]$ . Esta característica se hace más notoria al usar inicialización puramente aleatoria, aunque también se observa en menor medida al usar pesos preseleccionados (Tabla 7.17, Tabla 7.19).
- Lo dicho anteriormente impacta directamente sobre los resultados generales, donde el uso de valores en el rango  $[-1, 1]$  mostró un leve superioridad sobre el resto (Tabla 7.16, Tabla 7.18).

### 7.2.3. Comparación de ambos métodos

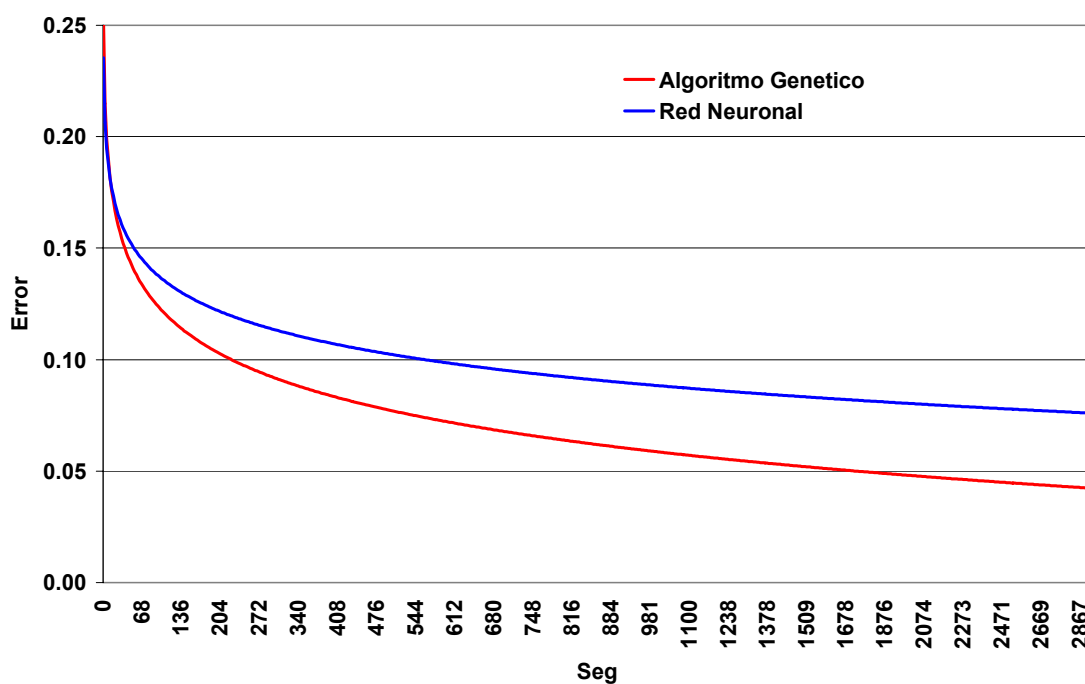
#### Descripción:

Los resultados presentados en esta sección son los más importantes del capítulo ya que permiten establecer una medida de comparación entre ambos métodos de entrenamiento. Para este conjunto de experimentos se utilizaron las configuraciones óptimas obtenidas de los experimentos anteriores. Se utilizó también un segundo set de datos, más rico en ejemplos y que permite tener una mejor medida de la capacidad de entrenamiento y generalización del método.

#### Resultados:

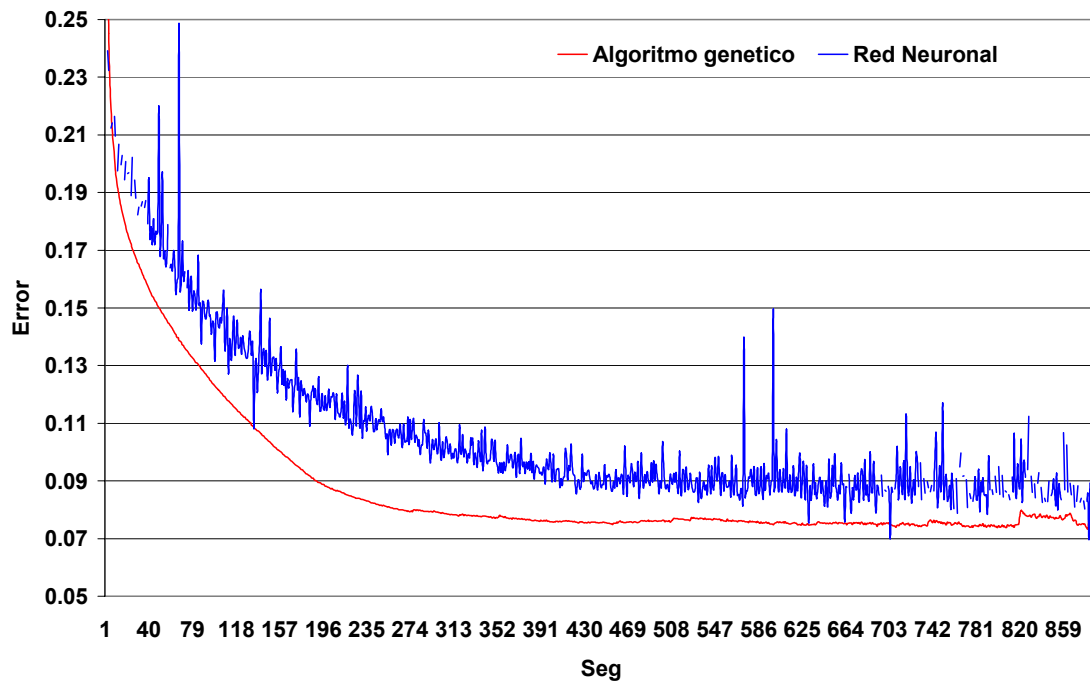
Tipo Algoritmo Entrenamiento	Cant. Corridas	% Entrenamiento	Tiempo Prom. Ent.	Desvío Tiempo Ent.
Algoritmo genético	50	100.0%	418	161
Red Neuronal	50	98.0%	687	415

**Tabla 7.20** Comparación de tiempos de entrenamiento



**Gráfico 7.16** Comparación de la tendencia del error de entrenamiento





**Gráfico 7.17** Comparación del error de entrenamiento

Tipo Algoritmo Entrenamiento	Corrida	Tiempo Prom. Ent.
Algoritmo genético	30	207
	42	226
	38	248
	25	260
	4	271
	49	274
	18	275
	34	275
	39	275
	36	279
	8	287
	12	291
	20	294
	7	301
	37	313
	9	316
	21	318
	24	318
	17	327
	46	338
Red Neuronal	33	338

**Tabla 7.21** Mejores tiempos individuales de entrenamiento.

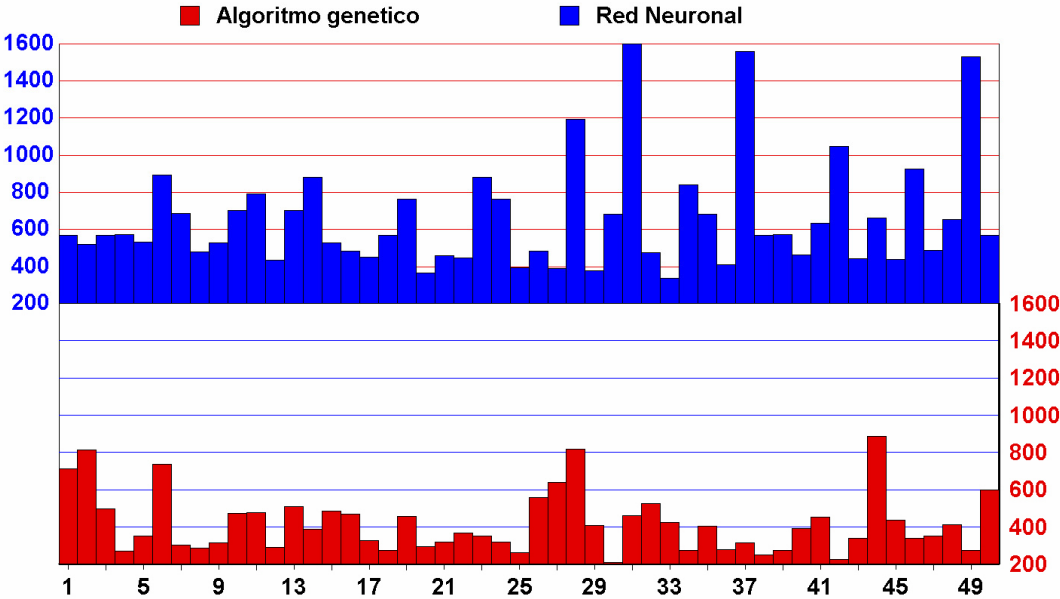


Gráfico 7.18 Tiempos individuales de entrenamiento

Tipo Algoritmo Entrenamiento	Aciertos Prom. Ent.	Desvío Aciertos Ent.	Aciertos Prom. Eval.	Desvío Aciertos Eval.	Aciertos Prom. Total	Desvío Aciertos Total
Algoritmo genético	94.09%	0.92%	82.70%	2.40%	91.81%	1.04%
Red Neuronal	94.32%	0.83%	83.29%	2.03%	92.11%	0.88%

Tabla 7.22 Comparación de la cantidad de aciertos

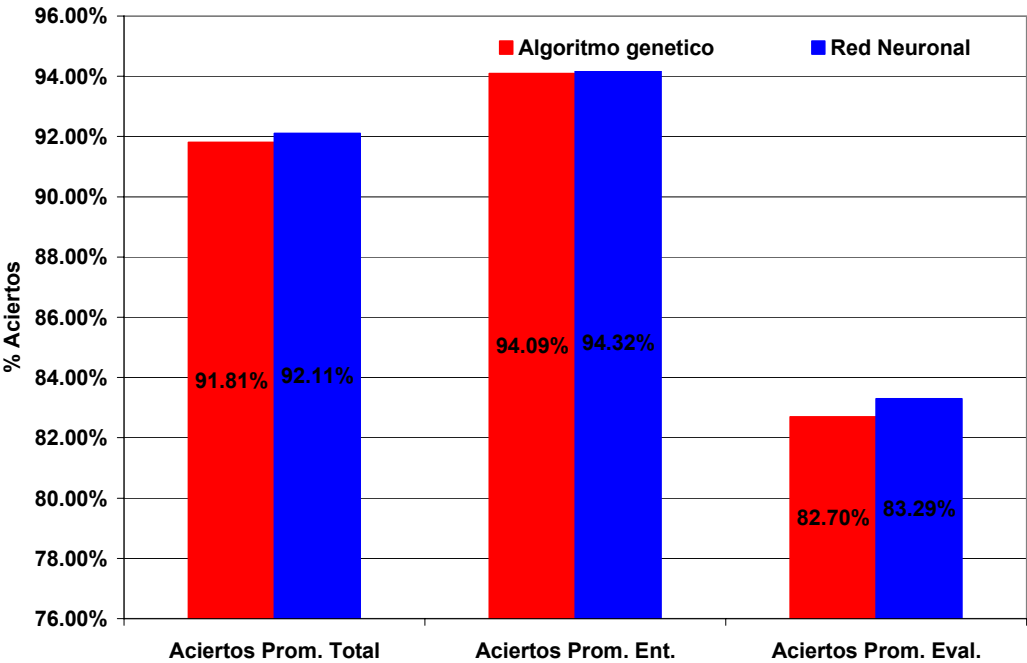


Gráfico 7.19 Comparación de la cantidad de aciertos

Tipo Algoritmo Entrenamiento	Corrida	Aciertos Prom. Ent.
Algoritmo genético	28	95.65%
Red Neuronal	49	95.65%
Algoritmo genético	2	95.47%
	6	95.47%
	19	95.47%
Red Neuronal	7	95.47%
	31	95.47%
Algoritmo genético	1	95.29%
	27	95.29%
	47	95.29%
Red Neuronal	5	95.29%
	34	95.29%
	35	95.29%
	45	95.29%

**Tabla 7.23** Mejores porcentajes de aciertos sobre el set de entrenamiento.

Tipo Algoritmo Entrenamiento	Corrida	Aciertos Prom. Eval.
Algoritmo genético	25	87.68%
	3	86.96%
Red Neuronal	20	86.96%
Algoritmo genético	19	85.51%
	28	85.51%
	43	85.51%
Red Neuronal	15	85.51%
	19	85.51%
	30	85.51%
	35	85.51%
	46	85.51%

**Tabla 7.24** Mejores porcentajes de aciertos sobre el set de validación.**Interpretación de los resultados:**

- De la comparación directa de los tiempos de entrenamiento promedio se observa una marcada superioridad de los algoritmos genéticos por sobre las redes neuronales. El tiempo promedio de entrenamiento del método propuesto es un aproximadamente un 40% menor que el obtenido por el algoritmo backpropagation. Esta característica se observa claramente en el gráfico 7.18, donde se presentan los tiempos individuales de entrenamiento para cada una de las corridas.
- En el mismo gráfico se observa que los tiempos obtenidos por el algoritmo genético presentan menor variación que los obtenidos por la red neuronal. Estos valores se pueden observar cuantitativamente en la tabla 7.20.

- El gráfico 7.16 muestra la tendencia del error promedio en el tiempo. Se observa que en el método propuesto tiene tendencia a acercarse rápidamente a la solución. En el gráfico 7.17, que muestra la evolución del error promedio de entrenamiento, se observa que durante los primeros instantes de entrenamiento se produce una aproximación rápida, que luego se vuelve en una lenta aproximación, cuando se acerca demasiado a la solución. Se observa que el algoritmo es estable, mostrando un descenso prácticamente continuo del error promedio.
- Cuando se estudian las redes neuronales se observa que en general la tendencia del error promedio presenta el mismo comportamiento que el método propuesto. Sin embargo, la curva del error se muestra siempre por encima de la de los algoritmos genéticos, llevando a una convergencia más lenta del algoritmo de entrenamiento. A su vez, en el gráfico 7.17 se observa que el método presenta menor estabilidad en el descenso del error, mostrando bruscas variaciones del error.
- Si observamos los tiempos individuales de entrenamiento, se ve que los mejores 20 tiempos individuales corresponden al algoritmo genético. Esta característica se observa gráficamente en el gráfico 7.18, donde se presentan los tiempos individuales de entrenamiento para cada una de las corridas. Los valores de los mejores 20 tiempos individuales de entrenamiento se presentan en la tabla 7.21.
- Si se estudian los porcentajes de aciertos promedios de ambos métodos se observa una leve superioridad de las redes neuronales por sobre los algoritmos genéticos. Sin embargo, como se muestra en el gráfico 7.19, esta superioridad no es muy notoria ya que representa menos del 1% de diferencia. Estos valores se observan cuantitativamente en la tabla 7.22.
- Esta paridad en cuanto a los porcentajes de aciertos promedio se observa nuevamente al estudiar el porcentaje de acierto individual para cada una de las corridas. En la tabla 7.23 se muestran los mejores porcentajes de aciertos obtenidos sobre el set de datos de entrenamiento. Como se puede observar existe una alta paridad entre ambos métodos. Análogamente, en la tabla 7.24 se muestran los mejores porcentajes de acierto sobre el set de datos de validación. De la lectura de la misma se extiende que ambos métodos poseen capacidades de generalización similares.

## Capítulo 8: Conclusiones y limitaciones

Los resultados presentados en el capítulo 7 demuestran que el método propuesto tiene la capacidad de entrenar redes neuronales con conexión hacia delante con las mismas características que los métodos tradicionales de entrenamiento. También demuestran que el método propuesto puede incluso mejorar la performance del entrenamiento en aquellos aspectos en que los métodos tradicionales suelen tener dificultades.

A continuación se abordan cada una de las cuestiones planteadas en el capítulo 6 y se responden a estos interrogantes. Las respuestas a estos interrogantes resumen los contenidos presentados a lo largo de este trabajo.

### 1. ¿Como influyen los parámetros del algoritmo backpropagation en el aprendizaje de la red?

El algoritmo backpropagation se caracteriza por su alta dependencia a los parámetros seleccionados. La manera en que se ajustan los pesos sinápticos esta determinada por la tasa de aprendizaje y la tasa de momento. El uso de tasas bajas en general lleva a aprendizajes lentos, con un alto riesgo de quedar atrapados en mínimos locales. Por su parte, si se utilizan tasas de aprendizaje grandes se aceleran los tiempos de entrenamiento, pero se corre el algoritmo oscile en torno a la solución sin lograr estabilizarse en la misma. En la primera serie de experimentos se observan las características mencionadas previamente. El estudio de diversas combinaciones de parámetros permitió determinar cual era la combinación óptima para el problema estudiado. Estos parámetros óptimos luego fueron utilizados para comparar el algoritmo backpropagation con el algoritmo genético.

### 2. ¿De qué manera influyen los valores de inicialización en el aprendizaje utilizando el algoritmo backpropagation?

Otra cuestión importante a resolver en el algoritmo backpropagation es de qué manera se genera la red inicial. La forma más habitual de hacerlo es aleatoriamente, con valores obtenidos a través de una distribución uniforme definida dentro de un rango determinado. En la segunda serie de experimentos se estudiaron distintos valores de inicialización y se observó que nuevamente el proceso de aprendizaje se ve fuertemente influenciado por los valores seleccionados. Los resultados obtenidos variaron no sólo en el tiempo necesario para alcanzar un nivel de entrenamiento adecuado sino también en el porcentaje de entrenamiento alcanzado.

### 3. ¿Cuál es el operador de selección adecuado para entrenar redes neuronales utilizando algoritmos genéticos?

La tercera serie de experimentos estuvo destinada a determinar cual es el operador de selección óptimo para entrenar redes neuronales

utilizando algoritmos genéticos. Se estudiaron cuatro variantes de este operador, las cuales se aplicaron todas en combinación con un proceso de selección elitista. Los resultados obtenidos demuestran que el método es poco dependiente del operador de selección utilizado. Esto se observa en la paridad de tiempos entre los distintos operadores, y en las bajas variaciones de tiempos entre las distintas ejecuciones del algoritmo. Esta paridad se puede atribuir en gran medida al proceso de selección elitista, que asegura que las mejores soluciones sobrevivan, lo que conduce a una mejora continua de la calidad de la solución obtenida.

**4. ¿Cuál es el operador de cruza adecuado para entrenar redes neuronales utilizando algoritmos genéticos?**

Otro punto importante que se estudió en este trabajo fue operador de cruza óptimo para entrenar redes neuronales utilizando algoritmos genéticos. Entre los resultados obtenidos se destaca la baja dependencia del algoritmo al operador de cruza seleccionado. Se observan grandes similitudes entre los tiempos de los distintos operadores estudiados, y bajas variaciones entre distintas ejecuciones del mismo operador. Como se mencionó en el capítulo 7, este operador se implementó directamente sobre el fenotipo, en contraste con los operadores de cruza clásicos utilizados en los algoritmos genéticos. Los resultados demuestran que la recombinación de neuronas permite formar nuevos individuos que preservan las características más importantes de sus padres, y se evitan de esta manera destruir estructuras complejas que han demostrado buenos niveles de adaptación. La elección de este tipo de cruza aumenta la importancia del operador de mutación, que es el encargado de introducir distorsiones que mejoren las características de los descendientes. Otro resultado importante que se observa es la baja influencia de los efectos de permutación descritos en el capítulo 6. Como se afirmó en dicho capítulo, los efectos de este problema en general están sobreestimados, y el proceso evolutivo tiene la capacidad suficiente para lidiar con dicho problema.

**5. ¿Cuál es la manera más efectiva de realizar la mutación para entrenar redes neuronales utilizando algoritmos genéticos? ¿Cuál es el operador de mutación adecuado para entrenar redes neuronales?**

Esta serie de experimentos permitió sacar varios resultados interesantes. En primer lugar, se observa de los resultados obtenidos que la mutación numérica es más adecuada que la mutación binaria. Mientras que la mutación numérica obtuvo en general altos porcentajes de entrenamiento, la mutación binaria se caracterizó por alcanzar bajos porcentajes de entrenamiento. Esto se debe a que la mutación numérica fue diseñada de acuerdo a las características propias del problema, mientras que la mutación binaria es un proceso general, cuyas características no son aplicables al problema. Otro resultado importante

que se observa en esta serie de experimentos es la manera de modificar la probabilidad de mutación. Los resultados muestran que es preferible mantener constante dicha probabilidad a lo largo de las distintas generaciones. Como se mencionó previamente, el operador de mutación asume una importancia mayor en el método propuesto que en el algoritmo genético canónico. El operador de mutación es el encargado de introducir distorsiones que mejoren las características de los descendientes. Debido a ello, las probabilidades de mutación utilizadas por este método deben ser superiores a las utilizadas por el algoritmo genético canónico. La importancia dada a este operador hace que los resultados obtenidos sean los que mostraron mayores variaciones entre sí.

**6. ¿De qué manera influyen los valores de inicialización en el aprendizaje utilizando el método propuesto? ¿Cuál es la ventaja de utilizar un esquema de inicialización selectivo?**

Una cuestión importante de resolver en el método propuesto es de qué manera se genera la población inicial. La forma más habitual de hacerlo en los algoritmos genéticos es generar la población inicial aleatoriamente, con valores obtenidos a través de una distribución uniforme definida dentro de un rango determinado. Los resultados obtenidos para esta serie de experimentos permiten afirmar que el método propuesto es menos dependiente de los valores iniciales que el algoritmo backpropagation. Esto se observa en la baja variación de los tiempos totales de entrenamiento. A pesar de ello, el uso de valores de inicialización muy grandes afecta al método propuesto, de la misma manera que lo hace con el algoritmo backpropagation. En ambos casos, el uso de valores de inicialización muy altos lleva a bajos porcentajes de entrenamiento. Sin embargo, este efecto puede ser reducido en gran medida, aunque no eliminado totalmente, utilizando un esquema de inicialización selectivo. Mediante la adición de valores preseleccionados en la población inicial se logra que el porcentaje de entrenamiento aumente en gran medida, para altos valores de inicialización.

**7. ¿Cuál es la ventaja del método propuesto en cuanto a los tiempos de entrenamiento?**

En la última serie de experimentos se realizaron comparaciones directas entre ambos métodos. Los resultados obtenidos demuestran una amplia superioridad en cuanto a tiempos de entrenamiento del método propuesto por sobre el algoritmo backpropagation. El tiempo de entrenamiento promedio del método propuesto es aproximadamente un 40% inferior que el obtenido por el algoritmo backpropagation. Otro resultado importante que se observa de los experimentos realizados es que la variación de los tiempos de entrenamiento es menor en el método propuesto que en el algoritmo backpropagation. Esta característica es de gran importancia ya que permite realizar una estimación inicial del tiempo necesario de entrenamiento de una red neuronal.

**8. ¿Cuál es la capacidad de generalización de las redes entrenadas por el método propuesto?**

El objetivo de un método de entrenamiento no es que la red neuronal aprenda todos los patrones de entrenamiento presentados, sino que extraiga de ellos las características principales para poder actuar correctamente sobre patrones nunca antes vistos. De la comparación directa entre ambos métodos se observa que el porcentaje de aciertos sobre el set de evaluación es levemente inferior para el método propuesto que para el algoritmo backpropagation. Esta característica también se observa sobre el set de entrenamiento. Sin embargo, las diferencias entre ambos métodos son inferiores al 1% por lo que ambos métodos pueden considerarse con similares capacidades de generalización

**9. ¿Cómo responde el método al fenómeno de saturación y a los mínimos locales?**

Como se describió en el capítulo 5 estos son dos problemas que afectan en gran medida la performance del algoritmo backpropagation. Sin embargo, estos problemas no tienen efecto en el método propuesto.

El fenómeno de saturación se produce cuando una neurona alcanza un máximo o un mínimo. En este caso, la derivada de la función de activación se hace nula, y los pesos de la red permanecen invariables. Como el método propuesto no hace uso de la derivada de la función de activación, el efecto de este fenómeno es completamente eliminado.

El algoritmo backpropagation suele también quedar atrapado en mínimos locales. Esto se debe a que el algoritmo hace uso de la información del gradiente de la función de error, y actualiza los pesos en la dirección de máximo decrecimiento. Nuevamente, este efecto no se produce en el método propuesto ya que el mismo no hace uso de las derivadas de la función de activación de la neurona.

Un efecto similar al de los mínimos locales que suele suceder en los algoritmos genéticos es el fenómeno de convergencia prematura. Esto sucede cuando un individuo es muy superior al resto de la población. En este caso la población tiende a hacerse homogénea y el algoritmo genético se estabiliza en una solución que puede no ser lo suficientemente buena. Para solucionar este problema se hace uso de operadores que garanticen la heterogeneidad de la población. Las características del método propuesto garantizan la heterogeneidad de la población, reduciendo al mínimo la probabilidad de que se produzca este fenómeno.

**10. ¿Cómo se adapta el método a diferentes topologías?**

Las características del método propuesto lo hacen fácilmente adaptable a otras topologías y tipos de aprendizaje. El método no hace uso de información de gradientes ni de ningún otro tipo de información propia de la topología. El proceso de actualización de pesos sólo requiere información de la calidad de la respuesta obtenida por un conjunto de



pesos. Esta información es suficiente para que el método ajuste iterativamente la solución hasta que la misma sea lo suficientemente buena. De esta manera, para entrenar redes neuronales con distintas topologías sólo será necesario estudiar las funciones de evaluación y de adaptación, adaptándolas a las características propias del problema cuando sea necesario. Esta afirmación es extensible a distintos tipos de entrenamiento, por ejemplo, el aprendizaje por refuerzo.

Los resultados obtenidos en este trabajo demuestran que el método propuesto es una alternativa más que interesante para el entrenamiento de redes neuronales. Los objetivos propuestos para el nuevo método fueron cumplidos de manera satisfactoria. Así, el método propuesto es capaz de entrenar redes neuronales con las mismas características que el algoritmo backpropagation, y evitando todos los problemas que en general suele tener dicho algoritmo. Sin embargo, existen algunos caminos que no han sido estudiados en este trabajo y que podrían llegar a mejorar la calidad del método.

Una de las características más importantes de las redes neuronales es su capacidad de generalización. En este trabajo se demostró que el método propuesto es capaz de alcanzar niveles de generalización similares a los del algoritmo backpropagation. Sin embargo esta capacidad de generalización podría ser incrementada si la función de evaluación hiciera uso de esta información. Mediante la adición de un término en dicha función que haga referencia a la capacidad de generalización se podría llegar a mejorar esta característica.

Otro punto que debería ser estudiado es la repetición de todos estos experimentos para distintos tipos de problemas y topologías. Si bien en este trabajo se estudiaron las características del método utilizando un problema real de difícil resolución, el uso de otros problemas permitirá extender y generalizar los resultados obtenidos en este trabajo. El uso de distintas topologías permitirá también comparar el método aquí propuesto con otros métodos de entrenamiento.

La utilización de un sistema híbrido que entrene redes neuronales utilizando los métodos tradicionales en conjunto con el método propuesto quedó fuera del alcance de este trabajo. Sin embargo, la utilización conjunta de ambos métodos es un camino interesante a seguir. Esta solución podría darse incluyendo un nuevo operador genético de aprendizaje, o utilizando ambos métodos, uno a continuación del otro. En este último caso se podría utilizar el método aquí propuesto para alcanzar una solución próxima a la deseada y finalizar el entrenamiento con algún método tradicional de entrenamiento.

Por último, sería interesante también estudiar las mejoras que introduce paralelizar el algoritmo genético. Los algoritmos genéticos son fácilmente paralelizables, pudiéndose de esta manera reducir aun más los tiempos de entrenamiento. Esta paralelización se puede realizar en dos sentidos. En primer lugar, haciendo uso de una única población se pueden llegar a aplicar conjuntamente los operadores genéticos en diferentes procesadores. De esta manera se podrían llegar a mutar, evaluar, etc., varios individuos simultáneamente. Otra forma de paralelizar el método propuesto consiste en hacer uso de varias poblaciones independientes. Esta técnica, llamada

algoritmos genéticos paralelos, es un área de investigación muy interesante que propone distintos modelos de paralelización.

## Referencias

- Adamidis, P. (1994) *Review of Parallel Genetic Algorithms Bibliography*. Version 1. Internal Technical Report. Automation and Robotics Lab., Dept. of Electrical and Computer Eng., Aristotle University of Thessaloniki, Greece.
- Bäch, T., Hoffmeister F. y Schwefel H. (1991) *A Survey of evolution strategies*. En *Proceedings of the Fourth International Conference on Genetic Algorithms* R. K. Belew and L. B. Booker (eds), pp. 2-9.
- Bäck, T. (1992) *Self-Adaptation in Genetic Algorithms*. En *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*. F. J. Varela and P. Bourgine (eds.). pp. 263-271. MIT Press
- Beasley, D., Bull, D. R., Martin, R. R. (1993) *An Overview of Genetic Algorithms: Part 1, Fundamentals*. En *University Computing*, 15(2) 58-69.
- Biondi, J., Michel, O. (1995). *From the chromosome to the neural network*. En *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*. D.W. Pearson, N.C. Steele, and R.F. Albrecht (eds)
- Bishop, C., Jordan, M. (1996). *Neural Networks*. En *CRC Handbook of Computer Science*, CRC Press. Tucker (ed).
- Blake C. L. y Merz C. J. (1998). *UCI Repository of machine learning databases*. <http://www.ics.uci.edu/~mllearn/MLRepository.html>. Irvine, CA: University of California, Department of Information and Computer Science.
- Booch G., Jacobson I., Rumbaugh J. (2000). *El Lenguaje Unificado de Modelado*. Addison-Wesley. Madrid. 2000.
- Branke, J. (1995) *Evolutionary Algorithms for Neural Network Design and Training*. En *Proceedings of the First Nordic Workshop on Genetic Algorithms and its Applications*, J. T. Alander (ed). 95(1) 145-163.
- Cantú-Paz, E. (1997). *A survey of Parallel Genetic Algorithms*. Technical Report Illinois Genetic Algorithms Laboratory. University of Illinois at Urbana-Champaign.
- Cauwenberghs, G. (1993). *A fast stochastic Error-Descent Algorithm for supervised learning and optimization*. En *Advances in Neural Information Processing Systems*, S. J. Hanson, J. D. Cowan, C. L. Giles (eds). pp. 244-251. Morgan Kaufmann, San Mateo, CA

- Chinrungrueng, C. (1988). *Analysis of simple neural networks*. Master's Report. Computer Science Department, University of California at Berkeley.
- Chinrungrueng, C. (1993). *Evaluation of Heterogeneous Architectures of Artificial Neural Networks*. Technical Report. Computer Science Department, University of California at Berkeley.
- Coello Coello, C. (2002). *Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A survey of the state of the Art*. En *Computer Methods in Applied Mechanics and Engineering*. 191(11-12):1245-1287
- Deb, K. (2004). *Genetic Algorithms for optimization*. En *Statistical Computing: Existing Methods and Recent Developments*. D. Kundu y A. Basu (eds.). pp. 85-123. New Delhi, India: Narosa Publishing House
- De Falco, I., Della Cioppa, A., Natale, P. Tarantino, E. (1997). *Artificial Neural Networks Optimization by means of Evolutionary Algorithms*. En *Soft Computing in Engineering Design and Manufacturing*. P. Chawdhry, R. Pant, R. Roy (eds). Springer Verlag. ISBN: 3540762140
- Del Brío, B. M., Sanz Molina, A. (2002) *Redes neuronales y Sistemas difusos*. Alfaomega Grupo Editor.
- Forrest, S. (1996). *Genetic Algorithms*. En *ACM Computer Survey*. 28(1). pp. 77-80
- Fritsch, J. (1996). *Modular Neural Networks for Speech Recognition*. Master's Thesis. Carnegie Mellon University.
- Funahashi, K.I. (1989). *On the approximate realization of continuous mappings by neural networks*. En *Neural Networks*, 2. pp. 183-192.
- García Martínez, R., Pasquini, D. y Servente, M. (2003) *Sistemas Inteligentes*. Nueva Librería.
- García Martínez, R. (1997). *Sistemas Autónomos. Aprendizaje Automático*. Nueva Librería.
- Gurney, K. (1997). *Neural Nets*. CRC Press. ISBN: 1857285034
- Hilera J. y Martínez V. (1995) *Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones*. RA-MA, Madrid.
- Holland J. H. (1975) *Adaptation in Natural and Artificial Systems*. University of Michigan Press (Ann Arbor).
- Hornik, K., Stichcombe, M. White, H. (1989) *Multilayer feedforward networks are universal approximators*. En *Neural Network*, 2(5): 359 - 366

- Ilachinsky, A. (1997). *Irreducible Semi-Autonomous Adaptive Combat (ISAAC): An Artificial-Life Approach to Land Combat*. Research Memorandum CRM 97-61.10. Center for Naval Analyses
- Isasi Viñuela, P., Galván León, I.M. (2004). *Redes Neuronales Artificiales. Un enfoque Práctico*. Prentice Hall
- Joglekar, A., Tungare, M. (2001). *Genetic Algorithms and their use in the design of Evolvable Hardware*. Fr. Conceicao Rodrigues College of Engineering.
- Jules, A., Wattenberg, M. (1994). *Stochastic Hillclimbing as a Baseline Method of Evaluating Genetic Algorithms*. Technical Report. Department of Computer Science, University of California at Berkeley
- Kim, H. (1992). *Mathematical Findings on Artificial Neural Nets and Their Physical Meaning*. Technical Report. Computer Science Department, University of California at Berkeley
- Koehn, P. (1994). *Combining Genetic Algorithms and Neural Networks: The encoding problem*. Master's thesis, University of Erlangen and The University of Tennessee, Knoxville.
- Koza, J.R. (1997). Genetic Programming. En *Encyclopedia of Science and Technology*, A. Kent y J. G. Williams (eds). 39(24) pp. 29-43
- Kolen, J., Pollack, J. (1991). *Back Propagation is Sensitive to Initial Conditions*. En *Advances in Neural Information Processing Systems*. R. P. Lippmann, J. E. Moody, D. S. Touretzky (eds). Vol. 3, pp. 860-867. Morgan Kaufmann Publishers, Inc.
- Korning, P. (1995). *Training of Neural Networks by means of Genetic Algorithms working on very long Chromosomes*. En *International Journal of Neural Systems*. Vol. 6(3), pp. 299-316.
- Kröse, B., Van der Smagt, P. (1993). *An Introduction to Neural Networks*. University of Amsterdam.
- Lehr, M., Widrow, B. (1990) *30 years of adaptive neural networks: Perceptron, Madaline and Backpropagation*. En *Proceedings of the IEEE*, 78(9), pp 1415-1442.
- Liu, A., Liu, Z., Niu, Z., Wang, C. (2004). *Evolving Neural Network using Real coded Genetic Algorithm (GA) for Multispectral Image Classification*. En *Future Generation Computer Systems The International Journal of Grid Computing: Theory, Methods and Applications*. Vol. 20(7), pp.1119-1129
- Liu, Y., Yao, X. (1996). *Evolving Artificial Neural Networks Through Evolutionary Programming*. En *Evolutionary Programming V: Proc. of the Fifth*

- Annual Conference on Evolutionary Programming*, pp.257-266. MIT Press.
- Liu Y. y Yao X. (1997). *A new evolutionary system for evolving artificial neural networks*. En *IEEE Transactions on Neural Networks*. Vol. 8(3), pp. 694-713
- Montana D.J. (1995). *Neural network weight selection using genetic algorithms*. En *Intelligent Hybrid Systems*, S. Goonatilake, S. Khebbal (eds), pp.85-104, Wiley, New York.
- Morrison, J. (1998). *Co-Evolution and Genetic Algorithms*. Master's thesis, School of Computer Science, Carleton University, Ottawa, Canada.
- Nascimento, C. (1994). *Artificial Neural Networks in Control and Optimization*. PhD Thesis. University of Manchester, Institute of Science and Technology (UMIST), Control Systems Centre, Manchester, United Kingdom.
- Ömer, B (1995) *Genetic Algorithms for Neural Network Training on Transputers*. University of Newcastle.
- Porto, V.W. (1998). *Evolutionary computation approaches to solving problems in neural computation*. En *The Handbook of Evolutionary Computation*, Bäck, T., Fogel, D., Michalewicz, Z. (eds.). ISBN: 0750303921.
- Potter, M. (1992). *A Genetic Cascade-Correlation Learning Algorithm*. En *Proceedings of {COGANN}-92 International Workshop on Combinations of Genetic Algorithms and Neural Networks*. Schaffer, J., Whitley D. (eds), p.p. 123-133. IEEE Computer Society Press
- Russell, S., Norving, P. (2004). *Inteligencia Artificial. Un enfoque moderno*. Segunda Edición. Pearson Educación.
- Sanger, T. (1989). *Optimal Unsupervised Learning in Feedforward neural networks*. En *Neural Networks*, Vol. 2, pp. 459-473.
- Sutton, R.S. (1986). *Two problems with Backpropagation and other steepest-descent learning procedures for networks*. En *Proc. of 8<sup>th</sup> Annual Conf. Of the Cognitive Science Society*, pp 823-831. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Walker, T. (1995). *Generating neural networks with genetic algorithms using a marker based encoding*. MSc Thesis. University of Edinburgh.
- Whitley, D. (1994). *A Genetic Algorithm Tutorial*. En *Statistics and Computing*. Vol 4, pp. 65-85.

- Whitley, D. (1995). *Genetic Algorithms and Neural Networks*. En *Genetic Algorithms in Engineering and Computer Science*. Periauxm J., Winter G. (eds). pp. 203-216. John Wiley & Sons Ltd.
- Whitley, D. (2001). An Overview of Evolutionary Algorithms: *Practical Issues and Common Pitfalls*. En *Journal of Information and Software Technology*. Vol 43(14), pp. 817-831.
- Whitley, D. (2002) *Genetic Algorithms and Evolutionary Computing*. En *Van Nostrand's Scientific Encyclopedia*.
- Wilson, E. (1994). *Backpropagation Learning for System with Discrete-Valued Functions*. En *Proceedings of the World Congress on Neural Networks*. Vol 3, pp. 332-339.
- Yao, X. (1993a). Evolutionary Artificial Neural Networks. En *International Journal of Neural Systems*. Vol (4), pp. 203-222.
- Yao, X. (1993b). *A Review of Evolutionary Artificial Neural Networks*. En *International Journal of Intelligent Systems*. Vol 8(4), pp. 539-567.
- Yao, X. (1995). *Evolutionary Artificial Neural Networks*. En *Encyclopedia of Computer Science and Technology*, Kent, A., J. G. Williams(eds). Vol 33, pp. 137-170, Marcel Dekker Inc.
- Yao, X. (1996). *An Overview of Evolutionary Computation*. En *Chinese Journal of Advanced Software Research*. Vol 3(1), pp. 12-29, Allerton Press Inc.
- Yao X (1999) *Evolving Artificial Neural Networks*. En *Proceedings of the IEEE*. Vol 87(9), pp.1423-1447.
- Zilouchian, A. (2001). *Fundamentals of Neural network*. En *Intelligent control Systems Using Soft Computing Methodologies*. A. Zilouchian , M. Jamshidi (eds.). CRC Press.





## Apéndice A: Datos Experimentales

En este apéndice se muestran los resultados obtenidos durante los experimentos realizados. Los experimentos se realizaron sobre un equipo Pentium IV 2.8 GHz, 512 MB de memoria, corriendo sobre un SO Windows 2000.

### A.1. Tasas de aprendizaje y momentos óptimos

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.01	0	1	SI	417	78.10%	96.43%	90.00%
0.01	0	2	SI	828	80.17%	96.43%	90.72%
0.01	0	3	SI	526	80.17%	95.76%	90.29%
0.01	0	4	SI	437	80.99%	95.76%	90.58%
0.01	0	5	SI	499	76.03%	96.21%	89.13%
0.01	0	6	SI	527	80.17%	96.21%	90.58%
0.01	0	7	SI	314	81.41%	96.21%	91.01%
0.01	0	8	SI	715	78.10%	96.88%	90.29%
0.01	0	9	SI	811	77.69%	95.98%	89.57%
0.01	0	10	SI	1,017	79.75%	96.21%	90.43%
0.01	0	11	SI	590	77.27%	94.64%	88.55%
0.01	0	12	SI	503	81.41%	95.98%	90.87%
0.01	0	13	SI	374	77.27%	96.65%	89.86%
0.01	0	14	SI	475	76.03%	95.76%	88.84%
0.01	0	15	SI	610	78.93%	95.54%	89.71%
0.01	0	16	SI	399	78.93%	95.54%	89.71%
0.01	0	17	SI	466	77.27%	96.43%	89.71%
0.01	0	18	SI	751	79.34%	96.65%	90.58%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.01	0	19	SI	507	79.75%	96.88%	90.87%
0.01	0	20	SI	726	78.93%	95.98%	90.00%
0.01	0	21	SI	448	77.69%	95.31%	89.13%
0.01	0	22	SI	637	78.51%	96.21%	90.00%
0.01	0	23	SI	389	78.10%	95.31%	89.28%
0.01	0	24	SI	587	79.75%	95.31%	89.86%
0.01	0	25	SI	772	78.93%	96.43%	90.29%
0.01	0	26	SI	706	76.86%	95.76%	89.13%
0.01	0	27	SI	625	78.10%	96.21%	89.86%
0.01	0	28	SI	565	74.79%	95.98%	88.55%
0.01	0	29	SI	557	80.17%	95.54%	90.14%
0.01	0	30	SI	612	76.45%	96.21%	89.28%
0.01	0	31	SI	440	76.45%	96.65%	89.57%
0.01	0	32	SI	432	76.86%	96.21%	89.42%
0.01	0	33	SI	332	77.27%	96.65%	89.86%
0.01	0	34	SI	428	76.45%	95.76%	88.99%
0.01	0	35	SI	365	75.21%	96.43%	88.99%
0.01	0	36	SI	444	78.51%	95.98%	89.86%
0.01	0	37	SI	316	79.75%	96.65%	90.72%
0.01	0	38	SI	289	79.34%	95.98%	90.14%
0.01	0	39	SI	573	78.10%	96.88%	90.29%
0.01	0	40	SI	790	79.75%	95.76%	90.14%
0.01	0	41	SI	442	79.75%	95.76%	90.14%
0.01	0	42	SI	629	76.45%	95.76%	88.99%
0.01	0	43	SI	322	77.69%	95.76%	89.42%
0.01	0	44	SI	765	80.58%	96.65%	91.01%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.01	0	45	SI	417	77.27%	95.54%	89.13%
0.01	0	46	SI	392	78.10%	95.54%	89.42%
0.01	0	47	SI	484	78.10%	95.98%	89.71%
0.01	0	48	SI	482	80.17%	96.65%	90.87%
0.01	0	49	SI	692	78.51%	95.54%	89.57%
0.01	0	50	SI	255	79.34%	95.98%	90.14%
0.01	0.01	1	SI	508	76.45%	96.21%	89.28%
0.01	0.01	2	SI	589	76.86%	96.43%	89.57%
0.01	0.01	3	SI	687	76.03%	95.09%	88.41%
0.01	0.01	4	SI	272	76.03%	95.31%	88.55%
0.01	0.01	5	SI	421	77.69%	96.21%	89.71%
0.01	0.01	6	SI	628	80.58%	95.98%	90.58%
0.01	0.01	7	SI	540	76.03%	96.43%	89.28%
0.01	0.01	8	SI	283	77.27%	95.98%	89.42%
0.01	0.01	9	SI	451	80.17%	96.65%	90.87%
0.01	0.01	10	SI	321	77.69%	96.21%	89.71%
0.01	0.01	11	SI	416	74.38%	94.64%	87.54%
0.01	0.01	12	SI	898	76.03%	96.65%	89.42%
0.01	0.01	13	SI	610	78.51%	96.21%	90.00%
0.01	0.01	14	SI	880	76.86%	95.98%	89.28%
0.01	0.01	15	SI	425	78.51%	95.54%	89.57%
0.01	0.01	16	SI	476	76.86%	96.65%	89.71%
0.01	0.01	17	SI	695	78.10%	95.54%	89.42%
0.01	0.01	18	SI	587	78.10%	96.43%	90.00%
0.01	0.01	19	SI	403	76.45%	95.31%	88.70%
0.01	0.01	20	SI	292	76.86%	96.21%	89.42%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.01	0.01	21	SI	1,377	78.93%	96.43%	90.29%
0.01	0.01	22	SI	537	78.51%	96.65%	90.29%
0.01	0.01	23	SI	1,541	76.45%	97.32%	90.00%
0.01	0.01	24	SI	500	76.45%	95.76%	88.99%
0.01	0.01	25	SI	364	80.17%	96.21%	90.58%
0.01	0.01	26	SI	408	76.86%	95.76%	89.13%
0.01	0.01	27	SI	713	77.27%	95.98%	89.42%
0.01	0.01	28	SI	424	75.21%	95.98%	88.70%
0.01	0.01	29	SI	465	78.93%	96.43%	90.29%
0.01	0.01	30	SI	538	78.10%	95.98%	89.71%
0.01	0.01	31	SI	490	75.62%	96.65%	89.28%
0.01	0.01	32	SI	381	77.69%	95.98%	89.57%
0.01	0.01	33	SI	572	79.34%	95.76%	90.00%
0.01	0.01	34	SI	481	77.27%	96.43%	89.71%
0.01	0.01	35	SI	669	78.93%	96.88%	90.58%
0.01	0.01	36	SI	397	77.27%	96.88%	90.00%
0.01	0.01	37	SI	926	76.45%	96.43%	89.42%
0.01	0.01	38	SI	516	79.34%	96.43%	90.43%
0.01	0.01	39	SI	842	77.69%	96.65%	90.00%
0.01	0.01	40	SI	1,113	78.93%	96.43%	90.29%
0.01	0.01	41	SI	331	77.69%	95.98%	89.57%
0.01	0.01	42	SI	389	78.10%	95.98%	89.71%
0.01	0.01	43	SI	404	77.69%	95.09%	88.99%
0.01	0.01	44	SI	617	78.93%	96.21%	90.14%
0.01	0.01	45	SI	432	78.93%	95.09%	89.42%
0.01	0.01	46	SI	585	79.34%	96.65%	90.58%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.01	0.01	47	SI	440	78.10%	96.88%	90.29%
0.01	0.01	48	SI	913	80.17%	96.88%	91.01%
0.01	0.01	49	SI	1,238	78.10%	97.10%	90.43%
0.01	0.01	50	SI	592	78.51%	96.21%	90.00%
0.01	0.05	1	SI	542	78.10%	95.31%	89.28%
0.01	0.05	2	SI	692	78.51%	97.10%	90.58%
0.01	0.05	3	SI	488	78.10%	96.21%	89.86%
0.01	0.05	4	SI	386	79.34%	95.98%	90.14%
0.01	0.05	5	SI	581	78.51%	95.98%	89.86%
0.01	0.05	6	SI	399	77.69%	95.76%	89.42%
0.01	0.05	7	SI	474	76.03%	95.31%	88.55%
0.01	0.05	8	SI	669	79.34%	95.31%	89.71%
0.01	0.05	9	SI	906	83.06%	96.43%	91.74%
0.01	0.05	10	SI	455	78.51%	96.88%	90.43%
0.01	0.05	11	SI	659	79.75%	96.65%	90.72%
0.01	0.05	12	SI	559	77.27%	96.65%	89.86%
0.01	0.05	13	SI	412	77.27%	95.31%	88.99%
0.01	0.05	14	SI	415	76.03%	96.21%	89.13%
0.01	0.05	15	SI	1,254	82.64%	96.65%	91.74%
0.01	0.05	16	SI	619	76.86%	96.21%	89.42%
0.01	0.05	17	SI	753	76.86%	95.98%	89.28%
0.01	0.05	18	SI	497	78.10%	95.98%	89.71%
0.01	0.05	19	SI	520	80.58%	96.88%	91.16%
0.01	0.05	20	SI	409	82.23%	96.43%	91.45%
0.01	0.05	21	SI	586	76.86%	96.65%	89.71%
0.01	0.05	22	SI	1,017	78.93%	96.43%	90.29%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.01	0.05	23	SI	337	76.45%	95.98%	89.13%
0.01	0.05	24	SI	316	75.21%	95.31%	88.26%
0.01	0.05	25	SI	690	75.62%	96.21%	88.99%
0.01	0.05	26	SI	433	81.41%	95.98%	90.87%
0.01	0.05	27	SI	388	77.27%	95.98%	89.42%
0.01	0.05	28	SI	1,055	78.51%	95.98%	89.86%
0.01	0.05	29	SI	899	81.41%	96.21%	91.01%
0.01	0.05	30	SI	340	78.10%	96.21%	89.86%
0.01	0.05	31	SI	672	77.69%	95.98%	89.57%
0.01	0.05	32	SI	840	80.58%	96.88%	91.16%
0.01	0.05	33	SI	335	76.45%	95.09%	88.55%
0.01	0.05	34	SI	502	76.45%	96.88%	89.71%
0.01	0.05	35	SI	423	78.10%	96.88%	90.29%
0.01	0.05	36	SI	420	78.10%	96.21%	89.86%
0.01	0.05	37	SI	706	77.27%	96.88%	90.00%
0.01	0.05	38	SI	458	76.03%	95.76%	88.84%
0.01	0.05	39	SI	396	78.51%	95.76%	89.71%
0.01	0.05	40	SI	829	78.93%	96.65%	90.43%
0.01	0.05	41	SI	594	73.97%	95.98%	88.26%
0.01	0.05	42	SI	482	74.79%	96.43%	88.84%
0.01	0.05	43	SI	905	78.93%	95.76%	89.86%
0.01	0.05	44	SI	532	80.99%	97.10%	91.45%
0.01	0.05	45	SI	612	80.58%	96.65%	91.01%
0.01	0.05	46	SI	498	76.45%	95.54%	88.84%
0.01	0.05	47	SI	386	76.03%	94.42%	87.97%
0.01	0.05	48	SI	573	79.34%	95.31%	89.71%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.01	0.05	49	SI	643	79.34%	96.65%	90.58%
0.01	0.05	50	SI	426	78.93%	96.88%	90.58%
0.01	0.1	1	SI	820	77.27%	97.32%	90.29%
0.01	0.1	2	SI	421	76.86%	96.88%	89.86%
0.01	0.1	3	SI	427	76.86%	95.98%	89.28%
0.01	0.1	4	SI	452	77.69%	96.21%	89.71%
0.01	0.1	5	SI	681	78.51%	95.98%	89.86%
0.01	0.1	6	SI	557	78.93%	96.88%	90.58%
0.01	0.1	7	SI	561	77.69%	96.21%	89.71%
0.01	0.1	8	SI	416	76.86%	95.31%	88.84%
0.01	0.1	9	SI	480	78.93%	95.54%	89.71%
0.01	0.1	10	SI	281	76.03%	95.31%	88.55%
0.01	0.1	11	SI	911	78.93%	95.76%	89.86%
0.01	0.1	12	SI	569	77.69%	97.10%	90.29%
0.01	0.1	13	SI	622	79.75%	96.88%	90.87%
0.01	0.1	14	SI	481	78.93%	95.76%	89.86%
0.01	0.1	15	SI	412	80.58%	96.21%	90.72%
0.01	0.1	16	SI	653	80.17%	96.88%	91.01%
0.01	0.1	17	SI	392	77.69%	95.54%	89.28%
0.01	0.1	18	SI	503	76.86%	96.21%	89.42%
0.01	0.1	19	SI	825	81.41%	96.88%	91.45%
0.01	0.1	20	SI	1,113	81.41%	95.98%	90.87%
0.01	0.1	21	SI	271	79.75%	96.65%	90.72%
0.01	0.1	22	SI	441	78.93%	96.21%	90.14%
0.01	0.1	23	SI	1,035	78.10%	96.43%	90.00%
0.01	0.1	24	SI	352	78.93%	96.88%	90.58%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.01	0.1	25	SI	379	77.69%	96.21%	89.71%
0.01	0.1	26	SI	509	76.86%	96.21%	89.42%
0.01	0.1	27	SI	423	80.99%	95.98%	90.72%
0.01	0.1	28	SI	664	81.41%	96.21%	91.01%
0.01	0.1	29	SI	567	79.34%	95.98%	90.14%
0.01	0.1	30	SI	309	80.58%	96.43%	90.87%
0.01	0.1	31	SI	449	79.75%	96.43%	90.58%
0.01	0.1	32	SI	575	77.27%	95.09%	88.84%
0.01	0.1	33	SI	541	78.10%	96.65%	90.14%
0.01	0.1	34	SI	597	78.10%	96.21%	89.86%
0.01	0.1	35	SI	515	80.58%	95.54%	90.29%
0.01	0.1	36	SI	656	77.69%	97.10%	90.29%
0.01	0.1	37	SI	395	78.10%	96.65%	90.14%
0.01	0.1	38	SI	409	73.55%	95.76%	87.97%
0.01	0.1	39	SI	632	79.34%	96.43%	90.43%
0.01	0.1	40	SI	426	73.97%	96.21%	88.41%
0.01	0.1	41	SI	669	79.34%	96.65%	90.58%
0.01	0.1	42	SI	617	80.17%	96.21%	90.58%
0.01	0.1	43	SI	407	79.75%	95.76%	90.14%
0.01	0.1	44	SI	608	78.51%	95.76%	89.71%
0.01	0.1	45	SI	622	78.93%	95.98%	90.00%
0.01	0.1	46	SI	380	78.51%	96.21%	90.00%
0.01	0.1	47	SI	403	75.62%	97.10%	89.57%
0.01	0.1	48	SI	432	76.45%	94.64%	88.26%
0.01	0.1	49	SI	780	77.69%	96.88%	90.14%
0.01	0.1	50	SI	449	78.51%	95.98%	89.86%



Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.05	0	1	SI	946	78.10%	96.88%	90.29%
0.05	0	2	SI	297	78.51%	96.21%	90.00%
0.05	0	3	SI	915	79.75%	97.10%	91.01%
0.05	0	4	SI	326	79.75%	95.09%	89.71%
0.05	0	5	SI	223	78.93%	95.98%	90.00%
0.05	0	6	SI	162	77.69%	96.43%	89.86%
0.05	0	7	SI	200	78.51%	97.32%	90.72%
0.05	0	8	SI	182	76.03%	95.98%	88.99%
0.05	0	9	SI	176	79.75%	94.64%	89.42%
0.05	0	10	SI	1,037	76.86%	96.88%	89.86%
0.05	0	11	SI	472	78.51%	96.43%	90.14%
0.05	0	12	SI	327	78.51%	95.09%	89.28%
0.05	0	13	SI	304	78.10%	96.88%	90.29%
0.05	0	14	SI	138	78.10%	95.98%	89.71%
0.05	0	15	SI	208	76.45%	95.31%	88.70%
0.05	0	16	SI	264	76.03%	95.09%	88.41%
0.05	0	17	SI	244	79.34%	96.65%	90.58%
0.05	0	18	SI	279	78.51%	96.65%	90.29%
0.05	0	19	SI	204	77.27%	95.54%	89.13%
0.05	0	20	SI	357	79.75%	94.64%	89.42%
0.05	0	21	SI	327	78.51%	97.10%	90.58%
0.05	0	22	SI	255	76.03%	96.43%	89.28%
0.05	0	23	SI	218	78.93%	95.54%	89.71%
0.05	0	24	SI	321	79.75%	96.65%	90.72%
0.05	0	25	SI	254	77.27%	96.88%	90.00%
0.05	0	26	SI	254	78.51%	96.88%	90.43%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.05	0	27	SI	446	78.51%	95.76%	89.71%
0.05	0	28	SI	220	78.93%	96.43%	90.29%
0.05	0	29	SI	228	80.58%	95.31%	90.14%
0.05	0	30	SI	383	80.99%	96.21%	90.87%
0.05	0	31	SI	172	78.93%	96.65%	90.43%
0.05	0	32	SI	176	78.10%	95.76%	89.57%
0.05	0	33	SI	266	80.17%	96.88%	91.01%
0.05	0	34	SI	596	79.34%	96.43%	90.43%
0.05	0	35	SI	297	79.34%	96.21%	90.29%
0.05	0	36	SI	201	79.75%	96.43%	90.58%
0.05	0	37	SI	265	78.51%	96.21%	90.00%
0.05	0	38	SI	291	78.51%	95.98%	89.86%
0.05	0	39	SI	458	78.10%	96.21%	89.86%
0.05	0	40	SI	223	80.58%	97.10%	91.30%
0.05	0	41	SI	413	79.75%	96.43%	90.58%
0.05	0	42	SI	390	80.17%	95.76%	90.29%
0.05	0	43	SI	284	78.10%	95.98%	89.71%
0.05	0	44	SI	196	76.86%	96.21%	89.42%
0.05	0	45	SI	185	79.75%	95.98%	90.29%
0.05	0	46	SI	302	76.03%	95.98%	88.99%
0.05	0	47	SI	290	78.93%	95.09%	89.42%
0.05	0	48	SI	361	78.93%	96.43%	90.29%
0.05	0	49	SI	460	81.41%	96.65%	91.30%
0.05	0	50	SI	200	78.51%	96.43%	90.14%
0.05	0.01	1	SI	160	80.58%	97.10%	91.30%
0.05	0.01	2	SI	284	81.82%	96.21%	91.16%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.05	0.01	3	SI	280	77.69%	96.21%	89.71%
0.05	0.01	4	SI	227	78.93%	96.88%	90.58%
0.05	0.01	5	SI	238	77.27%	96.43%	89.71%
0.05	0.01	6	SI	205	78.51%	95.98%	89.86%
0.05	0.01	7	SI	145	78.51%	96.88%	90.43%
0.05	0.01	8	SI	289	77.69%	95.76%	89.42%
0.05	0.01	9	SI	337	76.45%	96.65%	89.57%
0.05	0.01	10	SI	216	76.45%	95.98%	89.13%
0.05	0.01	11	SI	170	80.17%	95.76%	90.29%
0.05	0.01	12	SI	363	80.58%	96.65%	91.01%
0.05	0.01	13	SI	310	80.17%	96.43%	90.72%
0.05	0.01	14	SI	571	78.93%	95.76%	89.86%
0.05	0.01	15	SI	407	80.58%	97.10%	91.30%
0.05	0.01	16	SI	251	77.27%	94.87%	88.70%
0.05	0.01	17	SI	361	78.10%	96.43%	90.00%
0.05	0.01	18	SI	410	81.41%	96.65%	91.30%
0.05	0.01	19	SI	217	77.27%	95.09%	88.84%
0.05	0.01	20	SI	194	78.10%	95.09%	89.13%
0.05	0.01	21	SI	382	78.10%	97.10%	90.43%
0.05	0.01	22	SI	234	79.75%	95.09%	89.71%
0.05	0.01	23	SI	221	76.86%	95.09%	88.70%
0.05	0.01	24	SI	373	79.75%	96.88%	90.87%
0.05	0.01	25	SI	153	77.27%	97.32%	90.29%
0.05	0.01	26	SI	148	76.03%	95.76%	88.84%
0.05	0.01	27	SI	546	80.99%	96.65%	91.16%
0.05	0.01	28	SI	302	78.93%	95.76%	89.86%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.05	0.01	29	SI	168	78.93%	96.88%	90.58%
0.05	0.01	30	SI	257	77.69%	96.88%	90.14%
0.05	0.01	31	SI	350	77.27%	95.98%	89.42%
0.05	0.01	32	SI	294	80.58%	96.43%	90.87%
0.05	0.01	33	SI	189	78.93%	96.65%	90.43%
0.05	0.01	34	SI	233	78.93%	95.98%	90.00%
0.05	0.01	35	SI	231	79.75%	96.65%	90.72%
0.05	0.01	36	SI	241	80.17%	96.21%	90.58%
0.05	0.01	37	SI	270	76.86%	96.21%	89.42%
0.05	0.01	38	SI	323	79.34%	97.10%	90.87%
0.05	0.01	39	SI	252	78.10%	96.21%	89.86%
0.05	0.01	40	SI	244	78.10%	96.65%	90.14%
0.05	0.01	41	SI	251	80.17%	95.76%	90.29%
0.05	0.01	42	SI	157	77.27%	95.98%	89.42%
0.05	0.01	43	SI	197	76.86%	94.42%	88.26%
0.05	0.01	44	SI	335	73.97%	95.54%	87.97%
0.05	0.01	45	SI	174	80.58%	96.43%	90.87%
0.05	0.01	46	SI	240	78.93%	97.10%	90.72%
0.05	0.01	47	SI	215	76.03%	96.43%	89.28%
0.05	0.01	48	SI	273	79.34%	96.65%	90.58%
0.05	0.01	49	SI	246	78.51%	96.21%	90.00%
0.05	0.01	50	SI	311	77.27%	95.98%	89.42%
0.05	0.05	1	SI	579	78.93%	96.43%	90.29%
0.05	0.05	2	SI	273	80.99%	96.65%	91.16%
0.05	0.05	3	SI	252	78.10%	95.31%	89.28%
0.05	0.05	4	SI	506	76.86%	96.65%	89.71%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.05	0.05	5	SI	272	78.10%	95.76%	89.57%
0.05	0.05	6	SI	197	76.86%	95.31%	88.84%
0.05	0.05	7	SI	214	78.51%	95.98%	89.86%
0.05	0.05	8	SI	214	78.51%	95.98%	89.86%
0.05	0.05	9	SI	212	78.93%	95.76%	89.86%
0.05	0.05	10	SI	313	77.69%	97.10%	90.29%
0.05	0.05	11	SI	173	78.51%	96.43%	90.14%
0.05	0.05	12	SI	435	78.93%	96.65%	90.43%
0.05	0.05	13	SI	208	78.51%	95.31%	89.42%
0.05	0.05	14	SI	233	77.27%	96.65%	89.86%
0.05	0.05	15	SI	266	80.17%	97.10%	91.16%
0.05	0.05	16	SI	232	78.10%	95.09%	89.13%
0.05	0.05	17	SI	139	80.58%	95.54%	90.29%
0.05	0.05	18	SI	397	77.69%	96.43%	89.86%
0.05	0.05	19	SI	308	79.34%	97.10%	90.87%
0.05	0.05	20	SI	193	78.51%	96.43%	90.14%
0.05	0.05	21	SI	1,824	80.99%	97.32%	91.59%
0.05	0.05	22	SI	293	78.51%	96.65%	90.29%
0.05	0.05	23	SI	161	78.51%	94.87%	89.13%
0.05	0.05	24	SI	828	78.10%	94.64%	88.84%
0.05	0.05	25	SI	232	78.10%	96.43%	90.00%
0.05	0.05	26	SI	336	80.58%	97.10%	91.30%
0.05	0.05	27	SI	334	76.03%	95.31%	88.55%
0.05	0.05	28	SI	237	75.62%	95.76%	88.70%
0.05	0.05	29	SI	272	76.86%	95.76%	89.13%
0.05	0.05	30	SI	258	78.51%	95.54%	89.57%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.05	0.05	31	SI	434	79.34%	96.88%	90.72%
0.05	0.05	32	SI	372	80.17%	97.10%	91.16%
0.05	0.05	33	SI	400	76.86%	94.64%	88.41%
0.05	0.05	34	SI	358	77.69%	95.54%	89.28%
0.05	0.05	35	SI	252	79.34%	96.43%	90.43%
0.05	0.05	36	NO	1,967	80.58%	96.65%	91.01%
0.05	0.05	37	SI	427	78.51%	97.10%	90.58%
0.05	0.05	38	SI	248	78.93%	95.09%	89.42%
0.05	0.05	39	SI	252	77.69%	94.20%	88.41%
0.05	0.05	40	SI	307	77.27%	97.10%	90.14%
0.05	0.05	41	SI	916	76.45%	96.43%	89.42%
0.05	0.05	42	SI	273	78.51%	95.98%	89.86%
0.05	0.05	43	SI	212	77.27%	95.98%	89.42%
0.05	0.05	44	SI	160	78.10%	95.54%	89.42%
0.05	0.05	45	SI	353	77.27%	95.76%	89.28%
0.05	0.05	46	SI	228	77.27%	95.09%	88.84%
0.05	0.05	47	SI	253	79.34%	95.98%	90.14%
0.05	0.05	48	SI	232	79.34%	96.88%	90.72%
0.05	0.05	49	SI	192	76.86%	96.21%	89.42%
0.05	0.05	50	SI	156	77.27%	96.65%	89.86%
0.05	0.1	1	SI	305	79.34%	95.98%	90.14%
0.05	0.1	2	SI	162	77.27%	94.42%	88.41%
0.05	0.1	3	SI	185	80.17%	96.21%	90.58%
0.05	0.1	4	SI	237	82.64%	96.21%	91.45%
0.05	0.1	5	SI	204	78.51%	96.65%	90.29%
0.05	0.1	6	SI	175	78.10%	97.10%	90.43%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.05	0.1	7	SI	537	76.03%	96.21%	89.13%
0.05	0.1	8	SI	315	79.34%	96.65%	90.58%
0.05	0.1	9	SI	169	81.41%	96.21%	91.01%
0.05	0.1	10	SI	504	77.27%	96.43%	89.71%
0.05	0.1	11	SI	341	76.45%	96.43%	89.42%
0.05	0.1	12	SI	219	79.34%	96.65%	90.58%
0.05	0.1	13	SI	171	77.27%	94.64%	88.55%
0.05	0.1	14	SI	1,438	79.34%	96.21%	90.29%
0.05	0.1	15	SI	201	76.45%	96.43%	89.42%
0.05	0.1	16	SI	216	78.10%	96.43%	90.00%
0.05	0.1	17	SI	533	79.34%	95.98%	90.14%
0.05	0.1	18	SI	493	75.62%	95.09%	88.26%
0.05	0.1	19	SI	196	80.17%	96.21%	90.58%
0.05	0.1	20	SI	342	80.99%	96.43%	91.01%
0.05	0.1	21	SI	334	77.69%	94.87%	88.84%
0.05	0.1	22	SI	196	77.27%	95.54%	89.13%
0.05	0.1	23	SI	338	78.93%	95.31%	89.57%
0.05	0.1	24	SI	216	78.51%	97.10%	90.58%
0.05	0.1	25	SI	350	79.34%	96.43%	90.43%
0.05	0.1	26	SI	255	76.03%	96.43%	89.28%
0.05	0.1	27	SI	482	81.41%	96.88%	91.45%
0.05	0.1	28	SI	148	78.93%	96.21%	90.14%
0.05	0.1	29	SI	157	81.41%	96.88%	91.45%
0.05	0.1	30	SI	220	78.51%	96.65%	90.29%
0.05	0.1	31	SI	325	74.79%	94.87%	87.83%
0.05	0.1	32	SI	299	78.10%	96.65%	90.14%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.05	0.1	33	SI	377	76.86%	96.43%	89.57%
0.05	0.1	34	SI	207	77.69%	94.87%	88.84%
0.05	0.1	35	SI	194	79.75%	94.87%	89.57%
0.05	0.1	36	SI	226	80.17%	96.43%	90.72%
0.05	0.1	37	SI	252	78.93%	96.43%	90.29%
0.05	0.1	38	SI	285	78.51%	97.10%	90.58%
0.05	0.1	39	SI	193	78.93%	94.87%	89.28%
0.05	0.1	40	SI	200	77.69%	94.87%	88.84%
0.05	0.1	41	SI	194	79.75%	95.76%	90.14%
0.05	0.1	42	SI	208	79.34%	95.98%	90.14%
0.05	0.1	43	SI	359	78.93%	97.10%	90.72%
0.05	0.1	44	SI	288	79.75%	96.65%	90.72%
0.05	0.1	45	SI	296	78.10%	94.64%	88.84%
0.05	0.1	46	SI	314	79.75%	95.76%	90.14%
0.05	0.1	47	SI	244	79.75%	95.98%	90.29%
0.05	0.1	48	SI	209	77.69%	96.88%	90.14%
0.05	0.1	49	SI	302	78.10%	96.43%	90.00%
0.05	0.1	50	SI	191	80.58%	96.43%	90.87%
0.1	0	1	SI	347	77.27%	96.65%	89.86%
0.1	0	2	SI	388	82.23%	97.32%	92.03%
0.1	0	3	SI	270	79.75%	96.21%	90.43%
0.1	0	4	SI	222	79.34%	95.98%	90.14%
0.1	0	5	SI	250	79.75%	95.54%	90.00%
0.1	0	6	SI	245	76.45%	95.09%	88.55%
0.1	0	7	SI	226	80.99%	96.65%	91.16%
0.1	0	8	SI	378	77.27%	95.76%	89.28%



Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.1	0	9	SI	723	77.27%	95.54%	89.13%
0.1	0	10	SI	228	76.86%	96.65%	89.71%
0.1	0	11	SI	376	80.58%	96.88%	91.16%
0.1	0	12	SI	212	74.38%	96.21%	88.55%
0.1	0	13	SI	395	80.17%	94.42%	89.42%
0.1	0	14	SI	279	77.69%	95.98%	89.57%
0.1	0	15	SI	338	80.58%	96.88%	91.16%
0.1	0	16	SI	337	77.69%	96.88%	90.14%
0.1	0	17	SI	269	78.10%	97.10%	90.43%
0.1	0	18	SI	499	79.75%	96.65%	90.72%
0.1	0	19	SI	422	80.58%	96.21%	90.72%
0.1	0	20	SI	335	77.27%	96.21%	89.57%
0.1	0	21	SI	270	80.58%	96.21%	90.72%
0.1	0	22	SI	759	79.34%	96.65%	90.58%
0.1	0	23	SI	232	75.21%	94.64%	87.83%
0.1	0	24	SI	239	78.10%	96.43%	90.00%
0.1	0	25	SI	398	79.34%	96.65%	90.58%
0.1	0	26	SI	345	78.10%	96.65%	90.14%
0.1	0	27	SI	338	78.93%	94.64%	89.13%
0.1	0	28	SI	330	78.51%	96.88%	90.43%
0.1	0	29	SI	480	80.58%	96.88%	91.16%
0.1	0	30	SI	387	78.51%	96.65%	90.29%
0.1	0	31	SI	690	80.58%	96.43%	90.87%
0.1	0	32	SI	376	77.69%	97.32%	90.43%
0.1	0	33	SI	274	77.27%	95.76%	89.28%
0.1	0	34	SI	462	79.34%	95.98%	90.14%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.1	0	35	SI	409	79.34%	96.21%	90.29%
0.1	0	36	SI	253	77.27%	96.43%	89.71%
0.1	0	37	SI	248	78.51%	95.98%	89.86%
0.1	0	38	SI	539	79.75%	96.43%	90.58%
0.1	0	39	SI	425	79.34%	97.10%	90.87%
0.1	0	40	SI	536	78.93%	95.98%	90.00%
0.1	0	41	SI	210	78.93%	96.21%	90.14%
0.1	0	42	SI	255	80.99%	96.43%	91.01%
0.1	0	43	SI	737	78.93%	97.10%	90.72%
0.1	0	44	SI	206	78.51%	97.10%	90.58%
0.1	0	45	SI	500	78.51%	95.76%	89.71%
0.1	0	46	SI	349	78.10%	95.09%	89.13%
0.1	0	47	SI	252	80.17%	97.10%	91.16%
0.1	0	48	SI	346	77.27%	95.76%	89.28%
0.1	0	49	SI	308	79.75%	97.10%	91.01%
0.1	0	50	SI	234	79.34%	97.32%	91.01%
0.1	0.01	1	SI	284	78.51%	95.54%	89.57%
0.1	0.01	2	SI	313	78.93%	97.10%	90.72%
0.1	0.01	3	SI	232	78.51%	95.76%	89.71%
0.1	0.01	4	SI	333	81.41%	96.88%	91.45%
0.1	0.01	5	SI	382	78.93%	96.21%	90.14%
0.1	0.01	6	SI	212	78.93%	95.09%	89.42%
0.1	0.01	7	SI	399	80.58%	97.10%	91.30%
0.1	0.01	8	SI	305	78.51%	96.65%	90.29%
0.1	0.01	9	SI	232	76.86%	95.54%	88.99%
0.1	0.01	10	SI	448	78.93%	96.43%	90.29%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.1	0.01	11	SI	222	78.10%	96.21%	89.86%
0.1	0.01	12	SI	186	76.86%	96.65%	89.71%
0.1	0.01	13	SI	204	79.75%	97.10%	91.01%
0.1	0.01	14	SI	137	78.93%	96.65%	90.43%
0.1	0.01	15	SI	317	78.93%	95.76%	89.86%
0.1	0.01	16	SI	386	80.17%	95.31%	90.00%
0.1	0.01	17	SI	548	78.51%	96.43%	90.14%
0.1	0.01	18	SI	375	78.93%	94.87%	89.28%
0.1	0.01	19	SI	273	77.69%	97.32%	90.43%
0.1	0.01	20	SI	365	76.45%	95.54%	88.84%
0.1	0.01	21	SI	408	74.38%	93.75%	86.96%
0.1	0.01	22	SI	785	80.17%	96.43%	90.72%
0.1	0.01	23	SI	386	78.93%	95.09%	89.42%
0.1	0.01	24	SI	174	76.86%	96.43%	89.57%
0.1	0.01	25	SI	181	77.27%	95.09%	88.84%
0.1	0.01	26	SI	233	80.99%	96.43%	91.01%
0.1	0.01	27	SI	329	80.99%	97.10%	91.45%
0.1	0.01	28	SI	288	80.58%	95.31%	90.14%
0.1	0.01	29	SI	259	77.69%	96.21%	89.71%
0.1	0.01	30	SI	379	77.69%	96.43%	89.86%
0.1	0.01	31	SI	326	78.51%	97.10%	90.58%
0.1	0.01	32	SI	372	78.93%	95.09%	89.42%
0.1	0.01	33	SI	218	78.10%	95.76%	89.57%
0.1	0.01	34	SI	189	78.10%	96.88%	90.29%
0.1	0.01	35	SI	333	77.27%	96.43%	89.71%
0.1	0.01	36	SI	420	79.75%	97.10%	91.01%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.1	0.01	37	SI	323	79.75%	96.65%	90.72%
0.1	0.01	38	SI	184	78.51%	95.76%	89.71%
0.1	0.01	39	SI	344	80.99%	96.65%	91.16%
0.1	0.01	40	SI	290	78.51%	95.98%	89.86%
0.1	0.01	41	SI	299	80.99%	95.54%	90.43%
0.1	0.01	42	SI	171	80.58%	95.54%	90.29%
0.1	0.01	43	SI	231	77.69%	94.87%	88.84%
0.1	0.01	44	SI	301	78.10%	95.31%	89.28%
0.1	0.01	45	SI	1,091	79.34%	96.88%	90.72%
0.1	0.01	46	SI	321	79.34%	96.21%	90.29%
0.1	0.01	47	SI	581	78.51%	95.54%	89.57%
0.1	0.01	48	SI	412	79.75%	96.43%	90.58%
0.1	0.01	49	SI	259	79.34%	96.65%	90.58%
0.1	0.01	50	SI	288	78.51%	95.98%	89.86%
0.1	0.05	1	SI	347	78.51%	96.43%	90.14%
0.1	0.05	2	SI	204	79.34%	96.21%	90.29%
0.1	0.05	3	NO	1,956	80.99%	96.88%	91.30%
0.1	0.05	4	SI	306	77.69%	96.43%	89.86%
0.1	0.05	5	SI	260	78.10%	96.21%	89.86%
0.1	0.05	6	SI	266	81.41%	96.65%	91.30%
0.1	0.05	7	SI	391	80.17%	96.21%	90.58%
0.1	0.05	8	SI	278	76.03%	95.09%	88.41%
0.1	0.05	9	SI	199	77.27%	96.21%	89.57%
0.1	0.05	10	SI	278	78.10%	97.10%	90.43%
0.1	0.05	11	SI	163	78.51%	96.21%	90.00%
0.1	0.05	12	SI	278	76.86%	96.21%	89.42%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.1	0.05	13	SI	522	79.34%	96.43%	90.43%
0.1	0.05	14	SI	223	78.51%	95.54%	89.57%
0.1	0.05	15	SI	237	79.75%	96.43%	90.58%
0.1	0.05	16	SI	591	77.27%	96.43%	89.71%
0.1	0.05	17	SI	265	78.93%	95.76%	89.86%
0.1	0.05	18	SI	324	74.38%	96.43%	88.70%
0.1	0.05	19	SI	382	76.03%	94.64%	88.12%
0.1	0.05	20	SI	191	78.10%	94.87%	88.99%
0.1	0.05	21	SI	386	80.17%	96.65%	90.87%
0.1	0.05	22	SI	202	78.10%	95.54%	89.42%
0.1	0.05	23	SI	282	78.51%	95.76%	89.71%
0.1	0.05	24	SI	1,076	79.75%	96.88%	90.87%
0.1	0.05	25	SI	327	80.17%	96.88%	91.01%
0.1	0.05	26	SI	615	79.34%	96.65%	90.58%
0.1	0.05	27	SI	290	78.93%	97.10%	90.72%
0.1	0.05	28	SI	213	79.34%	96.43%	90.43%
0.1	0.05	29	SI	193	78.51%	95.31%	89.42%
0.1	0.05	30	SI	292	76.45%	94.87%	88.41%
0.1	0.05	31	SI	327	79.75%	95.76%	90.14%
0.1	0.05	32	SI	210	76.03%	96.43%	89.28%
0.1	0.05	33	SI	349	81.41%	96.65%	91.30%
0.1	0.05	34	SI	370	79.34%	96.43%	90.43%
0.1	0.05	35	SI	257	79.75%	96.21%	90.43%
0.1	0.05	36	SI	303	79.34%	96.88%	90.72%
0.1	0.05	37	SI	330	78.10%	96.43%	90.00%
0.1	0.05	38	SI	278	78.51%	96.43%	90.14%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.1	0.05	39	SI	426	78.93%	96.21%	90.14%
0.1	0.05	40	SI	386	78.93%	96.88%	90.58%
0.1	0.05	41	SI	174	78.93%	96.21%	90.14%
0.1	0.05	42	SI	203	75.62%	95.76%	88.70%
0.1	0.05	43	SI	1,224	79.34%	96.65%	90.58%
0.1	0.05	44	SI	281	78.93%	95.76%	89.86%
0.1	0.05	45	SI	461	82.23%	97.32%	92.03%
0.1	0.05	46	SI	541	79.34%	96.43%	90.43%
0.1	0.05	47	SI	252	79.34%	95.98%	90.14%
0.1	0.05	48	SI	310	78.51%	95.09%	89.28%
0.1	0.05	49	SI	301	77.69%	95.76%	89.42%
0.1	0.05	50	SI	267	77.27%	96.43%	89.71%
0.1	0.1	1	SI	253	79.75%	97.32%	91.16%
0.1	0.1	2	SI	273	79.75%	96.88%	90.87%
0.1	0.1	3	NO	1,959	79.75%	96.43%	90.58%
0.1	0.1	4	SI	223	77.69%	95.76%	89.42%
0.1	0.1	5	SI	645	81.82%	97.10%	91.74%
0.1	0.1	6	SI	414	78.10%	96.43%	90.00%
0.1	0.1	7	SI	321	76.86%	96.43%	89.57%
0.1	0.1	8	SI	706	79.34%	97.10%	90.87%
0.1	0.1	9	SI	407	82.23%	96.88%	91.74%
0.1	0.1	10	SI	345	80.17%	96.65%	90.87%
0.1	0.1	11	SI	457	78.10%	97.10%	90.43%
0.1	0.1	12	SI	272	79.34%	96.43%	90.43%
0.1	0.1	13	SI	199	79.75%	96.21%	90.43%
0.1	0.1	14	SI	278	77.69%	95.76%	89.42%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.1	0.1	15	SI	194	77.69%	96.43%	89.86%
0.1	0.1	16	SI	431	77.69%	96.43%	89.86%
0.1	0.1	17	SI	426	76.86%	95.76%	89.13%
0.1	0.1	18	SI	400	79.75%	96.43%	90.58%
0.1	0.1	19	SI	233	78.51%	96.21%	90.00%
0.1	0.1	20	SI	523	80.17%	96.65%	90.87%
0.1	0.1	21	SI	506	79.34%	95.09%	89.57%
0.1	0.1	22	SI	237	80.58%	94.64%	89.71%
0.1	0.1	23	SI	639	77.27%	96.88%	90.00%
0.1	0.1	24	SI	218	80.58%	95.76%	90.43%
0.1	0.1	25	SI	307	76.45%	97.32%	90.00%
0.1	0.1	26	SI	259	76.03%	97.32%	89.86%
0.1	0.1	27	SI	240	80.99%	96.43%	91.01%
0.1	0.1	28	SI	222	78.10%	97.32%	90.58%
0.1	0.1	29	SI	322	78.51%	96.65%	90.29%
0.1	0.1	30	SI	277	78.51%	95.31%	89.42%
0.1	0.1	31	SI	464	78.93%	96.43%	90.29%
0.1	0.1	32	SI	155	80.99%	96.43%	91.01%
0.1	0.1	33	SI	215	80.17%	96.21%	90.58%
0.1	0.1	34	SI	265	78.93%	95.76%	89.86%
0.1	0.1	35	SI	644	78.10%	94.64%	88.84%
0.1	0.1	36	SI	267	76.86%	94.87%	88.55%
0.1	0.1	37	SI	340	79.34%	95.76%	90.00%
0.1	0.1	38	SI	339	80.58%	96.88%	91.16%
0.1	0.1	39	SI	271	77.69%	95.09%	88.99%
0.1	0.1	40	SI	298	79.34%	97.10%	90.87%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.1	0.1	41	SI	279	78.93%	95.98%	90.00%
0.1	0.1	42	SI	308	74.38%	95.54%	88.12%
0.1	0.1	43	SI	217	77.27%	95.54%	89.13%
0.1	0.1	44	SI	354	79.75%	96.65%	90.72%
0.1	0.1	45	SI	203	80.99%	96.43%	91.01%
0.1	0.1	46	SI	189	76.86%	95.98%	89.28%
0.1	0.1	47	SI	309	76.03%	96.43%	89.28%
0.1	0.1	48	SI	539	77.27%	95.98%	89.42%
0.1	0.1	49	SI	287	78.93%	97.10%	90.72%
0.1	0.1	50	SI	272	79.34%	96.43%	90.43%
0.25	0	1	NO	1,968	80.58%	93.97%	89.28%
0.25	0	2	NO	1,967	83.47%	94.87%	90.87%
0.25	0	3	NO	1,963	78.51%	96.65%	90.29%
0.25	0	4	NO	1,966	83.06%	91.74%	88.70%
0.25	0	5	NO	1,964	82.64%	94.42%	90.29%
0.25	0	6	NO	1,961	80.99%	93.75%	89.28%
0.25	0	7	NO	1,972	39.26%	44.42%	42.61%
0.25	0	8	NO	1,964	80.17%	95.09%	89.86%
0.25	0	9	NO	1,968	83.47%	94.20%	90.43%
0.25	0	10	NO	2,198	83.06%	93.75%	90.00%
0.25	0	11	NO	1,980	81.41%	93.75%	89.42%
0.25	0	12	NO	1,972	81.82%	94.42%	90.00%
0.25	0	13	NO	1,965	79.75%	93.30%	88.55%
0.25	0	14	NO	1,966	80.17%	95.76%	90.29%
0.25	0	15	NO	1,972	83.06%	95.98%	91.45%
0.25	0	16	NO	1,965	81.82%	95.54%	90.72%



Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.25	0	17	NO	1,968	80.99%	95.98%	90.72%
0.25	0	18	NO	1,968	80.99%	96.65%	91.16%
0.25	0	19	NO	1,970	82.64%	96.21%	91.45%
0.25	0	20	NO	1,964	80.99%	95.31%	90.29%
0.25	0	21	NO	2,062	82.64%	92.63%	89.13%
0.25	0	22	NO	1,974	82.23%	94.42%	90.14%
0.25	0	23	NO	1,976	82.23%	92.19%	88.70%
0.25	0	24	NO	1,982	80.17%	92.63%	88.26%
0.25	0	25	NO	1,992	83.06%	94.64%	90.58%
0.25	0	26	NO	1,969	82.23%	93.97%	89.86%
0.25	0	27	NO	1,982	80.99%	95.76%	90.58%
0.25	0	28	NO	1,981	81.82%	94.87%	90.29%
0.25	0	29	NO	1,972	79.34%	89.29%	85.80%
0.25	0	30	NO	1,971	81.41%	94.64%	90.00%
0.25	0	31	NO	1,972	52.48%	54.02%	53.48%
0.25	0	32	NO	1,997	81.41%	93.97%	89.57%
0.25	0	33	NO	1,983	80.58%	94.87%	89.86%
0.25	0	34	NO	1,969	81.82%	93.75%	89.57%
0.25	0	35	NO	1,978	81.82%	91.96%	88.41%
0.25	0	36	NO	1,979	82.64%	95.76%	91.16%
0.25	0	37	NO	1,974	81.82%	94.64%	90.14%
0.25	0	38	NO	1,979	81.82%	95.54%	90.72%
0.25	0	39	NO	1,971	81.82%	94.64%	90.14%
0.25	0	40	NO	1,985	82.23%	94.64%	90.29%
0.25	0	41	NO	1,977	82.64%	95.31%	90.87%
0.25	0	42	NO	1,979	79.75%	89.96%	86.38%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.25	0	43	NO	1,978	42.15%	52.01%	48.55%
0.25	0	44	NO	1,975	81.82%	95.76%	90.87%
0.25	0	45	NO	1,969	81.82%	95.54%	90.72%
0.25	0	46	NO	1,981	82.23%	96.21%	91.30%
0.25	0	47	NO	1,972	83.06%	94.42%	90.43%
0.25	0	48	NO	1,978	80.99%	94.42%	89.71%
0.25	0	49	NO	1,977	80.99%	95.98%	90.72%
0.25	0	50	NO	1,970	83.06%	89.29%	87.10%
0.25	0.01	1	NO	1,986	82.23%	92.63%	88.99%
0.25	0.01	2	NO	1,962	84.30%	95.31%	91.45%
0.25	0.01	3	NO	1,970	82.64%	94.20%	90.14%
0.25	0.01	4	NO	1,984	78.10%	92.63%	87.54%
0.25	0.01	5	NO	1,995	81.82%	93.08%	89.13%
0.25	0.01	6	NO	1,968	81.41%	96.43%	91.16%
0.25	0.01	7	NO	1,970	83.06%	94.64%	90.58%
0.25	0.01	8	NO	1,964	80.17%	89.51%	86.23%
0.25	0.01	9	NO	1,976	82.23%	94.20%	90.00%
0.25	0.01	10	NO	2,178	82.23%	94.64%	90.29%
0.25	0.01	11	NO	1,969	83.47%	95.31%	91.16%
0.25	0.01	12	NO	1,978	82.23%	95.31%	90.72%
0.25	0.01	13	NO	1,974	80.99%	95.76%	90.58%
0.25	0.01	14	NO	1,970	82.23%	94.64%	90.29%
0.25	0.01	15	NO	1,974	81.41%	95.09%	90.29%
0.25	0.01	16	NO	1,969	82.64%	93.97%	90.00%
0.25	0.01	17	NO	1,999	82.23%	87.95%	85.94%
0.25	0.01	18	NO	1,976	81.41%	95.54%	90.58%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.25	0.01	19	NO	1,977	80.58%	93.53%	88.99%
0.25	0.01	20	NO	1,987	82.23%	95.76%	91.01%
0.25	0.01	21	NO	1,969	81.82%	94.64%	90.14%
0.25	0.01	22	NO	1,981	83.06%	94.20%	90.29%
0.25	0.01	23	NO	1,967	81.82%	95.76%	90.87%
0.25	0.01	24	NO	1,973	80.99%	95.54%	90.43%
0.25	0.01	25	NO	1,974	82.23%	95.31%	90.72%
0.25	0.01	26	NO	1,981	83.06%	93.75%	90.00%
0.25	0.01	27	NO	1,979	82.23%	93.97%	89.86%
0.25	0.01	28	NO	1,989	81.82%	93.97%	89.71%
0.25	0.01	29	NO	1,992	80.58%	93.53%	88.99%
0.25	0.01	30	NO	1,976	82.23%	94.87%	90.43%
0.25	0.01	31	NO	1,974	82.23%	94.64%	90.29%
0.25	0.01	32	NO	1,981	80.99%	94.87%	90.00%
0.25	0.01	33	NO	1,981	79.75%	94.64%	89.42%
0.25	0.01	34	NO	1,970	82.23%	94.87%	90.43%
0.25	0.01	35	NO	1,985	81.41%	94.42%	89.86%
0.25	0.01	36	NO	1,975	79.75%	95.98%	90.29%
0.25	0.01	37	NO	1,973	82.64%	94.87%	90.58%
0.25	0.01	38	NO	1,978	80.58%	93.97%	89.28%
0.25	0.01	39	NO	1,982	79.34%	94.87%	89.42%
0.25	0.01	40	NO	1,975	82.23%	93.30%	89.42%
0.25	0.01	41	NO	1,995	82.64%	94.42%	90.29%
0.25	0.01	42	NO	1,981	80.99%	91.74%	87.97%
0.25	0.01	43	NO	1,980	66.12%	71.21%	69.42%
0.25	0.01	44	NO	1,985	80.99%	93.30%	88.99%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.25	0.01	45	NO	1,979	80.58%	94.87%	89.86%
0.25	0.01	46	NO	1,980	80.58%	95.09%	90.00%
0.25	0.01	47	NO	2,282	80.17%	95.31%	90.00%
0.25	0.01	48	NO	1,982	47.52%	58.48%	54.64%
0.25	0.01	49	NO	1,989	80.99%	94.42%	89.71%
0.25	0.01	50	NO	1,994	81.82%	93.08%	89.13%
0.25	0.05	1	NO	1,987	83.06%	89.29%	87.10%
0.25	0.05	2	NO	1,974	82.64%	94.20%	90.14%
0.25	0.05	3	NO	1,975	81.82%	93.53%	89.42%
0.25	0.05	4	NO	1,976	83.06%	93.53%	89.86%
0.25	0.05	5	NO	1,975	79.34%	90.85%	86.81%
0.25	0.05	6	NO	1,996	82.64%	89.06%	86.81%
0.25	0.05	7	NO	1,974	83.88%	94.20%	90.58%
0.25	0.05	8	NO	1,984	80.99%	95.31%	90.29%
0.25	0.05	9	NO	1,981	79.34%	93.75%	88.70%
0.25	0.05	10	NO	1,969	82.64%	93.97%	90.00%
0.25	0.05	11	NO	1,994	82.64%	93.75%	89.86%
0.25	0.05	12	NO	1,986	80.99%	91.29%	87.68%
0.25	0.05	13	NO	1,971	80.17%	89.73%	86.38%
0.25	0.05	14	NO	1,974	80.99%	95.31%	90.29%
0.25	0.05	15	NO	1,988	83.47%	94.64%	90.72%
0.25	0.05	16	NO	1,980	82.64%	93.53%	89.71%
0.25	0.05	17	NO	1,995	79.34%	85.71%	83.48%
0.25	0.05	18	NO	1,970	83.47%	94.42%	90.58%
0.25	0.05	19	NO	1,980	80.58%	92.19%	88.12%
0.25	0.05	20	NO	1,997	81.82%	87.95%	85.80%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.25	0.05	21	NO	1,994	83.47%	92.19%	89.13%
0.25	0.05	22	NO	1,993	79.75%	88.17%	85.22%
0.25	0.05	23	NO	1,982	82.23%	94.87%	90.43%
0.25	0.05	24	NO	1,984	79.34%	93.08%	88.26%
0.25	0.05	25	NO	1,989	82.23%	93.97%	89.86%
0.25	0.05	26	NO	1,989	84.30%	93.75%	90.43%
0.25	0.05	27	NO	1,980	81.82%	94.87%	90.29%
0.25	0.05	28	NO	1,995	83.06%	91.52%	88.55%
0.25	0.05	29	NO	1,983	82.23%	94.20%	90.00%
0.25	0.05	30	NO	1,983	81.82%	94.20%	89.86%
0.25	0.05	31	NO	1,978	82.64%	93.75%	89.86%
0.25	0.05	32	NO	1,989	83.47%	92.63%	89.42%
0.25	0.05	33	NO	1,987	81.41%	92.63%	88.70%
0.25	0.05	34	NO	1,985	81.82%	93.53%	89.42%
0.25	0.05	35	NO	1,992	82.64%	94.42%	90.29%
0.25	0.05	36	NO	1,982	82.64%	93.75%	89.86%
0.25	0.05	37	NO	1,994	80.99%	94.87%	90.00%
0.25	0.05	38	NO	1,972	44.63%	51.34%	48.99%
0.25	0.05	39	NO	1,987	81.41%	92.63%	88.70%
0.25	0.05	40	NO	2,124	74.79%	85.71%	81.88%
0.25	0.05	41	NO	1,986	54.96%	55.80%	55.51%
0.25	0.05	42	NO	2,001	79.34%	91.96%	87.54%
0.25	0.05	43	NO	1,983	80.99%	94.20%	89.57%
0.25	0.05	44	NO	1,993	83.47%	94.20%	90.43%
0.25	0.05	45	NO	1,998	80.58%	88.39%	85.65%
0.25	0.05	46	NO	1,985	81.82%	94.42%	90.00%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.25	0.05	47	NO	2,084	3.31%	1.79%	2.32%
0.25	0.05	48	NO	1,978	83.47%	94.20%	90.43%
0.25	0.05	49	NO	1,978	82.64%	94.42%	90.29%
0.25	0.05	50	NO	2,001	42.15%	46.88%	45.22%
0.25	0.1	1	NO	2,001	83.47%	92.41%	89.28%
0.25	0.1	2	NO	1,985	80.58%	90.85%	87.25%
0.25	0.1	3	NO	1,982	42.15%	45.54%	44.35%
0.25	0.1	4	NO	1,994	13.64%	10.27%	11.45%
0.25	0.1	5	NO	1,985	73.55%	83.26%	79.86%
0.25	0.1	6	NO	1,983	82.64%	94.20%	90.14%
0.25	0.1	7	NO	1,977	16.12%	8.93%	11.45%
0.25	0.1	8	NO	1,977	81.82%	92.41%	88.70%
0.25	0.1	9	NO	1,979	83.06%	93.75%	90.00%
0.25	0.1	10	NO	2,008	45.87%	51.12%	49.28%
0.25	0.1	11	NO	1,989	45.04%	48.21%	47.10%
0.25	0.1	12	NO	1,981	81.41%	88.84%	86.23%
0.25	0.1	13	NO	1,995	80.58%	91.07%	87.39%
0.25	0.1	14	NO	1,980	80.58%	92.63%	88.41%
0.25	0.1	15	NO	1,982	81.41%	93.97%	89.57%
0.25	0.1	16	NO	1,993	84.30%	94.87%	91.16%
0.25	0.1	17	NO	1,981	83.47%	92.19%	89.13%
0.25	0.1	18	NO	1,994	81.82%	92.63%	88.84%
0.25	0.1	19	NO	1,982	83.06%	94.64%	90.58%
0.25	0.1	20	NO	1,989	83.06%	91.07%	88.26%
0.25	0.1	21	NO	1,983	82.23%	90.40%	87.54%
0.25	0.1	22	NO	2,005	81.41%	93.97%	89.57%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.25	0.1	23	NO	1,986	83.88%	93.97%	90.43%
0.25	0.1	24	NO	1,995	82.64%	93.30%	89.57%
0.25	0.1	25	NO	2,004	71.07%	77.46%	75.22%
0.25	0.1	26	NO	1,986	83.06%	95.54%	91.16%
0.25	0.1	27	NO	1,983	81.82%	93.53%	89.42%
0.25	0.1	28	NO	2,014	54.13%	54.02%	54.06%
0.25	0.1	29	NO	2,013	81.82%	93.30%	89.28%
0.25	0.1	30	NO	1,985	80.99%	90.85%	87.39%
0.25	0.1	31	NO	2,001	82.23%	91.74%	88.41%
0.25	0.1	32	NO	1,983	81.82%	87.95%	85.80%
0.25	0.1	33	NO	2,000	83.88%	92.41%	89.42%
0.25	0.1	34	NO	1,999	83.47%	91.07%	88.41%
0.25	0.1	35	NO	2,001	81.82%	90.63%	87.54%
0.25	0.1	36	NO	1,995	82.23%	88.39%	86.23%
0.25	0.1	37	NO	1,998	82.64%	92.86%	89.28%
0.25	0.1	38	NO	1,996	82.23%	90.18%	87.39%
0.25	0.1	39	NO	1,992	80.58%	93.30%	88.84%
0.25	0.1	40	NO	2,275	79.75%	92.63%	88.12%
0.25	0.1	41	NO	2,012	81.82%	87.95%	85.80%
0.25	0.1	42	NO	1,990	81.82%	89.73%	86.96%
0.25	0.1	43	NO	1,999	81.82%	93.53%	89.42%
0.25	0.1	44	NO	2,005	81.82%	91.07%	87.83%
0.25	0.1	45	NO	2,006	82.23%	90.18%	87.39%
0.25	0.1	46	NO	2,003	80.17%	91.07%	87.25%
0.25	0.1	47	NO	2,016	81.82%	90.85%	87.68%
0.25	0.1	48	NO	1,989	84.30%	93.97%	90.58%

Alfa	Beta	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
0.25	0.1	49	NO	1,995	82.64%	91.29%	88.26%
0.25	0.1	50	NO	1,990	82.64%	91.29%	88.26%

## A.2. Valores de inicialización óptimos

Min Inicialización	Max Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
-5	5	1	NO	1,980	80.58%	94.42%	89.57%
-5	5	2	NO	1,970	80.17%	96.21%	90.58%
-5	5	3	NO	1,968	81.41%	95.98%	90.87%
-5	5	4	NO	1,973	79.34%	91.74%	87.39%
-5	5	5	NO	1,969	80.17%	96.21%	90.58%
-5	5	6	NO	1,966	78.93%	95.54%	89.71%
-5	5	7	NO	1,971	80.58%	94.20%	89.42%
-5	5	8	NO	1,968	76.86%	90.63%	85.80%
-5	5	9	NO	1,966	77.69%	92.86%	87.54%
-5	5	10	NO	1,973	78.10%	96.65%	90.14%
-5	5	11	NO	1,972	78.93%	95.31%	89.57%
-5	5	12	NO	1,968	80.58%	94.42%	89.57%
-5	5	13	NO	2,440	80.58%	94.64%	89.71%
-5	5	14	NO	1,968	79.75%	94.20%	89.13%
-5	5	15	NO	1,970	77.27%	90.85%	86.09%
-5	5	16	NO	1,969	73.14%	87.05%	82.17%
-5	5	17	NO	1,973	81.41%	96.43%	91.16%
-5	5	18	NO	1,977	80.99%	94.87%	90.00%
-5	5	19	NO	1,973	75.21%	90.85%	85.36%
-5	5	20	NO	1,974	82.23%	93.53%	89.57%



Min Inicialización	Max Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
-5	5	21	NO	1,967	77.69%	96.43%	89.86%
-5	5	22	NO	1,971	78.10%	96.43%	90.00%
-5	5	23	NO	1,972	79.75%	92.86%	88.26%
-5	5	24	NO	1,971	81.41%	95.09%	90.29%
-5	5	25	NO	1,972	82.64%	94.87%	90.58%
-5	5	26	NO	1,973	80.17%	89.73%	86.38%
-5	5	27	NO	1,977	80.17%	94.87%	89.71%
-5	5	28	NO	1,968	78.93%	95.09%	89.42%
-5	5	29	NO	1,968	79.34%	96.88%	90.72%
-5	5	30	NO	1,970	79.34%	93.08%	88.26%
-5	5	31	NO	1,972	81.82%	95.76%	90.87%
-5	5	32	NO	1,966	78.10%	93.08%	87.83%
-5	5	33	NO	1,972	79.75%	96.21%	90.43%
-5	5	34	NO	1,965	83.06%	92.19%	88.99%
-5	5	35	NO	1,969	80.58%	91.74%	87.83%
-5	5	36	NO	1,967	80.58%	95.09%	90.00%
-5	5	37	NO	1,968	71.07%	96.65%	87.68%
-5	5	38	NO	1,971	80.58%	93.53%	88.99%
-5	5	39	NO	1,974	80.17%	93.30%	88.70%
-5	5	40	NO	1,968	81.82%	96.21%	91.16%
-5	5	41	NO	1,967	78.93%	95.76%	89.86%
-5	5	42	NO	1,975	78.51%	95.09%	89.28%
-5	5	43	NO	1,969	77.27%	97.32%	90.29%
-5	5	44	NO	1,969	76.03%	95.98%	88.99%
-5	5	45	NO	1,972	75.62%	95.09%	88.26%
-5	5	46	NO	1,974	76.03%	90.63%	85.51%
-5	5	47	NO	1,970	77.27%	96.65%	89.86%

Min Inicialización	Max Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
-5	5	48	NO	1,973	77.27%	96.21%	89.57%
-5	5	49	NO	1,970	77.27%	95.54%	89.13%
-5	5	50	NO	1,968	81.82%	92.86%	88.99%
-1	1	1	SI	263	79.34%	96.21%	90.29%
-1	1	2	SI	331	79.75%	96.88%	90.87%
-1	1	3	SI	152	77.27%	97.10%	90.14%
-1	1	4	SI	313	79.75%	96.88%	90.87%
-1	1	5	SI	245	78.93%	96.65%	90.43%
-1	1	6	SI	298	79.75%	95.54%	90.00%
-1	1	7	SI	187	76.45%	96.88%	89.71%
-1	1	8	SI	386	77.69%	95.98%	89.57%
-1	1	9	SI	359	79.34%	95.98%	90.14%
-1	1	10	SI	167	80.99%	96.43%	91.01%
-1	1	11	SI	317	79.34%	96.21%	90.29%
-1	1	12	SI	256	78.10%	95.54%	89.42%
-1	1	13	SI	349	77.27%	96.21%	89.57%
-1	1	14	SI	281	78.93%	95.76%	89.86%
-1	1	15	SI	360	75.21%	96.43%	88.99%
-1	1	16	SI	141	78.93%	96.88%	90.58%
-1	1	17	SI	382	79.75%	96.65%	90.72%
-1	1	18	SI	303	75.62%	97.32%	89.71%
-1	1	19	SI	465	76.86%	96.65%	89.71%
-1	1	20	SI	196	79.34%	96.43%	90.43%
-1	1	21	SI	362	77.27%	96.43%	89.71%
-1	1	22	SI	203	78.10%	95.98%	89.71%
-1	1	23	SI	350	77.27%	96.43%	89.71%
-1	1	24	SI	165	78.10%	94.64%	88.84%

Min Inicialización	Max Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
-1	1	25	SI	213	77.27%	95.54%	89.13%
-1	1	26	SI	400	78.93%	95.76%	89.86%
-1	1	27	SI	332	77.69%	95.98%	89.57%
-1	1	28	SI	296	79.75%	96.43%	90.58%
-1	1	29	SI	293	78.93%	96.88%	90.58%
-1	1	30	SI	214	78.10%	95.31%	89.28%
-1	1	31	SI	194	75.62%	96.43%	89.13%
-1	1	32	SI	124	78.51%	96.43%	90.14%
-1	1	33	SI	260	76.86%	95.98%	89.28%
-1	1	34	SI	213	76.86%	95.98%	89.28%
-1	1	35	SI	331	80.99%	96.65%	91.16%
-1	1	36	SI	599	78.51%	95.98%	89.86%
-1	1	37	SI	220	81.41%	95.31%	90.43%
-1	1	38	SI	194	75.62%	94.64%	87.97%
-1	1	39	SI	237	80.17%	96.43%	90.72%
-1	1	40	SI	244	76.45%	95.09%	88.55%
-1	1	41	SI	136	76.03%	96.65%	89.42%
-1	1	42	SI	251	78.51%	96.65%	90.29%
-1	1	43	SI	214	80.99%	95.09%	90.14%
-1	1	44	SI	387	79.75%	94.87%	89.57%
-1	1	45	SI	438	75.21%	95.31%	88.26%
-1	1	46	SI	380	79.75%	95.54%	90.00%
-1	1	47	SI	1,609	77.69%	96.65%	90.00%
-1	1	48	SI	253	78.51%	97.10%	90.58%
-1	1	49	SI	173	76.45%	95.09%	88.55%
-1	1	50	SI	260	76.45%	95.98%	89.13%
-0.1	0.1	1	SI	529	78.93%	95.98%	90.00%

Min Inicialización	Max Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
-0.1	0.1	2	SI	677	78.10%	96.21%	89.86%
-0.1	0.1	3	SI	585	80.58%	96.21%	90.72%
-0.1	0.1	4	SI	682	80.99%	95.54%	90.43%
-0.1	0.1	5	SI	474	77.27%	96.88%	90.00%
-0.1	0.1	6	SI	706	80.58%	96.65%	91.01%
-0.1	0.1	7	SI	560	78.93%	95.98%	90.00%
-0.1	0.1	8	SI	588	78.51%	96.65%	90.29%
-0.1	0.1	9	SI	616	76.86%	96.21%	89.42%
-0.1	0.1	10	SI	801	79.75%	96.21%	90.43%
-0.1	0.1	11	SI	523	77.27%	96.21%	89.57%
-0.1	0.1	12	SI	527	78.10%	94.42%	88.70%
-0.1	0.1	13	SI	518	79.34%	96.88%	90.72%
-0.1	0.1	14	SI	606	75.62%	94.87%	88.12%
-0.1	0.1	15	SI	468	78.10%	96.21%	89.86%
-0.1	0.1	16	SI	573	81.41%	95.98%	90.87%
-0.1	0.1	17	SI	593	80.17%	95.76%	90.29%
-0.1	0.1	18	SI	662	80.17%	95.98%	90.43%
-0.1	0.1	19	SI	531	82.23%	96.88%	91.74%
-0.1	0.1	20	SI	435	78.93%	97.10%	90.72%
-0.1	0.1	21	SI	535	79.75%	96.43%	90.58%
-0.1	0.1	22	SI	562	79.75%	95.76%	90.14%
-0.1	0.1	23	SI	593	79.75%	96.43%	90.58%
-0.1	0.1	24	SI	383	77.27%	97.10%	90.14%
-0.1	0.1	25	SI	624	76.45%	95.76%	88.99%
-0.1	0.1	26	SI	471	80.17%	96.43%	90.72%
-0.1	0.1	27	SI	474	77.27%	96.65%	89.86%
-0.1	0.1	28	SI	425	80.58%	96.21%	90.72%

Min Inicialización	Max Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
-0.1	0.1	29	SI	476	79.34%	96.88%	90.72%
-0.1	0.1	30	SI	563	77.69%	95.54%	89.28%
-0.1	0.1	31	SI	876	78.93%	96.43%	90.29%
-0.1	0.1	32	SI	514	78.51%	95.54%	89.57%
-0.1	0.1	33	SI	544	78.10%	94.64%	88.84%
-0.1	0.1	34	SI	410	80.17%	96.88%	91.01%
-0.1	0.1	35	SI	708	78.93%	95.98%	90.00%
-0.1	0.1	36	SI	564	79.34%	96.43%	90.43%
-0.1	0.1	37	SI	752	78.93%	96.88%	90.58%
-0.1	0.1	38	SI	572	80.17%	95.98%	90.43%
-0.1	0.1	39	SI	356	78.93%	96.65%	90.43%
-0.1	0.1	40	SI	553	79.34%	96.43%	90.43%
-0.1	0.1	41	SI	491	80.17%	96.88%	91.01%
-0.1	0.1	42	SI	538	80.99%	96.88%	91.30%
-0.1	0.1	43	SI	560	79.75%	96.65%	90.72%
-0.1	0.1	44	SI	309	78.10%	96.43%	90.00%
-0.1	0.1	45	SI	746	80.58%	96.65%	91.01%
-0.1	0.1	46	SI	525	78.10%	95.76%	89.57%
-0.1	0.1	47	SI	585	78.93%	96.43%	90.29%
-0.1	0.1	48	SI	543	79.34%	96.21%	90.29%
-0.1	0.1	49	SI	638	80.17%	96.88%	91.01%
-0.1	0.1	50	SI	389	77.27%	97.10%	90.14%
-0.01	0.01	1	SI	667	80.58%	97.10%	91.30%
-0.01	0.01	2	SI	827	79.34%	95.54%	89.86%
-0.01	0.01	3	SI	722	78.10%	94.87%	88.99%
-0.01	0.01	4	SI	584	79.75%	96.21%	90.43%
-0.01	0.01	5	SI	605	78.93%	96.43%	90.29%

Min Inicialización	Max Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
-0.01	0.01	6	SI	586	78.93%	95.98%	90.00%
-0.01	0.01	7	SI	685	78.93%	96.88%	90.58%
-0.01	0.01	8	SI	697	77.27%	96.43%	89.71%
-0.01	0.01	9	SI	785	78.93%	96.65%	90.43%
-0.01	0.01	10	SI	729	78.51%	96.65%	90.29%
-0.01	0.01	11	SI	705	79.75%	96.21%	90.43%
-0.01	0.01	12	SI	424	78.93%	95.31%	89.57%
-0.01	0.01	13	SI	636	80.58%	96.21%	90.72%
-0.01	0.01	14	SI	556	80.17%	97.10%	91.16%
-0.01	0.01	15	SI	475	76.45%	96.65%	89.57%
-0.01	0.01	16	SI	573	76.45%	96.65%	89.57%
-0.01	0.01	17	SI	753	80.58%	96.65%	91.01%
-0.01	0.01	18	SI	771	80.58%	96.65%	91.01%
-0.01	0.01	19	SI	579	76.86%	96.65%	89.71%
-0.01	0.01	20	SI	872	80.17%	95.54%	90.14%
-0.01	0.01	21	SI	665	79.75%	95.76%	90.14%
-0.01	0.01	22	SI	468	79.75%	96.65%	90.72%
-0.01	0.01	23	SI	636	81.82%	96.65%	91.45%
-0.01	0.01	24	SI	717	79.34%	96.43%	90.43%
-0.01	0.01	25	SI	877	80.17%	95.76%	90.29%
-0.01	0.01	26	SI	886	78.10%	96.43%	90.00%
-0.01	0.01	27	SI	643	78.93%	96.21%	90.14%
-0.01	0.01	28	SI	760	79.34%	96.65%	90.58%
-0.01	0.01	29	SI	439	78.51%	96.88%	90.43%
-0.01	0.01	30	SI	734	81.82%	96.88%	91.59%
-0.01	0.01	31	SI	636	77.69%	95.76%	89.42%
-0.01	0.01	32	SI	597	79.75%	96.21%	90.43%

Min Inicialización	Max Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
-0.01	0.01	33	SI	638	79.34%	96.21%	90.29%
-0.01	0.01	34	SI	728	80.17%	97.10%	91.16%
-0.01	0.01	35	SI	908	80.58%	96.88%	91.16%
-0.01	0.01	36	SI	790	80.17%	97.10%	91.16%
-0.01	0.01	37	SI	608	77.69%	95.31%	89.13%
-0.01	0.01	38	SI	647	77.69%	96.65%	90.00%
-0.01	0.01	39	SI	651	77.27%	95.76%	89.28%
-0.01	0.01	40	SI	567	79.34%	96.43%	90.43%
-0.01	0.01	41	SI	948	79.34%	97.10%	90.87%
-0.01	0.01	42	SI	669	79.75%	96.43%	90.58%
-0.01	0.01	43	SI	878	77.27%	95.98%	89.42%
-0.01	0.01	44	SI	771	79.34%	96.43%	90.43%
-0.01	0.01	45	SI	884	79.34%	96.21%	90.29%
-0.01	0.01	46	SI	723	81.82%	96.88%	91.59%
-0.01	0.01	47	SI	723	79.34%	97.10%	90.87%
-0.01	0.01	48	SI	483	78.10%	96.43%	90.00%
-0.01	0.01	49	SI	717	79.75%	96.88%	90.87%
-0.01	0.01	50	SI	473	78.93%	96.88%	90.58%

### **A.3. Función de selección**

Función Selección	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Selección proporcional	1	SI	251	80.17%	96.65%	90.87%
Selección proporcional	2	SI	720	79.34%	97.32%	91.01%
Selección proporcional	3	SI	261	74.38%	94.87%	87.68%
Selección proporcional	4	SI	213	80.17%	95.76%	90.29%
Selección proporcional	5	SI	226	76.45%	95.76%	88.99%

<b>Función Selección</b>	<b>Corrida</b>	<b>Ent?</b>	<b>Tiempo Prom. Ent.</b>	<b>Aciertos Prom. Eval.</b>	<b>Aciertos Prom. Ent.</b>	<b>Aciertos Prom. Total</b>
Selección proporcional	6	SI	173	77.69%	96.43%	89.86%
Selección proporcional	7	SI	340	75.62%	95.98%	88.84%
Selección proporcional	8	SI	173	79.34%	95.31%	89.71%
Selección proporcional	9	SI	232	79.34%	96.88%	90.72%
Selección proporcional	10	SI	307	80.99%	96.88%	91.30%
Selección proporcional	11	NO	773	82.64%	95.31%	90.87%
Selección proporcional	12	SI	191	76.86%	95.54%	88.99%
Selección proporcional	13	SI	260	78.93%	96.65%	90.43%
Selección proporcional	14	SI	289	78.51%	95.31%	89.42%
Selección proporcional	15	SI	209	78.93%	95.98%	90.00%
Selección proporcional	16	SI	276	77.69%	95.76%	89.42%
Selección proporcional	17	SI	295	78.51%	96.65%	90.29%
Selección proporcional	18	SI	355	78.10%	96.65%	90.14%
Selección proporcional	19	SI	246	77.27%	96.21%	89.57%
Selección proporcional	20	SI	150	80.99%	95.98%	90.72%
Selección proporcional	21	SI	560	78.10%	96.88%	90.29%
Selección proporcional	22	SI	155	76.86%	95.98%	89.28%
Selección proporcional	23	SI	200	79.34%	95.98%	90.14%
Selección proporcional	24	SI	279	78.93%	94.64%	89.13%
Selección proporcional	25	SI	467	78.10%	95.98%	89.71%
Selección proporcional	26	SI	136	76.86%	93.97%	87.97%
Selección proporcional	27	SI	200	76.45%	96.43%	89.42%
Selección proporcional	28	SI	240	77.69%	95.98%	89.57%
Selección proporcional	29	SI	248	75.21%	95.98%	88.70%
Selección proporcional	30	SI	550	80.99%	97.32%	91.59%
Selección proporcional	31	SI	206	72.31%	94.42%	86.67%
Selección proporcional	32	SI	483	80.99%	96.88%	91.30%



<b>Función Selección</b>	<b>Corrida</b>	<b>Ent?</b>	<b>Tiempo Prom. Ent.</b>	<b>Aciertos Prom. Eval.</b>	<b>Aciertos Prom. Ent.</b>	<b>Aciertos Prom. Total</b>
Selección proporcional	33	NO	774	83.88%	97.10%	92.46%
Selección proporcional	34	SI	376	78.93%	96.88%	90.58%
Selección proporcional	35	SI	244	78.93%	96.65%	90.43%
Selección proporcional	36	SI	201	77.27%	96.65%	89.86%
Selección proporcional	37	SI	310	80.17%	96.65%	90.87%
Selección proporcional	38	SI	210	76.86%	96.21%	89.42%
Selección proporcional	39	SI	204	77.27%	95.54%	89.13%
Selección proporcional	40	SI	140	79.34%	97.10%	90.87%
Selección proporcional	41	SI	236	76.03%	96.21%	89.13%
Selección proporcional	42	SI	376	76.45%	94.87%	88.41%
Selección proporcional	43	SI	262	79.75%	96.65%	90.72%
Selección proporcional	44	SI	218	76.03%	96.43%	89.28%
Selección proporcional	45	SI	239	79.75%	95.98%	90.29%
Selección proporcional	46	SI	506	81.82%	97.32%	91.88%
Selección proporcional	47	SI	225	76.03%	94.20%	87.83%
Selección proporcional	48	SI	147	77.69%	95.31%	89.13%
Selección proporcional	49	SI	154	79.75%	95.09%	89.71%
Selección proporcional	50	SI	310	76.03%	96.21%	89.13%
Selección ruleta	1	SI	257	78.51%	96.43%	90.14%
Selección ruleta	2	SI	529	78.51%	96.88%	90.43%
Selección ruleta	3	SI	393	78.10%	96.21%	89.86%
Selección ruleta	4	SI	263	76.86%	95.98%	89.28%
Selección ruleta	5	SI	259	77.69%	97.32%	90.43%
Selección ruleta	6	SI	409	78.10%	96.65%	90.14%
Selección ruleta	7	SI	274	77.69%	95.98%	89.57%
Selección ruleta	8	SI	357	76.03%	95.54%	88.70%
Selección ruleta	9	SI	216	79.34%	96.21%	90.29%

<b>Función Selección</b>	<b>Corrida</b>	<b>Ent?</b>	<b>Tiempo Prom. Ent.</b>	<b>Aciertos Prom. Eval.</b>	<b>Aciertos Prom. Ent.</b>	<b>Aciertos Prom. Total</b>
Selección ruleta	10	SI	229	77.69%	95.54%	89.28%
Selección ruleta	11	SI	223	80.17%	95.54%	90.14%
Selección ruleta	12	SI	200	76.45%	96.88%	89.71%
Selección ruleta	13	SI	223	75.21%	96.21%	88.84%
Selección ruleta	14	SI	282	75.21%	95.09%	88.12%
Selección ruleta	15	SI	466	80.99%	95.54%	90.43%
Selección ruleta	16	SI	269	76.86%	95.09%	88.70%
Selección ruleta	17	SI	300	77.69%	95.09%	88.99%
Selección ruleta	18	SI	272	77.27%	95.76%	89.28%
Selección ruleta	19	SI	256	76.03%	96.21%	89.13%
Selección ruleta	20	SI	202	78.10%	96.21%	89.86%
Selección ruleta	21	SI	327	79.34%	96.43%	90.43%
Selección ruleta	22	SI	475	79.34%	95.54%	89.86%
Selección ruleta	23	SI	307	76.86%	96.43%	89.57%
Selección ruleta	24	SI	242	74.79%	96.21%	88.70%
Selección ruleta	25	SI	300	80.58%	95.98%	90.58%
Selección ruleta	26	SI	541	80.17%	96.88%	91.01%
Selección ruleta	27	SI	319	80.58%	96.21%	90.72%
Selección ruleta	28	SI	216	80.99%	96.65%	91.16%
Selección ruleta	29	SI	257	76.03%	95.98%	88.99%
Selección ruleta	30	SI	351	84.30%	96.43%	92.17%
Selección ruleta	31	SI	288	76.45%	96.21%	89.28%
Selección ruleta	32	SI	211	79.75%	95.98%	90.29%
Selección ruleta	33	SI	253	77.69%	96.21%	89.71%
Selección ruleta	34	SI	229	78.10%	96.65%	90.14%
Selección ruleta	35	SI	199	78.51%	96.21%	90.00%
Selección ruleta	36	SI	385	79.34%	96.43%	90.43%

<b>Función Selección</b>	<b>Corrida</b>	<b>Ent?</b>	<b>Tiempo Prom. Ent.</b>	<b>Aciertos Prom. Eval.</b>	<b>Aciertos Prom. Ent.</b>	<b>Aciertos Prom. Total</b>
Selección ruleta	37	SI	277	78.10%	96.43%	90.00%
Selección ruleta	38	SI	510	80.58%	96.88%	91.16%
Selección ruleta	39	SI	314	76.45%	96.43%	89.42%
Selección ruleta	40	SI	309	78.93%	95.98%	90.00%
Selección ruleta	41	SI	381	79.75%	96.43%	90.58%
Selección ruleta	42	SI	273	80.17%	96.65%	90.87%
Selección ruleta	43	SI	332	77.27%	96.43%	89.71%
Selección ruleta	44	SI	531	76.45%	95.98%	89.13%
Selección ruleta	45	SI	754	82.23%	96.65%	91.59%
Selección ruleta	46	SI	399	77.69%	96.21%	89.71%
Selección ruleta	47	SI	431	78.10%	96.65%	90.14%
Selección ruleta	48	SI	388	75.62%	96.21%	88.99%
Selección ruleta	49	SI	239	75.62%	95.98%	88.84%
Selección ruleta	50	SI	356	78.10%	96.65%	90.14%
Selección torneo	1	SI	189	79.75%	96.43%	90.58%
Selección torneo	2	SI	202	76.86%	96.21%	89.42%
Selección torneo	3	SI	511	78.93%	97.10%	90.72%
Selección torneo	4	SI	164	78.93%	95.54%	89.71%
Selección torneo	5	SI	186	76.45%	95.76%	88.99%
Selección torneo	6	SI	160	76.45%	96.43%	89.42%
Selección torneo	7	SI	216	77.69%	95.31%	89.13%
Selección torneo	8	SI	201	76.86%	96.43%	89.57%
Selección torneo	9	SI	186	79.75%	96.21%	90.43%
Selección torneo	10	SI	200	80.58%	96.65%	91.01%
Selección torneo	11	SI	166	79.34%	96.65%	90.58%
Selección torneo	12	SI	409	75.62%	95.98%	88.84%
Selección torneo	13	SI	183	79.34%	94.87%	89.42%

<b>Función Selección</b>	<b>Corrida</b>	<b>Ent?</b>	<b>Tiempo Prom. Ent.</b>	<b>Aciertos Prom. Eval.</b>	<b>Aciertos Prom. Ent.</b>	<b>Aciertos Prom. Total</b>
Selección torneo	14	SI	197	77.69%	95.76%	89.42%
Selección torneo	15	SI	420	77.27%	94.64%	88.55%
Selección torneo	16	SI	149	76.45%	95.09%	88.55%
Selección torneo	17	SI	257	76.03%	94.64%	88.12%
Selección torneo	18	SI	429	78.10%	94.87%	88.99%
Selección torneo	19	SI	169	77.27%	95.54%	89.13%
Selección torneo	20	SI	205	76.86%	96.21%	89.42%
Selección torneo	21	SI	184	77.69%	96.21%	89.71%
Selección torneo	22	SI	168	78.93%	96.21%	90.14%
Selección torneo	23	SI	272	80.17%	96.43%	90.72%
Selección torneo	24	SI	162	79.75%	96.88%	90.87%
Selección torneo	25	SI	172	78.10%	95.76%	89.57%
Selección torneo	26	SI	206	76.45%	96.65%	89.57%
Selección torneo	27	SI	178	78.10%	96.65%	90.14%
Selección torneo	28	SI	165	78.51%	96.21%	90.00%
Selección torneo	29	SI	205	78.93%	95.31%	89.57%
Selección torneo	30	SI	163	79.34%	95.76%	90.00%
Selección torneo	31	SI	179	76.45%	96.65%	89.57%
Selección torneo	32	SI	129	79.34%	94.87%	89.42%
Selección torneo	33	SI	144	76.86%	95.31%	88.84%
Selección torneo	34	SI	181	78.51%	96.65%	90.29%
Selección torneo	35	SI	157	78.10%	95.09%	89.13%
Selección torneo	36	SI	201	76.45%	95.76%	88.99%
Selección torneo	37	SI	229	81.41%	95.54%	90.58%
Selección torneo	38	SI	248	78.93%	95.54%	89.71%
Selección torneo	39	SI	158	78.93%	96.65%	90.43%
Selección torneo	40	SI	216	76.86%	94.64%	88.41%

<b>Función Selección</b>	<b>Corrida</b>	<b>Ent?</b>	<b>Tiempo Prom. Ent.</b>	<b>Aciertos Prom. Eval.</b>	<b>Aciertos Prom. Ent.</b>	<b>Aciertos Prom. Total</b>
Selección torneo	41	SI	171	79.34%	96.21%	90.29%
Selección torneo	42	SI	329	78.51%	95.31%	89.42%
Selección torneo	43	SI	148	80.58%	95.98%	90.58%
Selección torneo	44	SI	196	78.93%	95.76%	89.86%
Selección torneo	45	SI	185	78.10%	96.21%	89.86%
Selección torneo	46	SI	134	77.69%	96.21%	89.71%
Selección torneo	47	SI	221	78.93%	96.21%	90.14%
Selección torneo	48	SI	166	75.21%	94.87%	87.97%
Selección torneo	49	SI	159	74.79%	95.98%	88.55%
Selección torneo	50	SI	176	76.86%	95.09%	88.70%
Selección ranking-ruleta	1	SI	370	75.62%	96.21%	88.99%
Selección ranking-ruleta	2	SI	231	80.58%	95.54%	90.29%
Selección ranking-ruleta	3	SI	293	78.10%	97.10%	90.43%
Selección ranking-ruleta	4	SI	153	78.93%	95.09%	89.42%
Selección ranking-ruleta	5	SI	498	79.75%	96.65%	90.72%
Selección ranking-ruleta	6	SI	205	77.27%	97.10%	90.14%
Selección ranking-ruleta	7	SI	216	76.45%	96.43%	89.42%
Selección ranking-ruleta	8	SI	242	79.75%	96.88%	90.87%
Selección ranking-ruleta	9	SI	288	78.93%	96.43%	90.29%
Selección ranking-ruleta	10	SI	260	78.93%	96.21%	90.14%
Selección ranking-ruleta	11	SI	238	77.27%	96.21%	89.57%
Selección ranking-ruleta	12	SI	166	79.75%	95.54%	90.00%
Selección ranking-ruleta	13	SI	240	78.93%	96.65%	90.43%
Selección ranking-ruleta	14	SI	251	76.86%	95.98%	89.28%
Selección ranking-ruleta	15	SI	213	77.69%	96.65%	90.00%
Selección ranking-ruleta	16	SI	245	75.62%	95.09%	88.26%
Selección ranking-ruleta	17	SI	455	78.10%	95.98%	89.71%

<b>Función Selección</b>	<b>Corrida</b>	<b>Ent?</b>	<b>Tiempo Prom. Ent.</b>	<b>Aciertos Prom. Eval.</b>	<b>Aciertos Prom. Ent.</b>	<b>Aciertos Prom. Total</b>
Selección ranking-ruleta	18	SI	186	76.86%	95.76%	89.13%
Selección ranking-ruleta	19	SI	324	77.69%	95.31%	89.13%
Selección ranking-ruleta	20	SI	211	76.86%	96.43%	89.57%
Selección ranking-ruleta	21	SI	263	76.86%	95.76%	89.13%
Selección ranking-ruleta	22	SI	284	77.69%	93.08%	87.68%
Selección ranking-ruleta	23	SI	234	76.45%	95.76%	88.99%
Selección ranking-ruleta	24	SI	186	79.75%	95.98%	90.29%
Selección ranking-ruleta	25	SI	313	78.10%	95.76%	89.57%
Selección ranking-ruleta	26	SI	266	76.86%	96.43%	89.57%
Selección ranking-ruleta	27	SI	330	82.23%	96.65%	91.59%
Selección ranking-ruleta	28	SI	160	80.99%	95.76%	90.58%
Selección ranking-ruleta	29	SI	220	79.75%	96.21%	90.43%
Selección ranking-ruleta	30	SI	430	77.69%	96.43%	89.86%
Selección ranking-ruleta	31	SI	288	78.10%	96.21%	89.86%
Selección ranking-ruleta	32	SI	239	81.41%	96.65%	91.30%
Selección ranking-ruleta	33	SI	279	75.21%	95.98%	88.70%
Selección ranking-ruleta	34	SI	362	75.62%	96.43%	89.13%
Selección ranking-ruleta	35	SI	352	80.58%	97.10%	91.30%
Selección ranking-ruleta	36	SI	176	77.69%	95.76%	89.42%
Selección ranking-ruleta	37	SI	407	80.99%	97.10%	91.45%
Selección ranking-ruleta	38	SI	170	75.62%	95.31%	88.41%
Selección ranking-ruleta	39	SI	241	75.21%	96.43%	88.99%
Selección ranking-ruleta	40	SI	188	76.86%	95.98%	89.28%
Selección ranking-ruleta	41	SI	272	76.03%	96.88%	89.57%
Selección ranking-ruleta	42	SI	270	78.51%	96.88%	90.43%
Selección ranking-ruleta	43	SI	188	78.51%	95.54%	89.57%
Selección ranking-ruleta	44	SI	280	80.58%	95.98%	90.58%

<b>Función Selección</b>	<b>Corrida</b>	<b>Ent?</b>	<b>Tiempo Prom. Ent.</b>	<b>Aciertos Prom. Eval.</b>	<b>Aciertos Prom. Ent.</b>	<b>Aciertos Prom. Total</b>
Selección ranking-ruleta	45	SI	342	76.45%	95.54%	88.84%
Selección ranking-ruleta	46	SI	259	78.93%	96.88%	90.58%
Selección ranking-ruleta	47	SI	289	76.86%	96.21%	89.42%
Selección ranking-ruleta	48	SI	177	78.51%	95.31%	89.42%
Selección ranking-ruleta	49	SI	217	77.27%	96.65%	89.86%
Selección ranking-ruleta	50	SI	359	78.10%	96.65%	90.14%

#### **A.4. Función de cruza**

<b>Tipo Función Cruza</b>	<b>Corrida</b>	<b>Ent?</b>	<b>Tiempo Prom. Ent.</b>	<b>Aciertos Prom. Eval.</b>	<b>Aciertos Prom. Ent.</b>	<b>Aciertos Prom. Total</b>
Cruza probabilística	1	SI	170	76.86%	96.88%	89.86%
Cruza probabilística	2	SI	139	76.86%	95.76%	89.13%
Cruza probabilística	3	SI	238	78.51%	95.54%	89.57%
Cruza probabilística	4	SI	154	77.69%	95.76%	89.42%
Cruza probabilística	5	SI	180	76.45%	95.54%	88.84%
Cruza probabilística	6	NO	775	80.17%	97.32%	91.30%
Cruza probabilística	7	SI	195	77.69%	96.21%	89.71%
Cruza probabilística	8	SI	167	76.03%	93.75%	87.54%
Cruza probabilística	9	SI	281	79.75%	96.65%	90.72%
Cruza probabilística	10	SI	195	73.14%	93.53%	86.38%
Cruza probabilística	11	SI	171	76.45%	96.88%	89.71%
Cruza probabilística	12	SI	153	77.69%	95.54%	89.28%
Cruza probabilística	13	SI	256	80.17%	96.21%	90.58%
Cruza probabilística	14	SI	131	78.10%	95.76%	89.57%
Cruza probabilística	15	SI	142	78.51%	96.65%	90.29%
Cruza probabilística	16	SI	191	78.93%	95.54%	89.71%
Cruza probabilística	17	SI	183	77.27%	95.09%	88.84%

Tipo Función Cruza	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Cruza probabilística	18	SI	159	75.21%	95.54%	88.41%
Cruza probabilística	19	SI	139	77.27%	96.65%	89.86%
Cruza probabilística	20	SI	177	76.45%	95.98%	89.13%
Cruza probabilística	21	SI	199	78.10%	96.65%	90.14%
Cruza probabilística	22	SI	220	80.17%	96.88%	91.01%
Cruza probabilística	23	SI	167	77.69%	96.21%	89.71%
Cruza probabilística	24	SI	276	79.34%	96.43%	90.43%
Cruza probabilística	25	SI	346	79.34%	96.65%	90.58%
Cruza probabilística	26	SI	204	79.34%	95.54%	89.86%
Cruza probabilística	27	SI	124	78.10%	95.76%	89.57%
Cruza probabilística	28	SI	286	76.86%	96.21%	89.42%
Cruza probabilística	29	SI	151	79.75%	95.98%	90.29%
Cruza probabilística	30	SI	181	76.45%	95.76%	88.99%
Cruza probabilística	31	SI	147	79.75%	96.21%	90.43%
Cruza probabilística	32	SI	131	74.38%	95.76%	88.26%
Cruza probabilística	33	SI	163	79.34%	95.98%	90.14%
Cruza probabilística	34	SI	284	80.58%	96.88%	91.16%
Cruza probabilística	35	SI	335	80.58%	96.65%	91.01%
Cruza probabilística	36	SI	178	77.69%	95.09%	88.99%
Cruza probabilística	37	SI	148	78.93%	96.43%	90.29%
Cruza probabilística	38	SI	216	78.51%	95.54%	89.57%
Cruza probabilística	39	SI	308	78.93%	96.21%	90.14%
Cruza probabilística	40	SI	127	76.03%	94.87%	88.26%
Cruza probabilística	41	SI	134	74.79%	95.09%	87.97%
Cruza probabilística	42	SI	161	77.27%	96.21%	89.57%
Cruza probabilística	43	SI	198	77.69%	95.54%	89.28%
Cruza probabilística	44	SI	178	80.17%	95.76%	90.29%



Tipo Función Cruza	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Cruza probabilística	45	SI	143	77.27%	96.43%	89.71%
Cruza probabilística	46	SI	153	77.27%	96.21%	89.57%
Cruza probabilística	47	SI	250	78.93%	96.21%	90.14%
Cruza probabilística	48	SI	205	78.10%	96.65%	90.14%
Cruza probabilística	49	SI	159	78.10%	96.43%	90.00%
Cruza probabilística	50	SI	152	78.10%	96.43%	90.00%
Cruza uniforme	1	SI	157	76.86%	96.21%	89.42%
Cruza uniforme	2	SI	341	79.34%	97.32%	91.01%
Cruza uniforme	3	SI	167	80.17%	95.09%	89.86%
Cruza uniforme	4	SI	147	77.69%	95.54%	89.28%
Cruza uniforme	5	SI	246	78.10%	94.64%	88.84%
Cruza uniforme	6	SI	279	78.10%	95.76%	89.57%
Cruza uniforme	7	SI	149	75.21%	95.76%	88.55%
Cruza uniforme	8	SI	221	80.17%	96.65%	90.87%
Cruza uniforme	9	SI	212	78.93%	95.54%	89.71%
Cruza uniforme	10	SI	155	79.75%	96.43%	90.58%
Cruza uniforme	11	SI	141	76.03%	96.43%	89.28%
Cruza uniforme	12	SI	135	78.51%	96.21%	90.00%
Cruza uniforme	13	SI	214	77.27%	96.65%	89.86%
Cruza uniforme	14	SI	146	78.10%	95.76%	89.57%
Cruza uniforme	15	SI	162	73.55%	95.31%	87.68%
Cruza uniforme	16	SI	153	79.34%	96.21%	90.29%
Cruza uniforme	17	SI	176	79.34%	96.21%	90.29%
Cruza uniforme	18	SI	147	79.34%	96.43%	90.43%
Cruza uniforme	19	SI	217	75.21%	95.09%	88.12%
Cruza uniforme	20	SI	359	76.03%	95.54%	88.70%
Cruza uniforme	21	SI	136	78.10%	96.21%	89.86%

Tipo Función Cruza	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Cruza uniforme	22	SI	141	75.21%	95.09%	88.12%
Cruza uniforme	23	SI	165	78.93%	95.98%	90.00%
Cruza uniforme	24	SI	294	73.14%	96.21%	88.12%
Cruza uniforme	25	SI	225	77.27%	96.43%	89.71%
Cruza uniforme	26	SI	154	77.27%	96.21%	89.57%
Cruza uniforme	27	SI	231	75.62%	95.98%	88.84%
Cruza uniforme	28	SI	174	76.03%	95.54%	88.70%
Cruza uniforme	29	SI	173	79.75%	96.65%	90.72%
Cruza uniforme	30	SI	233	80.58%	95.98%	90.58%
Cruza uniforme	31	SI	183	76.45%	95.31%	88.70%
Cruza uniforme	32	SI	222	76.86%	94.20%	88.12%
Cruza uniforme	33	SI	174	76.03%	96.65%	89.42%
Cruza uniforme	34	SI	196	77.27%	96.65%	89.86%
Cruza uniforme	35	SI	203	80.17%	96.43%	90.72%
Cruza uniforme	36	SI	178	78.93%	96.43%	90.29%
Cruza uniforme	37	SI	193	78.93%	96.43%	90.29%
Cruza uniforme	38	SI	135	78.51%	95.76%	89.71%
Cruza uniforme	39	SI	171	77.69%	95.76%	89.42%
Cruza uniforme	40	SI	201	79.34%	96.43%	90.43%
Cruza uniforme	41	SI	148	78.93%	96.65%	90.43%
Cruza uniforme	42	SI	166	79.34%	96.21%	90.29%
Cruza uniforme	43	SI	192	76.03%	94.87%	88.26%
Cruza uniforme	44	SI	284	74.79%	96.65%	88.99%
Cruza uniforme	45	SI	182	79.75%	96.43%	90.58%
Cruza uniforme	46	SI	147	78.10%	96.65%	90.14%
Cruza uniforme	47	SI	193	77.69%	96.43%	89.86%
Cruza uniforme	48	SI	172	77.27%	96.65%	89.86%

Tipo Función Cruza	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Cruza uniforme	49	SI	157	78.51%	95.98%	89.86%
Cruza uniforme	50	SI	161	78.51%	95.31%	89.42%
Cruza simple	1	SI	394	77.27%	96.65%	89.86%
Cruza simple	2	SI	203	76.03%	95.09%	88.41%
Cruza simple	3	SI	184	80.58%	95.98%	90.58%
Cruza simple	4	SI	317	79.75%	97.10%	91.01%
Cruza simple	5	SI	602	78.51%	97.32%	90.72%
Cruza simple	6	SI	209	77.69%	95.76%	89.42%
Cruza simple	7	SI	182	76.45%	96.21%	89.28%
Cruza simple	8	SI	188	76.86%	96.65%	89.71%
Cruza simple	9	SI	168	76.03%	96.65%	89.42%
Cruza simple	10	SI	135	78.93%	96.88%	90.58%
Cruza simple	11	SI	234	77.69%	96.21%	89.71%
Cruza simple	12	SI	622	81.41%	97.32%	91.74%
Cruza simple	13	SI	259	78.10%	96.88%	90.29%
Cruza simple	14	SI	240	80.17%	96.43%	90.72%
Cruza simple	15	SI	197	78.10%	96.21%	89.86%
Cruza simple	16	SI	277	81.41%	95.98%	90.87%
Cruza simple	17	SI	148	78.93%	96.21%	90.14%
Cruza simple	18	SI	169	76.86%	96.43%	89.57%
Cruza simple	19	SI	283	79.75%	95.98%	90.29%
Cruza simple	20	SI	152	75.21%	95.76%	88.55%
Cruza simple	21	SI	225	77.27%	94.87%	88.70%
Cruza simple	22	SI	162	79.75%	95.76%	90.14%
Cruza simple	23	SI	177	79.75%	96.43%	90.58%
Cruza simple	24	SI	397	77.27%	95.54%	89.13%
Cruza simple	25	SI	204	78.93%	95.98%	90.00%

Tipo Función Cruza	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Cruza simple	26	SI	182	78.51%	96.21%	90.00%
Cruza simple	27	SI	270	79.34%	95.54%	89.86%
Cruza simple	28	SI	435	80.17%	96.65%	90.87%
Cruza simple	29	SI	169	78.51%	95.98%	89.86%
Cruza simple	30	SI	254	80.17%	95.76%	90.29%
Cruza simple	31	SI	260	78.93%	95.31%	89.57%
Cruza simple	32	SI	191	80.58%	96.21%	90.72%
Cruza simple	33	SI	243	77.69%	95.54%	89.28%
Cruza simple	34	SI	148	79.34%	95.76%	90.00%
Cruza simple	35	SI	189	76.45%	94.87%	88.41%
Cruza simple	36	SI	148	75.62%	95.54%	88.55%
Cruza simple	37	SI	217	76.86%	95.54%	88.99%
Cruza simple	38	SI	439	78.51%	97.10%	90.58%
Cruza simple	39	SI	357	80.58%	96.43%	90.87%
Cruza simple	40	SI	385	79.75%	97.10%	91.01%
Cruza simple	41	SI	171	78.10%	95.09%	89.13%
Cruza simple	42	SI	126	77.27%	93.97%	88.12%
Cruza simple	43	SI	206	78.10%	95.98%	89.71%
Cruza simple	44	SI	329	78.10%	95.76%	89.57%
Cruza simple	45	SI	264	78.10%	96.65%	90.14%
Cruza simple	46	SI	192	75.21%	95.31%	88.26%
Cruza simple	47	SI	150	78.93%	96.65%	90.43%
Cruza simple	48	SI	224	78.51%	95.76%	89.71%
Cruza simple	49	SI	403	79.75%	97.32%	91.16%
Cruza simple	50	SI	183	79.75%	95.98%	90.29%
Cruza multipuntos	1	SI	183	79.34%	96.21%	90.29%
Cruza multipuntos	2	SI	173	78.93%	96.21%	90.14%

Tipo Función Cruza	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Cruza multipuntos	3	SI	258	78.10%	96.21%	89.86%
Cruza multipuntos	4	SI	184	78.51%	95.31%	89.42%
Cruza multipuntos	5	SI	201	79.75%	94.87%	89.57%
Cruza multipuntos	6	SI	195	80.17%	95.09%	89.86%
Cruza multipuntos	7	SI	229	80.58%	96.65%	91.01%
Cruza multipuntos	8	SI	162	78.93%	95.98%	90.00%
Cruza multipuntos	9	SI	285	75.62%	96.65%	89.28%
Cruza multipuntos	10	SI	219	80.17%	96.43%	90.72%
Cruza multipuntos	11	SI	229	75.21%	96.43%	88.99%
Cruza multipuntos	12	SI	126	77.69%	95.98%	89.57%
Cruza multipuntos	13	SI	167	74.38%	94.64%	87.54%
Cruza multipuntos	14	SI	339	79.34%	96.65%	90.58%
Cruza multipuntos	15	SI	303	77.27%	96.43%	89.71%
Cruza multipuntos	16	SI	258	75.21%	95.31%	88.26%
Cruza multipuntos	17	SI	161	78.10%	95.98%	89.71%
Cruza multipuntos	18	SI	175	80.58%	96.21%	90.72%
Cruza multipuntos	19	SI	252	77.27%	96.21%	89.57%
Cruza multipuntos	20	SI	168	79.34%	96.21%	90.29%
Cruza multipuntos	21	SI	259	78.93%	97.10%	90.72%
Cruza multipuntos	22	SI	175	76.86%	95.54%	88.99%
Cruza multipuntos	23	SI	290	78.10%	95.31%	89.28%
Cruza multipuntos	24	SI	144	75.62%	95.76%	88.70%
Cruza multipuntos	25	SI	210	76.86%	96.21%	89.42%
Cruza multipuntos	26	SI	228	78.10%	95.54%	89.42%
Cruza multipuntos	27	SI	231	78.10%	96.88%	90.29%
Cruza multipuntos	28	SI	177	79.34%	95.76%	90.00%
Cruza multipuntos	29	SI	256	79.75%	96.43%	90.58%

Tipo Función Cruza	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Cruza multipuntos	30	SI	347	77.27%	95.76%	89.28%
Cruza multipuntos	31	SI	430	79.75%	97.10%	91.01%
Cruza multipuntos	32	SI	162	76.86%	94.87%	88.55%
Cruza multipuntos	33	SI	275	76.86%	96.65%	89.71%
Cruza multipuntos	34	SI	138	77.69%	95.76%	89.42%
Cruza multipuntos	35	SI	238	77.69%	95.76%	89.42%
Cruza multipuntos	36	SI	157	78.51%	96.88%	90.43%
Cruza multipuntos	37	SI	168	77.27%	95.54%	89.13%
Cruza multipuntos	38	SI	189	78.93%	96.43%	90.29%
Cruza multipuntos	39	SI	229	78.93%	95.76%	89.86%
Cruza multipuntos	40	SI	289	77.69%	97.10%	90.29%
Cruza multipuntos	41	SI	134	78.93%	96.65%	90.43%
Cruza multipuntos	42	SI	133	76.86%	95.54%	88.99%
Cruza multipuntos	43	SI	186	80.99%	96.88%	91.30%
Cruza multipuntos	44	SI	378	78.51%	96.43%	90.14%
Cruza multipuntos	45	SI	201	80.17%	96.88%	91.01%
Cruza multipuntos	46	SI	176	77.69%	95.76%	89.42%
Cruza multipuntos	47	SI	237	76.45%	97.10%	89.86%
Cruza multipuntos	48	SI	144	76.45%	95.54%	88.84%
Cruza multipuntos	49	SI	141	76.86%	95.76%	89.13%
Cruza multipuntos	50	SI	188	78.51%	95.76%	89.71%

**A.5. Función de mutación****Mutación numérica**

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación uniforme	0.01	0.01	1	SI	174	80.17%	95.98%	90.43%
Mutación uniforme	0.01	0.01	2	SI	213	77.69%	95.54%	89.28%
Mutación uniforme	0.01	0.01	3	SI	236	79.34%	95.98%	90.14%
Mutación uniforme	0.01	0.01	4	SI	225	77.69%	95.76%	89.42%
Mutación uniforme	0.01	0.01	5	SI	243	78.93%	94.20%	88.84%
Mutación uniforme	0.01	0.01	6	SI	333	80.17%	96.65%	90.87%
Mutación uniforme	0.01	0.01	7	SI	194	79.34%	95.98%	90.14%
Mutación uniforme	0.01	0.01	8	SI	200	75.62%	95.31%	88.41%
Mutación uniforme	0.01	0.01	9	SI	136	77.69%	95.98%	89.57%
Mutación uniforme	0.01	0.01	10	SI	170	80.17%	96.65%	90.87%
Mutación uniforme	0.01	0.01	11	SI	334	80.17%	96.65%	90.87%
Mutación uniforme	0.01	0.01	12	SI	188	78.93%	96.43%	90.29%
Mutación uniforme	0.01	0.01	13	SI	193	79.34%	95.98%	90.14%
Mutación uniforme	0.01	0.01	14	SI	234	75.62%	95.09%	88.26%
Mutación uniforme	0.01	0.01	15	SI	266	76.86%	95.76%	89.13%
Mutación uniforme	0.01	0.01	16	SI	186	79.34%	95.54%	89.86%
Mutación uniforme	0.01	0.01	17	SI	204	75.62%	95.76%	88.70%
Mutación uniforme	0.01	0.01	18	SI	179	78.10%	95.54%	89.42%
Mutación uniforme	0.01	0.01	19	SI	173	79.75%	95.54%	90.00%
Mutación uniforme	0.01	0.01	20	SI	177	79.75%	96.43%	90.58%
Mutación uniforme	0.01	0.01	21	SI	265	78.10%	96.65%	90.14%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación uniforme	0.01	0.01	22	SI	162	76.03%	95.76%	88.84%
Mutación uniforme	0.01	0.01	23	SI	190	80.17%	95.76%	90.29%
Mutación uniforme	0.01	0.01	24	SI	209	77.69%	95.54%	89.28%
Mutación uniforme	0.01	0.01	25	SI	224	78.51%	96.21%	90.00%
Mutación uniforme	0.01	0.01	26	SI	189	77.27%	95.54%	89.13%
Mutación uniforme	0.01	0.01	27	SI	450	72.73%	93.97%	86.52%
Mutación uniforme	0.01	0.01	28	SI	216	74.79%	94.64%	87.68%
Mutación uniforme	0.01	0.01	29	SI	186	77.27%	94.87%	88.70%
Mutación uniforme	0.01	0.01	30	SI	175	76.03%	94.64%	88.12%
Mutación uniforme	0.01	0.01	31	SI	151	78.10%	95.76%	89.57%
Mutación uniforme	0.01	0.01	32	SI	189	76.45%	96.21%	89.28%
Mutación uniforme	0.01	0.01	33	SI	182	76.45%	95.09%	88.55%
Mutación uniforme	0.01	0.01	34	SI	193	78.93%	96.43%	90.29%
Mutación uniforme	0.01	0.01	35	SI	181	75.21%	95.54%	88.41%
Mutación uniforme	0.01	0.01	36	SI	141	76.86%	94.87%	88.55%
Mutación uniforme	0.01	0.01	37	SI	200	76.86%	96.21%	89.42%
Mutación uniforme	0.01	0.01	38	SI	251	79.34%	96.43%	90.43%
Mutación uniforme	0.01	0.01	39	SI	211	78.51%	95.76%	89.71%
Mutación uniforme	0.01	0.01	40	SI	247	77.69%	94.87%	88.84%
Mutación uniforme	0.01	0.01	41	SI	245	73.97%	94.20%	87.10%
Mutación uniforme	0.01	0.01	42	SI	244	77.27%	94.64%	88.55%
Mutación uniforme	0.01	0.01	43	SI	211	75.62%	96.65%	89.28%
Mutación uniforme	0.01	0.01	44	SI	157	76.86%	95.09%	88.70%
Mutación uniforme	0.01	0.01	45	SI	260	79.34%	96.88%	90.72%
Mutación uniforme	0.01	0.01	46	SI	203	76.45%	96.21%	89.28%
Mutación uniforme	0.01	0.01	47	SI	255	79.34%	96.21%	90.29%



Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación uniforme	0.01	0.01	48	SI	242	79.34%	96.88%	90.72%
Mutación uniforme	0.01	0.01	49	SI	178	78.51%	95.98%	89.86%
Mutación uniforme	0.01	0.01	50	SI	238	78.93%	96.43%	90.29%
Mutación uniforme	0.02	0.02	1	SI	236	79.34%	95.76%	90.00%
Mutación uniforme	0.02	0.02	2	SI	149	76.86%	95.09%	88.70%
Mutación uniforme	0.02	0.02	3	SI	143	77.27%	95.09%	88.84%
Mutación uniforme	0.02	0.02	4	SI	204	78.51%	95.54%	89.57%
Mutación uniforme	0.02	0.02	5	SI	177	78.93%	96.88%	90.58%
Mutación uniforme	0.02	0.02	6	SI	235	76.03%	94.42%	87.97%
Mutación uniforme	0.02	0.02	7	SI	240	75.62%	96.43%	89.13%
Mutación uniforme	0.02	0.02	8	SI	266	78.93%	96.88%	90.58%
Mutación uniforme	0.02	0.02	9	SI	152	78.93%	96.21%	90.14%
Mutación uniforme	0.02	0.02	10	SI	147	77.27%	96.21%	89.57%
Mutación uniforme	0.02	0.02	11	SI	193	78.93%	97.10%	90.72%
Mutación uniforme	0.02	0.02	12	SI	241	79.34%	94.87%	89.42%
Mutación uniforme	0.02	0.02	13	SI	198	76.45%	95.54%	88.84%
Mutación uniforme	0.02	0.02	14	SI	159	80.17%	95.31%	90.00%
Mutación uniforme	0.02	0.02	15	SI	336	78.93%	95.76%	89.86%
Mutación uniforme	0.02	0.02	16	SI	204	78.10%	95.98%	89.71%
Mutación uniforme	0.02	0.02	17	SI	209	77.69%	96.43%	89.86%
Mutación uniforme	0.02	0.02	18	SI	162	79.34%	96.43%	90.43%
Mutación uniforme	0.02	0.02	19	SI	250	78.51%	96.65%	90.29%
Mutación uniforme	0.02	0.02	20	SI	184	78.93%	94.87%	89.28%
Mutación uniforme	0.02	0.02	21	SI	175	77.69%	96.21%	89.71%
Mutación uniforme	0.02	0.02	22	SI	157	76.45%	95.09%	88.55%
Mutación uniforme	0.02	0.02	23	SI	202	76.03%	95.31%	88.55%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación uniforme	0.02	0.02	24	SI	157	78.10%	95.98%	89.71%
Mutación uniforme	0.02	0.02	25	SI	188	78.93%	96.21%	90.14%
Mutación uniforme	0.02	0.02	26	SI	161	78.93%	95.54%	89.71%
Mutación uniforme	0.02	0.02	27	SI	179	76.03%	95.09%	88.41%
Mutación uniforme	0.02	0.02	28	SI	217	79.75%	96.43%	90.58%
Mutación uniforme	0.02	0.02	29	SI	171	77.69%	96.21%	89.71%
Mutación uniforme	0.02	0.02	30	SI	166	79.75%	96.65%	90.72%
Mutación uniforme	0.02	0.02	31	SI	203	78.10%	96.65%	90.14%
Mutación uniforme	0.02	0.02	32	SI	235	78.93%	96.21%	90.14%
Mutación uniforme	0.02	0.02	33	SI	216	80.17%	95.76%	90.29%
Mutación uniforme	0.02	0.02	34	SI	137	77.27%	95.98%	89.42%
Mutación uniforme	0.02	0.02	35	SI	153	79.75%	96.21%	90.43%
Mutación uniforme	0.02	0.02	36	SI	183	77.27%	94.87%	88.70%
Mutación uniforme	0.02	0.02	37	SI	160	77.69%	95.09%	88.99%
Mutación uniforme	0.02	0.02	38	SI	308	80.17%	97.10%	91.16%
Mutación uniforme	0.02	0.02	39	SI	158	80.58%	96.21%	90.72%
Mutación uniforme	0.02	0.02	40	SI	221	80.58%	95.76%	90.43%
Mutación uniforme	0.02	0.02	41	SI	436	80.58%	96.43%	90.87%
Mutación uniforme	0.02	0.02	42	SI	144	77.27%	96.43%	89.71%
Mutación uniforme	0.02	0.02	43	SI	284	80.58%	96.21%	90.72%
Mutación uniforme	0.02	0.02	44	SI	260	80.99%	96.21%	90.87%
Mutación uniforme	0.02	0.02	45	SI	184	78.10%	95.31%	89.28%
Mutación uniforme	0.02	0.02	46	SI	235	78.10%	96.88%	90.29%
Mutación uniforme	0.02	0.02	47	SI	136	75.62%	96.21%	88.99%
Mutación uniforme	0.02	0.02	48	SI	187	77.69%	96.43%	89.86%
Mutación uniforme	0.02	0.02	49	SI	233	79.34%	96.21%	90.29%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación uniforme	0.02	0.02	50	SI	224	74.38%	97.10%	89.13%
Mutación uniforme	0.04	0.04	1	SI	196	80.58%	96.43%	90.87%
Mutación uniforme	0.04	0.04	2	SI	255	78.51%	96.88%	90.43%
Mutación uniforme	0.04	0.04	3	SI	161	79.34%	95.98%	90.14%
Mutación uniforme	0.04	0.04	4	SI	186	81.82%	96.65%	91.45%
Mutación uniforme	0.04	0.04	5	SI	287	79.75%	96.65%	90.72%
Mutación uniforme	0.04	0.04	6	SI	170	74.79%	95.98%	88.55%
Mutación uniforme	0.04	0.04	7	SI	500	78.93%	97.32%	90.87%
Mutación uniforme	0.04	0.04	8	SI	768	81.82%	97.32%	91.88%
Mutación uniforme	0.04	0.04	9	SI	247	83.06%	96.65%	91.88%
Mutación uniforme	0.04	0.04	10	SI	145	80.17%	97.10%	91.16%
Mutación uniforme	0.04	0.04	11	SI	165	80.99%	96.43%	91.01%
Mutación uniforme	0.04	0.04	12	SI	293	78.10%	95.98%	89.71%
Mutación uniforme	0.04	0.04	13	SI	138	79.34%	96.65%	90.58%
Mutación uniforme	0.04	0.04	14	SI	166	80.58%	95.98%	90.58%
Mutación uniforme	0.04	0.04	15	SI	299	77.27%	96.43%	89.71%
Mutación uniforme	0.04	0.04	16	SI	161	81.82%	96.43%	91.30%
Mutación uniforme	0.04	0.04	17	SI	151	77.69%	95.76%	89.42%
Mutación uniforme	0.04	0.04	18	SI	168	79.34%	96.21%	90.29%
Mutación uniforme	0.04	0.04	19	SI	141	80.58%	97.10%	91.30%
Mutación uniforme	0.04	0.04	20	SI	166	78.10%	96.65%	90.14%
Mutación uniforme	0.04	0.04	21	SI	399	83.47%	97.32%	92.46%
Mutación uniforme	0.04	0.04	22	SI	183	78.10%	96.21%	89.86%
Mutación uniforme	0.04	0.04	23	SI	643	78.51%	97.32%	90.72%
Mutación uniforme	0.04	0.04	24	SI	135	79.75%	96.43%	90.58%
Mutación uniforme	0.04	0.04	25	SI	224	80.58%	97.10%	91.30%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación uniforme	0.04	0.04	26	SI	148	78.51%	95.98%	89.86%
Mutación uniforme	0.04	0.04	27	SI	226	76.86%	96.88%	89.86%
Mutación uniforme	0.04	0.04	28	SI	163	80.58%	96.21%	90.72%
Mutación uniforme	0.04	0.04	29	SI	222	81.41%	96.65%	91.30%
Mutación uniforme	0.04	0.04	30	SI	208	79.34%	96.43%	90.43%
Mutación uniforme	0.04	0.04	31	SI	280	78.51%	96.43%	90.14%
Mutación uniforme	0.04	0.04	32	SI	378	78.93%	96.43%	90.29%
Mutación uniforme	0.04	0.04	33	SI	290	80.58%	96.88%	91.16%
Mutación uniforme	0.04	0.04	34	SI	228	81.41%	96.65%	91.30%
Mutación uniforme	0.04	0.04	35	SI	580	80.58%	96.88%	91.16%
Mutación uniforme	0.04	0.04	36	SI	459	78.93%	96.43%	90.29%
Mutación uniforme	0.04	0.04	37	SI	182	78.93%	96.88%	90.58%
Mutación uniforme	0.04	0.04	38	SI	149	76.86%	96.43%	89.57%
Mutación uniforme	0.04	0.04	39	SI	125	77.69%	96.21%	89.71%
Mutación uniforme	0.04	0.04	40	NO	773	81.82%	96.65%	91.45%
Mutación uniforme	0.04	0.04	41	SI	169	77.69%	96.65%	90.00%
Mutación uniforme	0.04	0.04	42	SI	154	80.17%	96.65%	90.87%
Mutación uniforme	0.04	0.04	43	SI	231	78.93%	97.10%	90.72%
Mutación uniforme	0.04	0.04	44	SI	150	78.10%	95.54%	89.42%
Mutación uniforme	0.04	0.04	45	SI	590	78.10%	97.10%	90.43%
Mutación uniforme	0.04	0.04	46	SI	239	79.34%	96.43%	90.43%
Mutación uniforme	0.04	0.04	47	SI	263	82.64%	96.88%	91.88%
Mutación uniforme	0.04	0.04	48	SI	133	78.51%	95.31%	89.42%
Mutación uniforme	0.04	0.04	49	SI	545	80.58%	96.65%	91.01%
Mutación uniforme	0.04	0.04	50	SI	139	79.75%	96.65%	90.72%
Mutación lineal creciente	0.01	0.02	1	SI	154	81.41%	95.98%	90.87%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal creciente	0.01	0.02	2	SI	213	77.27%	95.54%	89.13%
Mutación lineal creciente	0.01	0.02	3	SI	208	77.69%	94.87%	88.84%
Mutación lineal creciente	0.01	0.02	4	SI	146	76.03%	95.54%	88.70%
Mutación lineal creciente	0.01	0.02	5	SI	238	74.38%	94.42%	87.39%
Mutación lineal creciente	0.01	0.02	6	SI	229	78.51%	95.76%	89.71%
Mutación lineal creciente	0.01	0.02	7	SI	184	81.41%	96.21%	91.01%
Mutación lineal creciente	0.01	0.02	8	SI	144	78.51%	95.98%	89.86%
Mutación lineal creciente	0.01	0.02	9	SI	200	76.86%	95.09%	88.70%
Mutación lineal creciente	0.01	0.02	10	SI	178	78.93%	95.54%	89.71%
Mutación lineal creciente	0.01	0.02	11	SI	284	79.75%	95.54%	90.00%
Mutación lineal creciente	0.01	0.02	12	SI	222	78.51%	96.43%	90.14%
Mutación lineal creciente	0.01	0.02	13	SI	146	78.93%	96.21%	90.14%
Mutación lineal creciente	0.01	0.02	14	SI	144	76.03%	95.31%	88.55%
Mutación lineal creciente	0.01	0.02	15	SI	171	75.62%	95.31%	88.41%
Mutación lineal creciente	0.01	0.02	16	SI	199	78.10%	95.98%	89.71%
Mutación lineal creciente	0.01	0.02	17	SI	188	75.62%	95.98%	88.84%
Mutación lineal creciente	0.01	0.02	18	SI	157	75.62%	95.09%	88.26%
Mutación lineal creciente	0.01	0.02	19	SI	142	78.93%	94.87%	89.28%
Mutación lineal creciente	0.01	0.02	20	SI	148	79.75%	96.43%	90.58%
Mutación lineal creciente	0.01	0.02	21	SI	164	76.45%	95.54%	88.84%
Mutación lineal creciente	0.01	0.02	22	SI	242	78.93%	96.21%	90.14%
Mutación lineal creciente	0.01	0.02	23	SI	147	79.34%	95.76%	90.00%
Mutación lineal creciente	0.01	0.02	24	SI	198	76.03%	95.31%	88.55%
Mutación lineal creciente	0.01	0.02	25	SI	197	75.62%	95.76%	88.70%
Mutación lineal creciente	0.01	0.02	26	SI	216	78.93%	95.54%	89.71%
Mutación lineal creciente	0.01	0.02	27	SI	218	77.27%	95.31%	88.99%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal creciente	0.01	0.02	28	SI	172	77.69%	96.21%	89.71%
Mutación lineal creciente	0.01	0.02	29	SI	197	79.34%	95.31%	89.71%
Mutación lineal creciente	0.01	0.02	30	SI	166	81.41%	95.98%	90.87%
Mutación lineal creciente	0.01	0.02	31	SI	173	77.27%	96.21%	89.57%
Mutación lineal creciente	0.01	0.02	32	SI	171	77.27%	96.21%	89.57%
Mutación lineal creciente	0.01	0.02	33	SI	153	77.27%	95.76%	89.28%
Mutación lineal creciente	0.01	0.02	34	SI	235	76.86%	95.98%	89.28%
Mutación lineal creciente	0.01	0.02	35	SI	376	79.75%	97.10%	91.01%
Mutación lineal creciente	0.01	0.02	36	SI	195	77.27%	96.43%	89.71%
Mutación lineal creciente	0.01	0.02	37	SI	147	79.34%	95.54%	89.86%
Mutación lineal creciente	0.01	0.02	38	SI	337	78.51%	97.10%	90.58%
Mutación lineal creciente	0.01	0.02	39	SI	281	80.17%	97.10%	91.16%
Mutación lineal creciente	0.01	0.02	40	SI	156	73.55%	93.75%	86.67%
Mutación lineal creciente	0.01	0.02	41	SI	162	77.69%	94.87%	88.84%
Mutación lineal creciente	0.01	0.02	42	SI	224	78.93%	95.09%	89.42%
Mutación lineal creciente	0.01	0.02	43	SI	169	76.86%	95.54%	88.99%
Mutación lineal creciente	0.01	0.02	44	SI	161	79.34%	96.43%	90.43%
Mutación lineal creciente	0.01	0.02	45	SI	194	81.41%	96.21%	91.01%
Mutación lineal creciente	0.01	0.02	46	SI	196	79.34%	95.98%	90.14%
Mutación lineal creciente	0.01	0.02	47	SI	223	77.69%	96.65%	90.00%
Mutación lineal creciente	0.01	0.02	48	SI	204	78.51%	96.21%	90.00%
Mutación lineal creciente	0.01	0.02	49	SI	169	78.51%	96.65%	90.29%
Mutación lineal creciente	0.01	0.02	50	SI	236	78.93%	96.65%	90.43%
Mutación lineal creciente	0.02	0.04	1	SI	193	76.03%	95.31%	88.55%
Mutación lineal creciente	0.02	0.04	2	SI	143	77.69%	94.87%	88.84%
Mutación lineal creciente	0.02	0.04	3	SI	165	80.58%	96.21%	90.72%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal creciente	0.02	0.04	4	SI	154	77.69%	96.65%	90.00%
Mutación lineal creciente	0.02	0.04	5	SI	184	77.27%	96.65%	89.86%
Mutación lineal creciente	0.02	0.04	6	SI	152	78.93%	96.88%	90.58%
Mutación lineal creciente	0.02	0.04	7	SI	143	79.34%	96.43%	90.43%
Mutación lineal creciente	0.02	0.04	8	SI	148	77.27%	95.54%	89.13%
Mutación lineal creciente	0.02	0.04	9	SI	141	77.69%	95.98%	89.57%
Mutación lineal creciente	0.02	0.04	10	SI	161	79.34%	96.65%	90.58%
Mutación lineal creciente	0.02	0.04	11	SI	185	80.99%	95.98%	90.72%
Mutación lineal creciente	0.02	0.04	12	SI	132	75.62%	96.21%	88.99%
Mutación lineal creciente	0.02	0.04	13	SI	184	78.51%	97.10%	90.58%
Mutación lineal creciente	0.02	0.04	14	SI	155	77.27%	96.65%	89.86%
Mutación lineal creciente	0.02	0.04	15	SI	211	80.17%	96.21%	90.58%
Mutación lineal creciente	0.02	0.04	16	SI	171	80.17%	96.65%	90.87%
Mutación lineal creciente	0.02	0.04	17	SI	180	77.69%	96.21%	89.71%
Mutación lineal creciente	0.02	0.04	18	SI	266	79.75%	96.43%	90.58%
Mutación lineal creciente	0.02	0.04	19	SI	391	77.27%	96.43%	89.71%
Mutación lineal creciente	0.02	0.04	20	SI	331	79.75%	97.54%	91.30%
Mutación lineal creciente	0.02	0.04	21	SI	235	80.99%	96.21%	90.87%
Mutación lineal creciente	0.02	0.04	22	SI	308	79.75%	97.10%	91.01%
Mutación lineal creciente	0.02	0.04	23	SI	157	78.51%	96.65%	90.29%
Mutación lineal creciente	0.02	0.04	24	SI	172	74.79%	95.31%	88.12%
Mutación lineal creciente	0.02	0.04	25	SI	164	78.10%	96.43%	90.00%
Mutación lineal creciente	0.02	0.04	26	SI	391	79.75%	96.88%	90.87%
Mutación lineal creciente	0.02	0.04	27	SI	240	79.75%	96.43%	90.58%
Mutación lineal creciente	0.02	0.04	28	SI	224	76.03%	96.65%	89.42%
Mutación lineal creciente	0.02	0.04	29	SI	197	76.45%	94.42%	88.12%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal creciente	0.02	0.04	30	SI	161	79.75%	96.88%	90.87%
Mutación lineal creciente	0.02	0.04	31	SI	147	76.86%	96.43%	89.57%
Mutación lineal creciente	0.02	0.04	32	SI	146	78.93%	96.21%	90.14%
Mutación lineal creciente	0.02	0.04	33	SI	157	80.17%	95.98%	90.43%
Mutación lineal creciente	0.02	0.04	34	SI	174	79.75%	96.21%	90.43%
Mutación lineal creciente	0.02	0.04	35	SI	207	78.10%	95.98%	89.71%
Mutación lineal creciente	0.02	0.04	36	SI	161	76.86%	95.31%	88.84%
Mutación lineal creciente	0.02	0.04	37	SI	169	77.27%	94.87%	88.70%
Mutación lineal creciente	0.02	0.04	38	SI	187	79.34%	96.43%	90.43%
Mutación lineal creciente	0.02	0.04	39	SI	191	77.27%	95.98%	89.42%
Mutación lineal creciente	0.02	0.04	40	SI	232	78.10%	96.21%	89.86%
Mutación lineal creciente	0.02	0.04	41	SI	165	80.17%	96.65%	90.87%
Mutación lineal creciente	0.02	0.04	42	SI	158	80.58%	96.43%	90.87%
Mutación lineal creciente	0.02	0.04	43	SI	206	80.99%	96.21%	90.87%
Mutación lineal creciente	0.02	0.04	44	SI	184	78.10%	96.21%	89.86%
Mutación lineal creciente	0.02	0.04	45	SI	137	79.75%	96.43%	90.58%
Mutación lineal creciente	0.02	0.04	46	SI	149	78.51%	95.31%	89.42%
Mutación lineal creciente	0.02	0.04	47	SI	202	76.03%	96.43%	89.28%
Mutación lineal creciente	0.02	0.04	48	SI	157	77.69%	96.43%	89.86%
Mutación lineal creciente	0.02	0.04	49	SI	159	78.51%	95.54%	89.57%
Mutación lineal creciente	0.02	0.04	50	SI	208	80.17%	96.88%	91.01%
Mutación lineal creciente	0.04	0.08	1	NO	776	79.75%	97.10%	91.01%
Mutación lineal creciente	0.04	0.08	2	SI	164	81.82%	96.65%	91.45%
Mutación lineal creciente	0.04	0.08	3	SI	213	80.58%	96.88%	91.16%
Mutación lineal creciente	0.04	0.08	4	SI	254	79.75%	96.65%	90.72%
Mutación lineal creciente	0.04	0.08	5	SI	334	80.99%	97.10%	91.45%



Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal creciente	0.04	0.08	6	SI	155	77.69%	96.43%	89.86%
Mutación lineal creciente	0.04	0.08	7	SI	615	79.75%	97.32%	91.16%
Mutación lineal creciente	0.04	0.08	8	SI	210	79.75%	95.31%	89.86%
Mutación lineal creciente	0.04	0.08	9	SI	234	77.69%	97.32%	90.43%
Mutación lineal creciente	0.04	0.08	10	SI	182	76.86%	95.76%	89.13%
Mutación lineal creciente	0.04	0.08	11	SI	286	80.17%	96.65%	90.87%
Mutación lineal creciente	0.04	0.08	12	SI	306	83.47%	97.10%	92.32%
Mutación lineal creciente	0.04	0.08	13	SI	225	76.86%	95.98%	89.28%
Mutación lineal creciente	0.04	0.08	14	SI	158	78.93%	96.88%	90.58%
Mutación lineal creciente	0.04	0.08	15	SI	467	82.23%	96.88%	91.74%
Mutación lineal creciente	0.04	0.08	16	SI	469	78.10%	97.32%	90.58%
Mutación lineal creciente	0.04	0.08	17	SI	601	80.58%	97.10%	91.30%
Mutación lineal creciente	0.04	0.08	18	NO	775	80.99%	96.88%	91.30%
Mutación lineal creciente	0.04	0.08	19	SI	225	80.58%	94.87%	89.86%
Mutación lineal creciente	0.04	0.08	20	SI	365	79.34%	96.88%	90.72%
Mutación lineal creciente	0.04	0.08	21	SI	194	80.17%	96.65%	90.87%
Mutación lineal creciente	0.04	0.08	22	SI	270	75.21%	96.21%	88.84%
Mutación lineal creciente	0.04	0.08	23	SI	492	80.17%	96.88%	91.01%
Mutación lineal creciente	0.04	0.08	24	SI	311	76.45%	96.43%	89.42%
Mutación lineal creciente	0.04	0.08	25	SI	178	81.82%	96.88%	91.59%
Mutación lineal creciente	0.04	0.08	26	SI	370	81.41%	96.88%	91.45%
Mutación lineal creciente	0.04	0.08	27	SI	192	78.93%	96.88%	90.58%
Mutación lineal creciente	0.04	0.08	28	SI	311	79.75%	95.98%	90.29%
Mutación lineal creciente	0.04	0.08	29	SI	336	76.45%	95.98%	89.13%
Mutación lineal creciente	0.04	0.08	30	SI	155	80.58%	96.88%	91.16%
Mutación lineal creciente	0.04	0.08	31	SI	224	79.75%	96.88%	90.87%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal creciente	0.04	0.08	32	SI	170	80.17%	95.98%	90.43%
Mutación lineal creciente	0.04	0.08	33	SI	207	79.34%	96.65%	90.58%
Mutación lineal creciente	0.04	0.08	34	SI	702	81.82%	97.10%	91.74%
Mutación lineal creciente	0.04	0.08	35	SI	408	77.69%	96.65%	90.00%
Mutación lineal creciente	0.04	0.08	36	SI	226	78.93%	96.43%	90.29%
Mutación lineal creciente	0.04	0.08	37	SI	149	78.10%	95.54%	89.42%
Mutación lineal creciente	0.04	0.08	38	NO	775	79.34%	96.65%	90.58%
Mutación lineal creciente	0.04	0.08	39	SI	282	75.62%	95.76%	88.70%
Mutación lineal creciente	0.04	0.08	40	SI	164	78.51%	96.65%	90.29%
Mutación lineal creciente	0.04	0.08	41	SI	735	76.45%	97.32%	90.00%
Mutación lineal creciente	0.04	0.08	42	SI	127	77.27%	96.65%	89.86%
Mutación lineal creciente	0.04	0.08	43	SI	166	78.51%	96.21%	90.00%
Mutación lineal creciente	0.04	0.08	44	SI	196	78.93%	97.10%	90.72%
Mutación lineal creciente	0.04	0.08	45	SI	214	76.86%	95.54%	88.99%
Mutación lineal creciente	0.04	0.08	46	SI	341	80.99%	96.43%	91.01%
Mutación lineal creciente	0.04	0.08	47	SI	290	80.99%	96.43%	91.01%
Mutación lineal creciente	0.04	0.08	48	SI	353	75.21%	96.43%	88.99%
Mutación lineal creciente	0.04	0.08	49	SI	206	78.93%	96.43%	90.29%
Mutación lineal creciente	0.04	0.08	50	SI	533	80.58%	97.32%	91.45%
Mutación lineal decreciente	0.005	0.01	1	SI	236	78.10%	95.98%	89.71%
Mutación lineal decreciente	0.005	0.01	2	SI	230	78.93%	95.31%	89.57%
Mutación lineal decreciente	0.005	0.01	3	SI	200	76.45%	95.76%	88.99%
Mutación lineal decreciente	0.005	0.01	4	SI	154	76.03%	95.09%	88.41%
Mutación lineal decreciente	0.005	0.01	5	SI	274	79.34%	95.31%	89.71%
Mutación lineal decreciente	0.005	0.01	6	SI	189	74.79%	94.64%	87.68%
Mutación lineal decreciente	0.005	0.01	7	SI	227	78.10%	95.09%	89.13%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal decreciente	0.005	0.01	8	SI	221	80.17%	96.21%	90.58%
Mutación lineal decreciente	0.005	0.01	9	SI	198	77.69%	95.98%	89.57%
Mutación lineal decreciente	0.005	0.01	10	SI	182	76.86%	94.64%	88.41%
Mutación lineal decreciente	0.005	0.01	11	SI	192	77.27%	95.76%	89.28%
Mutación lineal decreciente	0.005	0.01	12	SI	273	76.86%	95.98%	89.28%
Mutación lineal decreciente	0.005	0.01	13	SI	201	76.03%	94.87%	88.26%
Mutación lineal decreciente	0.005	0.01	14	NO	775	78.10%	96.21%	89.86%
Mutación lineal decreciente	0.005	0.01	15	SI	228	77.69%	95.76%	89.42%
Mutación lineal decreciente	0.005	0.01	16	NO	774	78.10%	95.31%	89.28%
Mutación lineal decreciente	0.005	0.01	17	NO	775	78.10%	95.98%	89.71%
Mutación lineal decreciente	0.005	0.01	18	SI	305	77.69%	95.54%	89.28%
Mutación lineal decreciente	0.005	0.01	19	SI	204	77.27%	95.98%	89.42%
Mutación lineal decreciente	0.005	0.01	20	NO	775	78.51%	96.88%	90.43%
Mutación lineal decreciente	0.005	0.01	21	SI	224	74.79%	96.21%	88.70%
Mutación lineal decreciente	0.005	0.01	22	SI	236	79.34%	96.43%	90.43%
Mutación lineal decreciente	0.005	0.01	23	SI	276	78.93%	96.43%	90.29%
Mutación lineal decreciente	0.005	0.01	24	SI	214	78.10%	95.31%	89.28%
Mutación lineal decreciente	0.005	0.01	25	SI	178	76.86%	95.54%	88.99%
Mutación lineal decreciente	0.005	0.01	26	SI	227	78.51%	95.76%	89.71%
Mutación lineal decreciente	0.005	0.01	27	SI	185	77.69%	95.76%	89.42%
Mutación lineal decreciente	0.005	0.01	28	NO	773	78.10%	96.65%	90.14%
Mutación lineal decreciente	0.005	0.01	29	SI	191	74.79%	94.64%	87.68%
Mutación lineal decreciente	0.005	0.01	30	SI	157	78.51%	95.76%	89.71%
Mutación lineal decreciente	0.005	0.01	31	SI	244	76.86%	95.54%	88.99%
Mutación lineal decreciente	0.005	0.01	32	SI	247	79.34%	95.54%	89.86%
Mutación lineal decreciente	0.005	0.01	33	NO	774	80.17%	96.65%	90.87%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal decreciente	0.005	0.01	34	SI	260	77.69%	95.98%	89.57%
Mutación lineal decreciente	0.005	0.01	35	SI	164	77.27%	96.21%	89.57%
Mutación lineal decreciente	0.005	0.01	36	SI	219	77.27%	96.65%	89.86%
Mutación lineal decreciente	0.005	0.01	37	SI	296	76.86%	95.54%	88.99%
Mutación lineal decreciente	0.005	0.01	38	SI	172	80.99%	95.31%	90.29%
Mutación lineal decreciente	0.005	0.01	39	SI	217	78.10%	95.76%	89.57%
Mutación lineal decreciente	0.005	0.01	40	SI	173	78.51%	96.65%	90.29%
Mutación lineal decreciente	0.005	0.01	41	NO	775	80.17%	95.09%	89.86%
Mutación lineal decreciente	0.005	0.01	42	SI	183	78.10%	95.31%	89.28%
Mutación lineal decreciente	0.005	0.01	43	SI	218	76.86%	95.54%	88.99%
Mutación lineal decreciente	0.005	0.01	44	NO	775	76.03%	95.98%	88.99%
Mutación lineal decreciente	0.005	0.01	45	NO	775	82.23%	95.76%	91.01%
Mutación lineal decreciente	0.005	0.01	46	SI	159	78.51%	96.21%	90.00%
Mutación lineal decreciente	0.005	0.01	47	SI	214	76.45%	95.76%	88.99%
Mutación lineal decreciente	0.005	0.01	48	SI	198	77.27%	94.64%	88.55%
Mutación lineal decreciente	0.005	0.01	49	SI	221	77.27%	95.76%	89.28%
Mutación lineal decreciente	0.005	0.01	50	NO	776	78.10%	95.76%	89.57%
Mutación lineal decreciente	0.01	0.02	1	NO	775	76.45%	96.43%	89.42%
Mutación lineal decreciente	0.01	0.02	2	SI	165	79.75%	95.98%	90.29%
Mutación lineal decreciente	0.01	0.02	3	SI	144	78.51%	95.09%	89.28%
Mutación lineal decreciente	0.01	0.02	4	SI	234	76.45%	95.76%	88.99%
Mutación lineal decreciente	0.01	0.02	5	SI	181	76.45%	95.31%	88.70%
Mutación lineal decreciente	0.01	0.02	6	SI	149	76.86%	96.65%	89.71%
Mutación lineal decreciente	0.01	0.02	7	SI	179	77.27%	95.76%	89.28%
Mutación lineal decreciente	0.01	0.02	8	SI	239	77.69%	96.43%	89.86%
Mutación lineal decreciente	0.01	0.02	9	SI	138	73.97%	95.54%	87.97%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal decreciente	0.01	0.02	10	SI	138	78.10%	95.76%	89.57%
Mutación lineal decreciente	0.01	0.02	11	SI	152	79.75%	96.43%	90.58%
Mutación lineal decreciente	0.01	0.02	12	SI	228	78.10%	95.76%	89.57%
Mutación lineal decreciente	0.01	0.02	13	NO	775	78.10%	96.88%	90.29%
Mutación lineal decreciente	0.01	0.02	14	SI	221	80.17%	95.76%	90.29%
Mutación lineal decreciente	0.01	0.02	15	SI	215	77.69%	95.98%	89.57%
Mutación lineal decreciente	0.01	0.02	16	SI	207	79.75%	95.54%	90.00%
Mutación lineal decreciente	0.01	0.02	17	SI	229	79.34%	95.98%	90.14%
Mutación lineal decreciente	0.01	0.02	18	SI	181	79.75%	95.76%	90.14%
Mutación lineal decreciente	0.01	0.02	19	SI	199	78.51%	95.54%	89.57%
Mutación lineal decreciente	0.01	0.02	20	SI	206	78.51%	96.21%	90.00%
Mutación lineal decreciente	0.01	0.02	21	SI	251	79.75%	96.65%	90.72%
Mutación lineal decreciente	0.01	0.02	22	SI	167	80.17%	96.21%	90.58%
Mutación lineal decreciente	0.01	0.02	23	SI	137	79.34%	96.21%	90.29%
Mutación lineal decreciente	0.01	0.02	24	SI	160	77.69%	95.54%	89.28%
Mutación lineal decreciente	0.01	0.02	25	SI	200	79.34%	96.65%	90.58%
Mutación lineal decreciente	0.01	0.02	26	NO	775	83.06%	96.65%	91.88%
Mutación lineal decreciente	0.01	0.02	27	NO	773	76.45%	95.98%	89.13%
Mutación lineal decreciente	0.01	0.02	28	SI	185	77.27%	95.76%	89.28%
Mutación lineal decreciente	0.01	0.02	29	SI	198	76.45%	93.75%	87.68%
Mutación lineal decreciente	0.01	0.02	30	SI	224	79.75%	96.21%	90.43%
Mutación lineal decreciente	0.01	0.02	31	NO	773	78.10%	97.10%	90.43%
Mutación lineal decreciente	0.01	0.02	32	SI	156	79.75%	96.43%	90.58%
Mutación lineal decreciente	0.01	0.02	33	SI	262	80.99%	95.76%	90.58%
Mutación lineal decreciente	0.01	0.02	34	SI	217	78.93%	96.21%	90.14%
Mutación lineal decreciente	0.01	0.02	35	SI	132	76.86%	96.21%	89.42%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal decreciente	0.01	0.02	36	SI	153	78.93%	95.98%	90.00%
Mutación lineal decreciente	0.01	0.02	37	SI	218	80.58%	95.54%	90.29%
Mutación lineal decreciente	0.01	0.02	38	NO	774	76.86%	96.21%	89.42%
Mutación lineal decreciente	0.01	0.02	39	SI	143	78.93%	95.76%	89.86%
Mutación lineal decreciente	0.01	0.02	40	SI	193	78.51%	94.87%	89.13%
Mutación lineal decreciente	0.01	0.02	41	SI	264	80.99%	97.32%	91.59%
Mutación lineal decreciente	0.01	0.02	42	SI	256	76.86%	96.43%	89.57%
Mutación lineal decreciente	0.01	0.02	43	SI	220	77.69%	96.21%	89.71%
Mutación lineal decreciente	0.01	0.02	44	SI	269	77.27%	95.76%	89.28%
Mutación lineal decreciente	0.01	0.02	45	SI	148	74.38%	94.87%	87.68%
Mutación lineal decreciente	0.01	0.02	46	SI	153	76.45%	96.65%	89.57%
Mutación lineal decreciente	0.01	0.02	47	SI	207	78.10%	95.54%	89.42%
Mutación lineal decreciente	0.01	0.02	48	NO	773	79.75%	96.43%	90.58%
Mutación lineal decreciente	0.01	0.02	49	SI	152	79.75%	95.98%	90.29%
Mutación lineal decreciente	0.01	0.02	50	SI	195	78.10%	96.88%	90.29%
Mutación lineal decreciente	0.02	0.04	1	SI	217	80.99%	96.21%	90.87%
Mutación lineal decreciente	0.02	0.04	2	SI	280	80.17%	96.88%	91.01%
Mutación lineal decreciente	0.02	0.04	3	SI	173	80.99%	96.65%	91.16%
Mutación lineal decreciente	0.02	0.04	4	SI	172	77.27%	96.88%	90.00%
Mutación lineal decreciente	0.02	0.04	5	SI	257	79.34%	97.54%	91.16%
Mutación lineal decreciente	0.02	0.04	6	NO	774	79.34%	95.09%	89.57%
Mutación lineal decreciente	0.02	0.04	7	NO	774	76.86%	95.76%	89.13%
Mutación lineal decreciente	0.02	0.04	8	SI	198	77.27%	96.21%	89.57%
Mutación lineal decreciente	0.02	0.04	9	SI	222	78.51%	96.88%	90.43%
Mutación lineal decreciente	0.02	0.04	10	SI	167	80.58%	96.43%	90.87%
Mutación lineal decreciente	0.02	0.04	11	SI	136	77.69%	95.98%	89.57%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal decreciente	0.02	0.04	12	SI	170	78.51%	95.76%	89.71%
Mutación lineal decreciente	0.02	0.04	13	SI	204	79.34%	97.32%	91.01%
Mutación lineal decreciente	0.02	0.04	14	SI	137	78.10%	96.21%	89.86%
Mutación lineal decreciente	0.02	0.04	15	SI	169	80.99%	95.76%	90.58%
Mutación lineal decreciente	0.02	0.04	16	SI	242	75.62%	96.21%	88.99%
Mutación lineal decreciente	0.02	0.04	17	SI	206	79.75%	95.98%	90.29%
Mutación lineal decreciente	0.02	0.04	18	SI	179	78.51%	96.43%	90.14%
Mutación lineal decreciente	0.02	0.04	19	SI	215	79.34%	94.87%	89.42%
Mutación lineal decreciente	0.02	0.04	20	SI	173	78.10%	96.65%	90.14%
Mutación lineal decreciente	0.02	0.04	21	SI	258	77.27%	95.54%	89.13%
Mutación lineal decreciente	0.02	0.04	22	SI	212	79.75%	97.10%	91.01%
Mutación lineal decreciente	0.02	0.04	23	SI	277	76.45%	95.31%	88.70%
Mutación lineal decreciente	0.02	0.04	24	SI	230	78.51%	96.65%	90.29%
Mutación lineal decreciente	0.02	0.04	25	SI	263	77.69%	96.21%	89.71%
Mutación lineal decreciente	0.02	0.04	26	NO	775	80.17%	96.65%	90.87%
Mutación lineal decreciente	0.02	0.04	27	SI	149	76.86%	96.43%	89.57%
Mutación lineal decreciente	0.02	0.04	28	SI	172	76.45%	96.88%	89.71%
Mutación lineal decreciente	0.02	0.04	29	SI	341	81.41%	96.65%	91.30%
Mutación lineal decreciente	0.02	0.04	30	SI	168	78.51%	96.43%	90.14%
Mutación lineal decreciente	0.02	0.04	31	SI	159	79.75%	96.21%	90.43%
Mutación lineal decreciente	0.02	0.04	32	SI	187	77.69%	96.88%	90.14%
Mutación lineal decreciente	0.02	0.04	33	SI	151	80.17%	96.43%	90.72%
Mutación lineal decreciente	0.02	0.04	34	SI	158	77.69%	96.43%	89.86%
Mutación lineal decreciente	0.02	0.04	35	SI	170	76.45%	95.98%	89.13%
Mutación lineal decreciente	0.02	0.04	36	SI	242	78.93%	96.21%	90.14%
Mutación lineal decreciente	0.02	0.04	37	SI	184	78.10%	96.21%	89.86%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal decreciente	0.02	0.04	38	NO	775	80.99%	96.65%	91.16%
Mutación lineal decreciente	0.02	0.04	39	SI	146	78.51%	96.88%	90.43%
Mutación lineal decreciente	0.02	0.04	40	SI	280	79.75%	97.10%	91.01%
Mutación lineal decreciente	0.02	0.04	41	SI	296	79.75%	97.10%	91.01%
Mutación lineal decreciente	0.02	0.04	42	SI	202	75.21%	94.87%	87.97%
Mutación lineal decreciente	0.02	0.04	43	NO	772	78.93%	96.88%	90.58%
Mutación lineal decreciente	0.02	0.04	44	NO	773	79.34%	95.98%	90.14%
Mutación lineal decreciente	0.02	0.04	45	SI	159	77.69%	96.43%	89.86%
Mutación lineal decreciente	0.02	0.04	46	NO	773	80.99%	96.88%	91.30%
Mutación lineal decreciente	0.02	0.04	47	SI	229	80.17%	96.88%	91.01%
Mutación lineal decreciente	0.02	0.04	48	SI	238	78.93%	96.88%	90.58%
Mutación lineal decreciente	0.02	0.04	49	SI	167	78.51%	96.88%	90.43%
Mutación lineal decreciente	0.02	0.04	50	SI	207	76.86%	96.21%	89.42%

**Mutación Binaria**

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación uniforme	0.00015625	0.00015625	1	SI	642	78.51%	95.31%	89.42%
Mutación uniforme	0.00015625	0.00015625	2	SI	575	78.51%	94.87%	89.13%
Mutación uniforme	0.00015625	0.00015625	3	SI	550	76.86%	95.31%	88.84%
Mutación uniforme	0.00015625	0.00015625	4	SI	716	76.86%	94.87%	88.55%
Mutación uniforme	0.00015625	0.00015625	5	SI	418	77.27%	95.98%	89.42%
Mutación uniforme	0.00015625	0.00015625	6	SI	679	77.69%	95.76%	89.42%
Mutación uniforme	0.00015625	0.00015625	7	SI	393	78.51%	95.09%	89.28%
Mutación uniforme	0.00015625	0.00015625	8	SI	433	80.17%	97.10%	91.16%



Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación uniforme	0.00015625	0.00015625	9	SI	277	78.93%	95.54%	89.71%
Mutación uniforme	0.00015625	0.00015625	10	SI	501	77.69%	94.20%	88.41%
Mutación uniforme	0.00015625	0.00015625	11	NO	898	79.75%	94.42%	89.28%
Mutación uniforme	0.00015625	0.00015625	12	NO	898	78.51%	94.42%	88.84%
Mutación uniforme	0.00015625	0.00015625	13	SI	296	80.17%	95.98%	90.43%
Mutación uniforme	0.00015625	0.00015625	14	SI	231	78.93%	95.54%	89.71%
Mutación uniforme	0.00015625	0.00015625	15	SI	244	77.69%	95.54%	89.28%
Mutación uniforme	0.00015625	0.00015625	16	SI	573	79.75%	95.76%	90.14%
Mutación uniforme	0.00015625	0.00015625	17	SI	266	80.58%	95.31%	90.14%
Mutación uniforme	0.00015625	0.00015625	18	SI	539	80.58%	95.98%	90.58%
Mutación uniforme	0.00015625	0.00015625	19	SI	403	78.51%	94.64%	88.99%
Mutación uniforme	0.00015625	0.00015625	20	SI	289	79.34%	95.54%	89.86%
Mutación uniforme	0.00015625	0.00015625	21	SI	376	80.99%	96.21%	90.87%
Mutación uniforme	0.00015625	0.00015625	22	SI	399	79.34%	94.87%	89.42%
Mutación uniforme	0.00015625	0.00015625	23	NO	897	76.86%	95.76%	89.13%
Mutación uniforme	0.00015625	0.00015625	24	SI	432	77.69%	95.76%	89.42%
Mutación uniforme	0.00015625	0.00015625	25	SI	304	78.51%	96.43%	90.14%
Mutación uniforme	0.00015625	0.00015625	26	NO	899	77.69%	95.31%	89.13%
Mutación uniforme	0.00015625	0.00015625	27	SI	277	78.10%	95.76%	89.57%
Mutación uniforme	0.00015625	0.00015625	28	SI	738	79.34%	95.09%	89.57%
Mutación uniforme	0.00015625	0.00015625	29	SI	950	75.62%	93.97%	87.54%
Mutación uniforme	0.00015625	0.00015625	30	SI	371	78.51%	95.54%	89.57%
Mutación uniforme	0.00015625	0.00015625	31	SI	483	77.27%	95.54%	89.13%
Mutación uniforme	0.00015625	0.00015625	32	SI	307	79.34%	95.98%	90.14%
Mutación uniforme	0.00015625	0.00015625	33	SI	521	78.93%	95.76%	89.86%
Mutación uniforme	0.00015625	0.00015625	34	SI	463	78.10%	95.98%	89.71%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación uniforme	0.00015625	0.00015625	35	SI	339	80.17%	95.98%	90.43%
Mutación uniforme	0.00015625	0.00015625	36	SI	325	77.27%	95.31%	88.99%
Mutación uniforme	0.00015625	0.00015625	37	NO	895	80.99%	95.76%	90.58%
Mutación uniforme	0.00015625	0.00015625	38	SI	324	78.93%	95.76%	89.86%
Mutación uniforme	0.00015625	0.00015625	39	SI	279	77.69%	94.64%	88.70%
Mutación uniforme	0.00015625	0.00015625	40	SI	547	76.03%	94.64%	88.12%
Mutación uniforme	0.00015625	0.00015625	41	SI	225	77.69%	95.54%	89.28%
Mutación uniforme	0.00015625	0.00015625	42	SI	893	77.27%	94.20%	88.26%
Mutación uniforme	0.00015625	0.00015625	43	SI	341	77.69%	95.09%	88.99%
Mutación uniforme	0.00015625	0.00015625	44	SI	315	78.51%	95.54%	89.57%
Mutación uniforme	0.00015625	0.00015625	45	SI	360	77.27%	95.76%	89.28%
Mutación uniforme	0.00015625	0.00015625	46	SI	193	76.45%	95.76%	88.99%
Mutación uniforme	0.00015625	0.00015625	47	SI	600	78.51%	95.54%	89.57%
Mutación uniforme	0.00015625	0.00015625	48	SI	432	77.27%	95.54%	89.13%
Mutación uniforme	0.00015625	0.00015625	49	SI	530	78.51%	95.31%	89.42%
Mutación uniforme	0.00015625	0.00015625	50	SI	425	78.51%	95.54%	89.57%
Mutación uniforme	0.0003125	0.0003125	1	SI	607	78.10%	95.76%	89.57%
Mutación uniforme	0.0003125	0.0003125	2	SI	349	76.45%	96.88%	89.71%
Mutación uniforme	0.0003125	0.0003125	3	SI	218	81.41%	96.65%	91.30%
Mutación uniforme	0.0003125	0.0003125	4	SI	271	79.34%	97.10%	90.87%
Mutación uniforme	0.0003125	0.0003125	5	SI	241	78.10%	95.09%	89.13%
Mutación uniforme	0.0003125	0.0003125	6	SI	722	74.79%	95.31%	88.12%
Mutación uniforme	0.0003125	0.0003125	7	SI	421	79.75%	95.76%	90.14%
Mutación uniforme	0.0003125	0.0003125	8	SI	598	77.69%	95.31%	89.13%
Mutación uniforme	0.0003125	0.0003125	9	SI	635	77.27%	94.64%	88.55%
Mutación uniforme	0.0003125	0.0003125	10	SI	787	77.27%	95.54%	89.13%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación uniforme	0.0003125	0.0003125	11	SI	397	77.27%	95.31%	88.99%
Mutación uniforme	0.0003125	0.0003125	12	SI	605	78.51%	94.87%	89.13%
Mutación uniforme	0.0003125	0.0003125	13	SI	428	77.69%	95.54%	89.28%
Mutación uniforme	0.0003125	0.0003125	14	SI	328	76.03%	94.20%	87.83%
Mutación uniforme	0.0003125	0.0003125	15	SI	336	78.93%	95.09%	89.42%
Mutación uniforme	0.0003125	0.0003125	16	SI	487	79.34%	94.87%	89.42%
Mutación uniforme	0.0003125	0.0003125	17	NO	904	79.75%	94.20%	89.13%
Mutación uniforme	0.0003125	0.0003125	18	SI	456	78.93%	94.42%	88.99%
Mutación uniforme	0.0003125	0.0003125	19	NO	897	75.21%	95.09%	88.12%
Mutación uniforme	0.0003125	0.0003125	20	SI	429	78.51%	94.42%	88.84%
Mutación uniforme	0.0003125	0.0003125	21	SI	450	76.86%	94.64%	88.41%
Mutación uniforme	0.0003125	0.0003125	22	SI	823	78.51%	95.31%	89.42%
Mutación uniforme	0.0003125	0.0003125	23	NO	898	81.41%	95.09%	90.29%
Mutación uniforme	0.0003125	0.0003125	24	SI	185	78.51%	96.21%	90.00%
Mutación uniforme	0.0003125	0.0003125	25	SI	673	76.45%	94.42%	88.12%
Mutación uniforme	0.0003125	0.0003125	26	SI	426	80.17%	94.20%	89.28%
Mutación uniforme	0.0003125	0.0003125	27	SI	260	80.17%	96.43%	90.72%
Mutación uniforme	0.0003125	0.0003125	28	SI	470	76.03%	92.41%	86.67%
Mutación uniforme	0.0003125	0.0003125	29	SI	537	79.34%	95.76%	90.00%
Mutación uniforme	0.0003125	0.0003125	30	SI	454	77.69%	95.09%	88.99%
Mutación uniforme	0.0003125	0.0003125	31	SI	397	80.58%	94.42%	89.57%
Mutación uniforme	0.0003125	0.0003125	32	SI	305	80.17%	95.31%	90.00%
Mutación uniforme	0.0003125	0.0003125	33	SI	370	81.41%	95.09%	90.29%
Mutación uniforme	0.0003125	0.0003125	34	NO	900	79.34%	95.54%	89.86%
Mutación uniforme	0.0003125	0.0003125	35	SI	372	77.69%	93.97%	88.26%
Mutación uniforme	0.0003125	0.0003125	36	SI	631	77.27%	93.53%	87.83%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación uniforme	0.0003125	0.0003125	37	SI	416	78.10%	95.54%	89.42%
Mutación uniforme	0.0003125	0.0003125	38	NO	900	78.10%	95.09%	89.13%
Mutación uniforme	0.0003125	0.0003125	39	SI	208	77.69%	95.54%	89.28%
Mutación uniforme	0.0003125	0.0003125	40	SI	401	78.93%	94.64%	89.13%
Mutación uniforme	0.0003125	0.0003125	41	SI	597	77.69%	94.64%	88.70%
Mutación uniforme	0.0003125	0.0003125	42	NO	896	79.34%	95.09%	89.57%
Mutación uniforme	0.0003125	0.0003125	43	SI	325	77.27%	95.76%	89.28%
Mutación uniforme	0.0003125	0.0003125	44	SI	520	77.69%	95.98%	89.57%
Mutación uniforme	0.0003125	0.0003125	45	SI	564	76.86%	94.20%	88.12%
Mutación uniforme	0.0003125	0.0003125	46	SI	153	76.86%	96.21%	89.42%
Mutación uniforme	0.0003125	0.0003125	47	SI	886	78.10%	94.87%	88.99%
Mutación uniforme	0.0003125	0.0003125	48	SI	334	79.75%	96.21%	90.43%
Mutación uniforme	0.0003125	0.0003125	49	SI	176	78.93%	94.42%	88.99%
Mutación uniforme	0.0003125	0.0003125	50	SI	449	78.51%	95.54%	89.57%
Mutación uniforme	0.000625	0.000625	1	NO	904	78.51%	94.42%	88.84%
Mutación uniforme	0.000625	0.000625	2	SI	340	78.10%	94.20%	88.55%
Mutación uniforme	0.000625	0.000625	3	SI	484	80.58%	95.31%	90.14%
Mutación uniforme	0.000625	0.000625	4	SI	426	78.93%	95.31%	89.57%
Mutación uniforme	0.000625	0.000625	5	SI	424	78.10%	95.76%	89.57%
Mutación uniforme	0.000625	0.000625	6	SI	434	80.58%	95.09%	90.00%
Mutación uniforme	0.000625	0.000625	7	SI	586	76.45%	95.76%	88.99%
Mutación uniforme	0.000625	0.000625	8	SI	346	78.51%	95.09%	89.28%
Mutación uniforme	0.000625	0.000625	9	SI	501	75.62%	93.75%	87.39%
Mutación uniforme	0.000625	0.000625	10	SI	458	80.99%	94.64%	89.86%
Mutación uniforme	0.000625	0.000625	11	SI	291	79.34%	95.76%	90.00%
Mutación uniforme	0.000625	0.000625	12	NO	901	78.51%	90.40%	86.23%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación uniforme	0.000625	0.000625	13	SI	527	75.62%	95.98%	88.84%
Mutación uniforme	0.000625	0.000625	14	NO	902	75.62%	94.20%	87.68%
Mutación uniforme	0.000625	0.000625	15	SI	376	75.62%	94.20%	87.68%
Mutación uniforme	0.000625	0.000625	16	SI	548	77.69%	95.54%	89.28%
Mutación uniforme	0.000625	0.000625	17	SI	468	80.58%	96.21%	90.72%
Mutación uniforme	0.000625	0.000625	18	SI	820	78.10%	96.21%	89.86%
Mutación uniforme	0.000625	0.000625	19	SI	634	76.03%	96.21%	89.13%
Mutación uniforme	0.000625	0.000625	20	SI	472	80.99%	95.09%	90.14%
Mutación uniforme	0.000625	0.000625	21	NO	913	75.62%	95.76%	88.70%
Mutación uniforme	0.000625	0.000625	22	SI	556	80.17%	95.98%	90.43%
Mutación uniforme	0.000625	0.000625	23	SI	369	78.10%	96.43%	90.00%
Mutación uniforme	0.000625	0.000625	24	SI	741	80.58%	95.54%	90.29%
Mutación uniforme	0.000625	0.000625	25	SI	676	80.58%	95.54%	90.29%
Mutación uniforme	0.000625	0.000625	26	SI	459	80.17%	96.88%	91.01%
Mutación uniforme	0.000625	0.000625	27	SI	582	79.34%	94.87%	89.42%
Mutación uniforme	0.000625	0.000625	28	SI	502	78.93%	95.76%	89.86%
Mutación uniforme	0.000625	0.000625	29	SI	454	77.69%	94.42%	88.55%
Mutación uniforme	0.000625	0.000625	30	SI	535	78.93%	95.76%	89.86%
Mutación uniforme	0.000625	0.000625	31	NO	908	77.69%	94.42%	88.55%
Mutación uniforme	0.000625	0.000625	32	SI	560	78.93%	95.31%	89.57%
Mutación uniforme	0.000625	0.000625	33	SI	803	76.03%	95.31%	88.55%
Mutación uniforme	0.000625	0.000625	34	SI	290	79.75%	95.98%	90.29%
Mutación uniforme	0.000625	0.000625	35	SI	462	79.75%	95.76%	90.14%
Mutación uniforme	0.000625	0.000625	36	SI	300	80.58%	95.54%	90.29%
Mutación uniforme	0.000625	0.000625	37	SI	359	77.69%	95.54%	89.28%
Mutación uniforme	0.000625	0.000625	38	SI	416	79.75%	95.54%	90.00%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación uniforme	0.000625	0.000625	39	SI	385	81.41%	95.76%	90.72%
Mutación uniforme	0.000625	0.000625	40	SI	661	78.51%	95.31%	89.42%
Mutación uniforme	0.000625	0.000625	41	SI	585	76.03%	94.87%	88.26%
Mutación uniforme	0.000625	0.000625	42	SI	758	79.34%	95.76%	90.00%
Mutación uniforme	0.000625	0.000625	43	SI	415	75.21%	94.87%	87.97%
Mutación uniforme	0.000625	0.000625	44	SI	599	78.10%	95.76%	89.57%
Mutación uniforme	0.000625	0.000625	45	SI	517	79.75%	95.76%	90.14%
Mutación uniforme	0.000625	0.000625	46	SI	306	77.27%	95.09%	88.84%
Mutación uniforme	0.000625	0.000625	47	SI	486	77.69%	93.97%	88.26%
Mutación uniforme	0.000625	0.000625	48	SI	538	78.51%	95.54%	89.57%
Mutación uniforme	0.000625	0.000625	49	SI	651	76.86%	95.54%	88.99%
Mutación uniforme	0.000625	0.000625	50	SI	499	75.62%	95.54%	88.55%
Mutación lineal creciente	0.00015625	0.0003125	1	SI	801	79.34%	95.31%	89.71%
Mutación lineal creciente	0.00015625	0.0003125	2	NO	905	78.10%	95.31%	89.28%
Mutación lineal creciente	0.00015625	0.0003125	3	SI	486	79.75%	96.65%	90.72%
Mutación lineal creciente	0.00015625	0.0003125	4	NO	902	76.86%	93.08%	87.39%
Mutación lineal creciente	0.00015625	0.0003125	5	SI	831	76.45%	96.21%	89.28%
Mutación lineal creciente	0.00015625	0.0003125	6	NO	899	77.27%	93.30%	87.68%
Mutación lineal creciente	0.00015625	0.0003125	7	NO	900	78.10%	93.75%	88.26%
Mutación lineal creciente	0.00015625	0.0003125	8	SI	371	79.75%	94.87%	89.57%
Mutación lineal creciente	0.00015625	0.0003125	9	SI	363	79.75%	95.54%	90.00%
Mutación lineal creciente	0.00015625	0.0003125	10	SI	380	81.41%	95.54%	90.58%
Mutación lineal creciente	0.00015625	0.0003125	11	NO	897	80.58%	94.42%	89.57%
Mutación lineal creciente	0.00015625	0.0003125	12	NO	902	79.75%	93.53%	88.70%
Mutación lineal creciente	0.00015625	0.0003125	13	NO	897	78.10%	94.20%	88.55%
Mutación lineal creciente	0.00015625	0.0003125	14	NO	898	77.69%	92.63%	87.39%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal creciente	0.00015625	0.0003125	15	NO	904	78.10%	94.42%	88.70%
Mutación lineal creciente	0.00015625	0.0003125	16	SI	214	77.69%	95.54%	89.28%
Mutación lineal creciente	0.00015625	0.0003125	17	NO	903	77.27%	93.30%	87.68%
Mutación lineal creciente	0.00015625	0.0003125	18	NO	898	76.45%	93.53%	87.54%
Mutación lineal creciente	0.00015625	0.0003125	19	SI	569	76.45%	95.31%	88.70%
Mutación lineal creciente	0.00015625	0.0003125	20	NO	901	77.69%	94.87%	88.84%
Mutación lineal creciente	0.00015625	0.0003125	21	NO	901	78.10%	95.54%	89.42%
Mutación lineal creciente	0.00015625	0.0003125	22	NO	904	77.69%	92.19%	87.10%
Mutación lineal creciente	0.00015625	0.0003125	23	NO	898	78.93%	93.30%	88.26%
Mutación lineal creciente	0.00015625	0.0003125	24	SI	802	76.45%	95.31%	88.70%
Mutación lineal creciente	0.00015625	0.0003125	25	NO	901	76.45%	93.97%	87.83%
Mutación lineal creciente	0.00015625	0.0003125	26	NO	898	78.51%	92.41%	87.54%
Mutación lineal creciente	0.00015625	0.0003125	27	NO	902	80.58%	94.87%	89.86%
Mutación lineal creciente	0.00015625	0.0003125	28	SI	423	80.99%	96.21%	90.87%
Mutación lineal creciente	0.00015625	0.0003125	29	NO	970	76.03%	93.08%	87.10%
Mutación lineal creciente	0.00015625	0.0003125	30	SI	215	81.41%	95.54%	90.58%
Mutación lineal creciente	0.00015625	0.0003125	31	SI	749	80.58%	95.09%	90.00%
Mutación lineal creciente	0.00015625	0.0003125	32	NO	899	79.34%	94.20%	88.99%
Mutación lineal creciente	0.00015625	0.0003125	33	NO	898	72.31%	93.75%	86.23%
Mutación lineal creciente	0.00015625	0.0003125	34	SI	380	77.69%	96.65%	90.00%
Mutación lineal creciente	0.00015625	0.0003125	35	NO	902	80.17%	92.19%	87.97%
Mutación lineal creciente	0.00015625	0.0003125	36	SI	194	79.34%	96.21%	90.29%
Mutación lineal creciente	0.00015625	0.0003125	37	NO	897	78.51%	95.31%	89.42%
Mutación lineal creciente	0.00015625	0.0003125	38	NO	898	79.75%	93.75%	88.84%
Mutación lineal creciente	0.00015625	0.0003125	39	SI	286	77.69%	95.09%	88.99%
Mutación lineal creciente	0.00015625	0.0003125	40	SI	318	77.69%	95.76%	89.42%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal creciente	0.00015625	0.0003125	41	NO	902	78.51%	94.64%	88.99%
Mutación lineal creciente	0.00015625	0.0003125	42	SI	311	77.69%	96.21%	89.71%
Mutación lineal creciente	0.00015625	0.0003125	43	NO	899	79.75%	95.54%	90.00%
Mutación lineal creciente	0.00015625	0.0003125	44	SI	312	79.75%	95.31%	89.86%
Mutación lineal creciente	0.00015625	0.0003125	45	SI	320	82.23%	95.76%	91.01%
Mutación lineal creciente	0.00015625	0.0003125	46	SI	344	79.34%	95.09%	89.57%
Mutación lineal creciente	0.00015625	0.0003125	47	NO	901	75.21%	94.87%	87.97%
Mutación lineal creciente	0.00015625	0.0003125	48	NO	901	76.86%	92.19%	86.81%
Mutación lineal creciente	0.00015625	0.0003125	49	NO	898	78.10%	92.86%	87.68%
Mutación lineal creciente	0.00015625	0.0003125	50	NO	898	76.45%	90.85%	85.80%
Mutación lineal creciente	0.0003125	0.000625	1	NO	910	76.03%	92.63%	86.81%
Mutación lineal creciente	0.0003125	0.000625	2	NO	913	74.38%	91.29%	85.36%
Mutación lineal creciente	0.0003125	0.000625	3	NO	912	77.69%	92.86%	87.54%
Mutación lineal creciente	0.0003125	0.000625	4	NO	912	78.93%	94.64%	89.13%
Mutación lineal creciente	0.0003125	0.000625	5	NO	916	80.17%	91.96%	87.83%
Mutación lineal creciente	0.0003125	0.000625	6	NO	915	75.62%	89.06%	84.35%
Mutación lineal creciente	0.0003125	0.000625	7	NO	908	77.69%	89.06%	85.07%
Mutación lineal creciente	0.0003125	0.000625	8	NO	913	76.86%	89.51%	85.07%
Mutación lineal creciente	0.0003125	0.000625	9	NO	910	78.10%	94.20%	88.55%
Mutación lineal creciente	0.0003125	0.000625	10	NO	915	73.55%	91.07%	84.93%
Mutación lineal creciente	0.0003125	0.000625	11	NO	913	78.10%	90.85%	86.38%
Mutación lineal creciente	0.0003125	0.000625	12	NO	912	80.99%	92.86%	88.70%
Mutación lineal creciente	0.0003125	0.000625	13	NO	915	78.93%	91.96%	87.39%
Mutación lineal creciente	0.0003125	0.000625	14	NO	915	78.10%	93.08%	87.83%
Mutación lineal creciente	0.0003125	0.000625	15	NO	912	76.03%	91.29%	85.94%
Mutación lineal creciente	0.0003125	0.000625	16	NO	912	77.69%	93.30%	87.83%



Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal creciente	0.0003125	0.000625	17	NO	915	76.86%	89.29%	84.93%
Mutación lineal creciente	0.0003125	0.000625	18	NO	913	80.99%	91.74%	87.97%
Mutación lineal creciente	0.0003125	0.000625	19	NO	913	78.10%	91.07%	86.52%
Mutación lineal creciente	0.0003125	0.000625	20	NO	917	75.62%	91.29%	85.80%
Mutación lineal creciente	0.0003125	0.000625	21	NO	908	78.51%	90.40%	86.23%
Mutación lineal creciente	0.0003125	0.000625	22	NO	909	77.27%	91.52%	86.52%
Mutación lineal creciente	0.0003125	0.000625	23	NO	912	78.51%	93.75%	88.41%
Mutación lineal creciente	0.0003125	0.000625	24	NO	910	79.34%	91.52%	87.25%
Mutación lineal creciente	0.0003125	0.000625	25	NO	911	76.86%	91.96%	86.67%
Mutación lineal creciente	0.0003125	0.000625	26	NO	916	77.27%	91.29%	86.38%
Mutación lineal creciente	0.0003125	0.000625	27	NO	911	77.27%	91.96%	86.81%
Mutación lineal creciente	0.0003125	0.000625	28	NO	913	80.58%	95.09%	90.00%
Mutación lineal creciente	0.0003125	0.000625	29	NO	982	79.75%	88.39%	85.36%
Mutación lineal creciente	0.0003125	0.000625	30	NO	913	82.23%	90.18%	87.39%
Mutación lineal creciente	0.0003125	0.000625	31	NO	912	82.64%	93.08%	89.42%
Mutación lineal creciente	0.0003125	0.000625	32	NO	913	77.27%	92.86%	87.39%
Mutación lineal creciente	0.0003125	0.000625	33	NO	916	80.17%	91.52%	87.54%
Mutación lineal creciente	0.0003125	0.000625	34	NO	916	78.51%	90.63%	86.38%
Mutación lineal creciente	0.0003125	0.000625	35	NO	911	78.51%	91.96%	87.25%
Mutación lineal creciente	0.0003125	0.000625	36	NO	909	77.69%	92.19%	87.10%
Mutación lineal creciente	0.0003125	0.000625	37	NO	910	77.69%	91.74%	86.81%
Mutación lineal creciente	0.0003125	0.000625	38	NO	911	80.17%	91.96%	87.83%
Mutación lineal creciente	0.0003125	0.000625	39	NO	916	77.27%	89.96%	85.51%
Mutación lineal creciente	0.0003125	0.000625	40	NO	914	77.27%	94.42%	88.41%
Mutación lineal creciente	0.0003125	0.000625	41	NO	911	80.58%	92.63%	88.41%
Mutación lineal creciente	0.0003125	0.000625	42	NO	916	78.51%	90.40%	86.23%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal creciente	0.0003125	0.000625	43	NO	913	78.10%	91.29%	86.67%
Mutación lineal creciente	0.0003125	0.000625	44	NO	912	77.27%	93.08%	87.54%
Mutación lineal creciente	0.0003125	0.000625	45	NO	916	74.79%	89.51%	84.35%
Mutación lineal creciente	0.0003125	0.000625	46	NO	910	75.62%	89.06%	84.35%
Mutación lineal creciente	0.0003125	0.000625	47	NO	909	79.75%	88.62%	85.51%
Mutación lineal creciente	0.0003125	0.000625	48	NO	913	78.93%	95.98%	90.00%
Mutación lineal creciente	0.0003125	0.000625	49	NO	912	77.69%	93.08%	87.68%
Mutación lineal creciente	0.0003125	0.000625	50	NO	912	76.86%	91.96%	86.67%
Mutación lineal creciente	0.000625	0.00125	1	NO	919	82.64%	90.63%	87.83%
Mutación lineal creciente	0.000625	0.00125	2	NO	924	78.51%	86.83%	83.91%
Mutación lineal creciente	0.000625	0.00125	3	NO	926	79.34%	91.29%	87.10%
Mutación lineal creciente	0.000625	0.00125	4	NO	922	78.51%	85.71%	83.19%
Mutación lineal creciente	0.000625	0.00125	5	NO	922	80.99%	90.85%	87.39%
Mutación lineal creciente	0.000625	0.00125	6	NO	920	77.27%	85.49%	82.61%
Mutación lineal creciente	0.000625	0.00125	7	NO	925	80.17%	87.05%	84.64%
Mutación lineal creciente	0.000625	0.00125	8	NO	923	79.75%	88.62%	85.51%
Mutación lineal creciente	0.000625	0.00125	9	NO	923	79.34%	91.29%	87.10%
Mutación lineal creciente	0.000625	0.00125	10	NO	925	82.64%	91.07%	88.12%
Mutación lineal creciente	0.000625	0.00125	11	NO	921	81.82%	89.73%	86.96%
Mutación lineal creciente	0.000625	0.00125	12	NO	925	79.75%	90.85%	86.96%
Mutación lineal creciente	0.000625	0.00125	13	NO	923	76.03%	88.39%	84.06%
Mutación lineal creciente	0.000625	0.00125	14	NO	928	82.23%	90.63%	87.68%
Mutación lineal creciente	0.000625	0.00125	15	NO	923	80.17%	91.74%	87.68%
Mutación lineal creciente	0.000625	0.00125	16	NO	926	81.82%	90.85%	87.68%
Mutación lineal creciente	0.000625	0.00125	17	NO	924	77.69%	90.40%	85.94%
Mutación lineal creciente	0.000625	0.00125	18	NO	925	74.79%	89.51%	84.35%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal creciente	0.000625	0.00125	19	NO	921	79.75%	89.06%	85.80%
Mutación lineal creciente	0.000625	0.00125	20	NO	923	79.75%	90.85%	86.96%
Mutación lineal creciente	0.000625	0.00125	21	NO	927	79.34%	91.52%	87.25%
Mutación lineal creciente	0.000625	0.00125	22	NO	925	80.58%	91.74%	87.83%
Mutación lineal creciente	0.000625	0.00125	23	NO	923	78.93%	92.19%	87.54%
Mutación lineal creciente	0.000625	0.00125	24	NO	926	80.17%	88.39%	85.51%
Mutación lineal creciente	0.000625	0.00125	25	NO	922	78.93%	88.17%	84.93%
Mutación lineal creciente	0.000625	0.00125	26	NO	923	81.41%	89.51%	86.67%
Mutación lineal creciente	0.000625	0.00125	27	NO	925	77.69%	90.85%	86.23%
Mutación lineal creciente	0.000625	0.00125	28	NO	924	73.14%	89.29%	83.62%
Mutación lineal creciente	0.000625	0.00125	29	NO	920	78.10%	89.96%	85.80%
Mutación lineal creciente	0.000625	0.00125	30	NO	924	75.62%	88.84%	84.20%
Mutación lineal creciente	0.000625	0.00125	31	NO	921	79.34%	90.40%	86.52%
Mutación lineal creciente	0.000625	0.00125	32	NO	925	74.79%	88.39%	83.62%
Mutación lineal creciente	0.000625	0.00125	33	NO	921	79.34%	91.29%	87.10%
Mutación lineal creciente	0.000625	0.00125	34	NO	924	80.58%	89.29%	86.23%
Mutación lineal creciente	0.000625	0.00125	35	NO	921	82.23%	89.96%	87.25%
Mutación lineal creciente	0.000625	0.00125	36	NO	923	81.82%	91.52%	88.12%
Mutación lineal creciente	0.000625	0.00125	37	NO	926	79.75%	89.96%	86.38%
Mutación lineal creciente	0.000625	0.00125	38	NO	926	73.55%	88.62%	83.33%
Mutación lineal creciente	0.000625	0.00125	39	NO	924	79.34%	90.63%	86.67%
Mutación lineal creciente	0.000625	0.00125	40	NO	924	78.10%	91.07%	86.52%
Mutación lineal creciente	0.000625	0.00125	41	NO	922	78.93%	89.29%	85.65%
Mutación lineal creciente	0.000625	0.00125	42	NO	924	76.03%	88.17%	83.91%
Mutación lineal creciente	0.000625	0.00125	43	NO	925	77.27%	88.17%	84.35%
Mutación lineal creciente	0.000625	0.00125	44	NO	924	78.10%	90.85%	86.38%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal creciente	0.000625	0.00125	45	NO	921	80.17%	89.51%	86.23%
Mutación lineal creciente	0.000625	0.00125	46	NO	924	77.69%	90.85%	86.23%
Mutación lineal creciente	0.000625	0.00125	47	NO	924	78.51%	90.18%	86.09%
Mutación lineal creciente	0.000625	0.00125	48	NO	922	73.14%	87.95%	82.75%
Mutación lineal creciente	0.000625	0.00125	49	NO	921	78.51%	88.62%	85.07%
Mutación lineal creciente	0.000625	0.00125	50	NO	925	79.75%	88.39%	85.36%
Mutación lineal decreciente	0.000078125	0.00015625	1	SI	266	77.27%	95.31%	88.99%
Mutación lineal decreciente	0.000078125	0.00015625	2	NO	899	79.75%	95.09%	89.71%
Mutación lineal decreciente	0.000078125	0.00015625	3	NO	898	78.93%	94.87%	89.28%
Mutación lineal decreciente	0.000078125	0.00015625	4	NO	896	77.69%	93.97%	88.26%
Mutación lineal decreciente	0.000078125	0.00015625	5	NO	902	77.69%	95.31%	89.13%
Mutación lineal decreciente	0.000078125	0.00015625	6	NO	898	78.10%	94.87%	88.99%
Mutación lineal decreciente	0.000078125	0.00015625	7	NO	901	78.10%	93.08%	87.83%
Mutación lineal decreciente	0.000078125	0.00015625	8	NO	898	78.51%	93.97%	88.55%
Mutación lineal decreciente	0.000078125	0.00015625	9	NO	897	80.17%	94.20%	89.28%
Mutación lineal decreciente	0.000078125	0.00015625	10	NO	896	82.23%	93.30%	89.42%
Mutación lineal decreciente	0.000078125	0.00015625	11	NO	895	79.34%	94.87%	89.42%
Mutación lineal decreciente	0.000078125	0.00015625	12	NO	900	79.34%	93.53%	88.55%
Mutación lineal decreciente	0.000078125	0.00015625	13	NO	896	77.69%	94.20%	88.41%
Mutación lineal decreciente	0.000078125	0.00015625	14	NO	906	81.41%	94.87%	90.14%
Mutación lineal decreciente	0.000078125	0.00015625	15	NO	901	76.86%	94.42%	88.26%
Mutación lineal decreciente	0.000078125	0.00015625	16	NO	897	77.69%	94.87%	88.84%
Mutación lineal decreciente	0.000078125	0.00015625	17	NO	895	76.03%	95.09%	88.41%
Mutación lineal decreciente	0.000078125	0.00015625	18	NO	897	80.58%	95.31%	90.14%
Mutación lineal decreciente	0.000078125	0.00015625	19	SI	350	78.93%	95.76%	89.86%
Mutación lineal decreciente	0.000078125	0.00015625	20	NO	903	76.03%	94.20%	87.83%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal decreciente	0.000078125	0.00015625	21	NO	896	79.75%	95.09%	89.71%
Mutación lineal decreciente	0.000078125	0.00015625	22	NO	901	79.34%	95.98%	90.14%
Mutación lineal decreciente	0.000078125	0.00015625	23	NO	898	78.51%	94.64%	88.99%
Mutación lineal decreciente	0.000078125	0.00015625	24	SI	299	80.58%	95.31%	90.14%
Mutación lineal decreciente	0.000078125	0.00015625	25	NO	897	77.69%	95.09%	88.99%
Mutación lineal decreciente	0.000078125	0.00015625	26	NO	896	78.51%	95.54%	89.57%
Mutación lineal decreciente	0.000078125	0.00015625	27	NO	899	80.99%	93.97%	89.42%
Mutación lineal decreciente	0.000078125	0.00015625	28	NO	899	80.17%	91.96%	87.83%
Mutación lineal decreciente	0.000078125	0.00015625	29	NO	898	76.86%	94.42%	88.26%
Mutación lineal decreciente	0.000078125	0.00015625	30	SI	221	78.51%	95.54%	89.57%
Mutación lineal decreciente	0.000078125	0.00015625	31	NO	897	77.27%	93.97%	88.12%
Mutación lineal decreciente	0.000078125	0.00015625	32	NO	899	79.75%	94.20%	89.13%
Mutación lineal decreciente	0.000078125	0.00015625	33	NO	896	79.75%	95.09%	89.71%
Mutación lineal decreciente	0.000078125	0.00015625	34	NO	896	78.93%	94.20%	88.84%
Mutación lineal decreciente	0.000078125	0.00015625	35	NO	902	78.93%	94.20%	88.84%
Mutación lineal decreciente	0.000078125	0.00015625	36	NO	897	77.27%	94.64%	88.55%
Mutación lineal decreciente	0.000078125	0.00015625	37	SI	295	77.69%	95.54%	89.28%
Mutación lineal decreciente	0.000078125	0.00015625	38	NO	895	80.17%	96.21%	90.58%
Mutación lineal decreciente	0.000078125	0.00015625	39	NO	896	78.93%	94.64%	89.13%
Mutación lineal decreciente	0.000078125	0.00015625	40	NO	895	78.93%	94.42%	88.99%
Mutación lineal decreciente	0.000078125	0.00015625	41	NO	893	75.21%	93.75%	87.25%
Mutación lineal decreciente	0.000078125	0.00015625	42	NO	897	80.58%	94.87%	89.86%
Mutación lineal decreciente	0.000078125	0.00015625	43	NO	898	78.10%	95.09%	89.13%
Mutación lineal decreciente	0.000078125	0.00015625	44	NO	898	77.69%	92.41%	87.25%
Mutación lineal decreciente	0.000078125	0.00015625	45	NO	900	78.51%	93.97%	88.55%
Mutación lineal decreciente	0.000078125	0.00015625	46	NO	900	78.51%	95.98%	89.86%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal decreciente	0.000078125	0.00015625	47	NO	898	78.93%	95.09%	89.42%
Mutación lineal decreciente	0.000078125	0.00015625	48	NO	895	76.45%	95.31%	88.70%
Mutación lineal decreciente	0.000078125	0.00015625	49	NO	897	80.58%	95.76%	90.43%
Mutación lineal decreciente	0.000078125	0.00015625	50	NO	899	79.34%	94.42%	89.13%
Mutación lineal decreciente	0.00015625	0.0003125	1	NO	898	77.69%	94.42%	88.55%
Mutación lineal decreciente	0.00015625	0.0003125	2	SI	299	79.34%	94.87%	89.42%
Mutación lineal decreciente	0.00015625	0.0003125	3	SI	310	80.58%	95.09%	90.00%
Mutación lineal decreciente	0.00015625	0.0003125	4	SI	250	79.75%	95.54%	90.00%
Mutación lineal decreciente	0.00015625	0.0003125	5	NO	897	78.93%	95.09%	89.42%
Mutación lineal decreciente	0.00015625	0.0003125	6	NO	897	80.58%	95.09%	90.00%
Mutación lineal decreciente	0.00015625	0.0003125	7	SI	204	76.45%	96.21%	89.28%
Mutación lineal decreciente	0.00015625	0.0003125	8	NO	896	78.93%	95.09%	89.42%
Mutación lineal decreciente	0.00015625	0.0003125	9	SI	225	77.69%	93.08%	87.68%
Mutación lineal decreciente	0.00015625	0.0003125	10	SI	200	80.99%	95.31%	90.29%
Mutación lineal decreciente	0.00015625	0.0003125	11	NO	900	77.69%	94.42%	88.55%
Mutación lineal decreciente	0.00015625	0.0003125	12	SI	328	77.69%	95.76%	89.42%
Mutación lineal decreciente	0.00015625	0.0003125	13	NO	903	79.34%	95.54%	89.86%
Mutación lineal decreciente	0.00015625	0.0003125	14	NO	897	78.93%	94.42%	88.99%
Mutación lineal decreciente	0.00015625	0.0003125	15	SI	306	77.27%	95.76%	89.28%
Mutación lineal decreciente	0.00015625	0.0003125	16	SI	298	79.75%	95.31%	89.86%
Mutación lineal decreciente	0.00015625	0.0003125	17	NO	897	81.82%	95.09%	90.43%
Mutación lineal decreciente	0.00015625	0.0003125	18	SI	268	76.86%	95.98%	89.28%
Mutación lineal decreciente	0.00015625	0.0003125	19	NO	894	80.99%	95.76%	90.58%
Mutación lineal decreciente	0.00015625	0.0003125	20	SI	271	80.17%	93.97%	89.13%
Mutación lineal decreciente	0.00015625	0.0003125	21	NO	897	80.58%	94.42%	89.57%
Mutación lineal decreciente	0.00015625	0.0003125	22	NO	898	77.69%	96.21%	89.71%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal decreciente	0.00015625	0.0003125	23	NO	898	77.69%	95.09%	88.99%
Mutación lineal decreciente	0.00015625	0.0003125	24	NO	899	79.75%	95.31%	89.86%
Mutación lineal decreciente	0.00015625	0.0003125	25	SI	311	78.10%	95.09%	89.13%
Mutación lineal decreciente	0.00015625	0.0003125	26	NO	898	76.45%	93.08%	87.25%
Mutación lineal decreciente	0.00015625	0.0003125	27	NO	902	77.27%	92.41%	87.10%
Mutación lineal decreciente	0.00015625	0.0003125	28	NO	896	77.69%	94.42%	88.55%
Mutación lineal decreciente	0.00015625	0.0003125	29	SI	226	78.93%	94.87%	89.28%
Mutación lineal decreciente	0.00015625	0.0003125	30	NO	904	76.86%	95.54%	88.99%
Mutación lineal decreciente	0.00015625	0.0003125	31	NO	895	78.93%	93.75%	88.55%
Mutación lineal decreciente	0.00015625	0.0003125	32	SI	265	79.75%	93.75%	88.84%
Mutación lineal decreciente	0.00015625	0.0003125	33	NO	901	79.75%	93.97%	88.99%
Mutación lineal decreciente	0.00015625	0.0003125	34	SI	286	79.34%	95.31%	89.71%
Mutación lineal decreciente	0.00015625	0.0003125	35	SI	305	75.62%	95.76%	88.70%
Mutación lineal decreciente	0.00015625	0.0003125	36	NO	896	80.58%	95.31%	90.14%
Mutación lineal decreciente	0.00015625	0.0003125	37	NO	898	79.75%	95.31%	89.86%
Mutación lineal decreciente	0.00015625	0.0003125	38	NO	896	75.62%	93.30%	87.10%
Mutación lineal decreciente	0.00015625	0.0003125	39	SI	296	80.99%	96.21%	90.87%
Mutación lineal decreciente	0.00015625	0.0003125	40	NO	896	76.45%	91.74%	86.38%
Mutación lineal decreciente	0.00015625	0.0003125	41	SI	196	79.75%	95.76%	90.14%
Mutación lineal decreciente	0.00015625	0.0003125	42	SI	235	78.10%	96.43%	90.00%
Mutación lineal decreciente	0.00015625	0.0003125	43	NO	900	78.93%	95.31%	89.57%
Mutación lineal decreciente	0.00015625	0.0003125	44	NO	897	78.10%	95.54%	89.42%
Mutación lineal decreciente	0.00015625	0.0003125	45	NO	897	78.10%	94.42%	88.70%
Mutación lineal decreciente	0.00015625	0.0003125	46	NO	897	79.75%	95.31%	89.86%
Mutación lineal decreciente	0.00015625	0.0003125	47	NO	898	76.86%	92.63%	87.10%
Mutación lineal decreciente	0.00015625	0.0003125	48	NO	897	76.45%	95.54%	88.84%

Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal decreciente	0.00015625	0.0003125	49	NO	895	78.93%	95.98%	90.00%
Mutación lineal decreciente	0.00015625	0.0003125	50	SI	205	80.58%	96.21%	90.72%
Mutación lineal decreciente	0.0003125	0.000625	1	SI	226	79.75%	95.98%	90.29%
Mutación lineal decreciente	0.0003125	0.000625	2	NO	902	79.75%	93.53%	88.70%
Mutación lineal decreciente	0.0003125	0.000625	3	SI	223	80.17%	96.21%	90.58%
Mutación lineal decreciente	0.0003125	0.000625	4	NO	901	77.27%	95.98%	89.42%
Mutación lineal decreciente	0.0003125	0.000625	5	SI	175	76.86%	93.75%	87.83%
Mutación lineal decreciente	0.0003125	0.000625	6	NO	899	78.93%	95.09%	89.42%
Mutación lineal decreciente	0.0003125	0.000625	7	NO	897	77.69%	93.75%	88.12%
Mutación lineal decreciente	0.0003125	0.000625	8	NO	905	75.21%	92.86%	86.67%
Mutación lineal decreciente	0.0003125	0.000625	9	NO	900	75.62%	93.30%	87.10%
Mutación lineal decreciente	0.0003125	0.000625	10	NO	900	78.10%	95.09%	89.13%
Mutación lineal decreciente	0.0003125	0.000625	11	SI	261	76.03%	93.53%	87.39%
Mutación lineal decreciente	0.0003125	0.000625	12	SI	180	77.69%	94.42%	88.55%
Mutación lineal decreciente	0.0003125	0.000625	13	NO	904	79.75%	95.98%	90.29%
Mutación lineal decreciente	0.0003125	0.000625	14	SI	185	79.34%	95.09%	89.57%
Mutación lineal decreciente	0.0003125	0.000625	15	SI	279	78.51%	95.54%	89.57%
Mutación lineal decreciente	0.0003125	0.000625	16	NO	902	78.10%	95.31%	89.28%
Mutación lineal decreciente	0.0003125	0.000625	17	SI	260	78.93%	95.09%	89.42%
Mutación lineal decreciente	0.0003125	0.000625	18	SI	356	78.10%	95.31%	89.28%
Mutación lineal decreciente	0.0003125	0.000625	19	NO	905	78.51%	94.64%	88.99%
Mutación lineal decreciente	0.0003125	0.000625	20	SI	346	77.69%	94.20%	88.41%
Mutación lineal decreciente	0.0003125	0.000625	21	SI	264	80.58%	96.21%	90.72%
Mutación lineal decreciente	0.0003125	0.000625	22	SI	291	79.34%	95.76%	90.00%
Mutación lineal decreciente	0.0003125	0.000625	23	NO	909	77.69%	95.31%	89.13%
Mutación lineal decreciente	0.0003125	0.000625	24	NO	909	80.17%	95.09%	89.86%



Tipo Función Mutación	Min Mutación	Max Mutación	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Mutación lineal decreciente	0.0003125	0.000625	25	NO	901	78.51%	92.86%	87.83%
Mutación lineal decreciente	0.0003125	0.000625	26	SI	255	80.17%	94.42%	89.42%
Mutación lineal decreciente	0.0003125	0.000625	27	SI	165	78.93%	95.09%	89.42%
Mutación lineal decreciente	0.0003125	0.000625	28	NO	900	80.99%	95.76%	90.58%
Mutación lineal decreciente	0.0003125	0.000625	29	SI	263	78.93%	96.65%	90.43%
Mutación lineal decreciente	0.0003125	0.000625	30	NO	902	79.34%	93.30%	88.41%
Mutación lineal decreciente	0.0003125	0.000625	31	SI	216	76.86%	95.31%	88.84%
Mutación lineal decreciente	0.0003125	0.000625	32	NO	905	78.51%	94.64%	88.99%
Mutación lineal decreciente	0.0003125	0.000625	33	NO	900	79.34%	95.31%	89.71%
Mutación lineal decreciente	0.0003125	0.000625	34	NO	896	80.99%	93.53%	89.13%
Mutación lineal decreciente	0.0003125	0.000625	35	SI	245	78.93%	95.09%	89.42%
Mutación lineal decreciente	0.0003125	0.000625	36	SI	281	78.51%	96.88%	90.43%
Mutación lineal decreciente	0.0003125	0.000625	37	SI	299	74.79%	95.76%	88.41%
Mutación lineal decreciente	0.0003125	0.000625	38	NO	905	80.58%	96.43%	90.87%
Mutación lineal decreciente	0.0003125	0.000625	39	SI	292	79.34%	95.98%	90.14%
Mutación lineal decreciente	0.0003125	0.000625	40	NO	904	80.99%	95.31%	90.29%
Mutación lineal decreciente	0.0003125	0.000625	41	SI	283	78.93%	95.54%	89.71%
Mutación lineal decreciente	0.0003125	0.000625	42	SI	323	77.69%	93.75%	88.12%
Mutación lineal decreciente	0.0003125	0.000625	43	SI	234	77.69%	94.64%	88.70%
Mutación lineal decreciente	0.0003125	0.000625	44	SI	202	79.75%	94.87%	89.57%
Mutación lineal decreciente	0.0003125	0.000625	45	SI	171	79.34%	96.21%	90.29%
Mutación lineal decreciente	0.0003125	0.000625	46	SI	291	74.79%	96.21%	88.70%
Mutación lineal decreciente	0.0003125	0.000625	47	SI	289	79.34%	95.98%	90.14%
Mutación lineal decreciente	0.0003125	0.000625	48	SI	283	79.34%	95.54%	89.86%
Mutación lineal decreciente	0.0003125	0.000625	49	SI	224	78.10%	92.86%	87.68%
Mutación lineal decreciente	0.0003125	0.000625	50	NO	904	80.17%	95.31%	90.00%

**A.6. Valores de inicialización**

Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random Selectiva [-1,1]	1	SI	191	77.27%	96.21%	89.57%
Inicialización Random Selectiva [-1,1]	2	SI	190	80.99%	96.88%	91.30%
Inicialización Random Selectiva [-1,1]	3	SI	219	80.17%	96.65%	90.87%
Inicialización Random Selectiva [-1,1]	4	SI	169	77.69%	95.31%	89.13%
Inicialización Random Selectiva [-1,1]	5	SI	251	78.10%	96.43%	90.00%
Inicialización Random Selectiva [-1,1]	6	SI	181	80.17%	96.21%	90.58%
Inicialización Random Selectiva [-1,1]	7	SI	226	81.41%	97.32%	91.74%
Inicialización Random Selectiva [-1,1]	8	SI	528	79.75%	96.88%	90.87%
Inicialización Random Selectiva [-1,1]	9	SI	138	77.27%	96.65%	89.86%
Inicialización Random Selectiva [-1,1]	10	SI	135	80.99%	96.43%	91.01%
Inicialización Random Selectiva [-1,1]	11	SI	154	77.27%	95.98%	89.42%
Inicialización Random Selectiva [-1,1]	12	SI	198	80.99%	96.21%	90.87%
Inicialización Random Selectiva [-1,1]	13	SI	181	79.34%	96.21%	90.29%
Inicialización Random Selectiva [-1,1]	14	SI	181	78.51%	96.43%	90.14%
Inicialización Random Selectiva [-1,1]	15	SI	217	77.27%	96.88%	90.00%
Inicialización Random Selectiva [-1,1]	16	SI	154	80.17%	94.64%	89.57%
Inicialización Random Selectiva [-1,1]	17	SI	250	80.58%	96.43%	90.87%
Inicialización Random Selectiva [-1,1]	18	SI	237	78.93%	96.43%	90.29%
Inicialización Random Selectiva [-1,1]	19	SI	142	77.27%	94.87%	88.70%
Inicialización Random Selectiva [-1,1]	20	SI	162	77.69%	96.43%	89.86%
Inicialización Random Selectiva [-1,1]	21	SI	172	77.27%	96.21%	89.57%
Inicialización Random Selectiva [-1,1]	22	SI	163	77.69%	95.31%	89.13%

Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random Selectiva [-1,1]	23	SI	744	80.17%	97.32%	91.30%
Inicialización Random Selectiva [-1,1]	24	SI	145	78.51%	96.21%	90.00%
Inicialización Random Selectiva [-1,1]	25	SI	253	76.86%	96.88%	89.86%
Inicialización Random Selectiva [-1,1]	26	SI	173	78.93%	96.21%	90.14%
Inicialización Random Selectiva [-1,1]	27	SI	238	73.97%	95.76%	88.12%
Inicialización Random Selectiva [-1,1]	28	SI	175	76.86%	95.54%	88.99%
Inicialización Random Selectiva [-1,1]	29	SI	154	78.51%	96.21%	90.00%
Inicialización Random Selectiva [-1,1]	30	SI	388	79.34%	97.10%	90.87%
Inicialización Random Selectiva [-1,1]	31	SI	197	79.75%	96.43%	90.58%
Inicialización Random Selectiva [-1,1]	32	SI	351	77.69%	97.10%	90.29%
Inicialización Random Selectiva [-1,1]	33	SI	227	79.75%	96.88%	90.87%
Inicialización Random Selectiva [-1,1]	34	SI	234	79.75%	96.88%	90.87%
Inicialización Random Selectiva [-1,1]	35	SI	175	80.58%	96.65%	91.01%
Inicialización Random Selectiva [-1,1]	36	SI	685	80.99%	96.43%	91.01%
Inicialización Random Selectiva [-1,1]	37	SI	176	78.51%	96.21%	90.00%
Inicialización Random Selectiva [-1,1]	38	SI	204	78.10%	96.21%	89.86%
Inicialización Random Selectiva [-1,1]	39	SI	191	76.03%	96.88%	89.57%
Inicialización Random Selectiva [-1,1]	40	SI	171	80.17%	95.54%	90.14%
Inicialización Random Selectiva [-1,1]	41	SI	195	77.69%	95.54%	89.28%
Inicialización Random Selectiva [-1,1]	42	SI	139	76.45%	95.31%	88.70%
Inicialización Random Selectiva [-1,1]	43	SI	193	82.23%	95.98%	91.16%
Inicialización Random Selectiva [-1,1]	44	SI	201	77.69%	96.88%	90.14%
Inicialización Random Selectiva [-1,1]	45	SI	256	78.51%	96.88%	90.43%
Inicialización Random Selectiva [-1,1]	46	SI	147	76.45%	96.43%	89.42%
Inicialización Random Selectiva [-1,1]	47	SI	236	79.34%	96.43%	90.43%
Inicialización Random Selectiva [-1,1]	48	SI	172	77.69%	96.21%	89.71%
Inicialización Random Selectiva [-1,1]	49	SI	195	76.03%	96.43%	89.28%

Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random Selectiva [-1,1]	50	SI	161	78.10%	95.76%	89.57%
Inicialización Random Selectiva [-0.01,0.01]	1	SI	190	80.99%	96.65%	91.16%
Inicialización Random Selectiva [-0.01,0.01]	2	SI	199	80.17%	95.76%	90.29%
Inicialización Random Selectiva [-0.01,0.01]	3	SI	168	77.69%	96.43%	89.86%
Inicialización Random Selectiva [-0.01,0.01]	4	SI	236	79.34%	96.88%	90.72%
Inicialización Random Selectiva [-0.01,0.01]	5	SI	219	78.10%	95.76%	89.57%
Inicialización Random Selectiva [-0.01,0.01]	6	SI	218	80.58%	96.21%	90.72%
Inicialización Random Selectiva [-0.01,0.01]	7	SI	240	74.79%	95.31%	88.12%
Inicialización Random Selectiva [-0.01,0.01]	8	SI	243	80.17%	97.32%	91.30%
Inicialización Random Selectiva [-0.01,0.01]	9	SI	246	78.10%	97.10%	90.43%
Inicialización Random Selectiva [-0.01,0.01]	10	SI	373	78.51%	95.54%	89.57%
Inicialización Random Selectiva [-0.01,0.01]	11	SI	163	76.45%	95.98%	89.13%
Inicialización Random Selectiva [-0.01,0.01]	12	SI	196	80.99%	96.88%	91.30%
Inicialización Random Selectiva [-0.01,0.01]	13	SI	207	78.10%	95.98%	89.71%
Inicialización Random Selectiva [-0.01,0.01]	14	SI	164	78.93%	96.88%	90.58%
Inicialización Random Selectiva [-0.01,0.01]	15	SI	251	79.34%	96.21%	90.29%
Inicialización Random Selectiva [-0.01,0.01]	16	SI	193	76.03%	95.76%	88.84%
Inicialización Random Selectiva [-0.01,0.01]	17	SI	282	78.10%	96.21%	89.86%
Inicialización Random Selectiva [-0.01,0.01]	18	SI	149	76.45%	95.98%	89.13%
Inicialización Random Selectiva [-0.01,0.01]	19	SI	272	78.10%	96.21%	89.86%
Inicialización Random Selectiva [-0.01,0.01]	20	SI	196	79.34%	94.64%	89.28%
Inicialización Random Selectiva [-0.01,0.01]	21	SI	346	80.58%	96.65%	91.01%
Inicialización Random Selectiva [-0.01,0.01]	22	SI	189	77.27%	96.43%	89.71%
Inicialización Random Selectiva [-0.01,0.01]	23	SI	170	76.03%	96.43%	89.28%
Inicialización Random Selectiva [-0.01,0.01]	24	SI	191	75.62%	95.31%	88.41%
Inicialización Random Selectiva [-0.01,0.01]	25	SI	275	74.38%	95.54%	88.12%
Inicialización Random Selectiva [-0.01,0.01]	26	SI	214	76.86%	96.43%	89.57%

Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random Selectiva [-0.01,0.01]	27	SI	306	78.93%	95.98%	90.00%
Inicialización Random Selectiva [-0.01,0.01]	28	SI	248	78.51%	96.43%	90.14%
Inicialización Random Selectiva [-0.01,0.01]	29	SI	310	80.99%	97.10%	91.45%
Inicialización Random Selectiva [-0.01,0.01]	30	SI	323	79.34%	96.43%	90.43%
Inicialización Random Selectiva [-0.01,0.01]	31	SI	282	75.21%	95.09%	88.12%
Inicialización Random Selectiva [-0.01,0.01]	32	SI	158	78.51%	95.76%	89.71%
Inicialización Random Selectiva [-0.01,0.01]	33	SI	319	78.93%	96.43%	90.29%
Inicialización Random Selectiva [-0.01,0.01]	34	SI	421	76.45%	97.10%	89.86%
Inicialización Random Selectiva [-0.01,0.01]	35	SI	214	78.10%	96.21%	89.86%
Inicialización Random Selectiva [-0.01,0.01]	36	SI	248	79.34%	96.43%	90.43%
Inicialización Random Selectiva [-0.01,0.01]	37	SI	278	79.75%	97.32%	91.16%
Inicialización Random Selectiva [-0.01,0.01]	38	SI	173	78.10%	95.09%	89.13%
Inicialización Random Selectiva [-0.01,0.01]	39	SI	224	78.51%	97.10%	90.58%
Inicialización Random Selectiva [-0.01,0.01]	40	SI	266	75.21%	96.43%	88.99%
Inicialización Random Selectiva [-0.01,0.01]	41	SI	226	76.03%	95.09%	88.41%
Inicialización Random Selectiva [-0.01,0.01]	42	SI	135	76.03%	95.54%	88.70%
Inicialización Random Selectiva [-0.01,0.01]	43	SI	188	78.93%	95.54%	89.71%
Inicialización Random Selectiva [-0.01,0.01]	44	SI	202	75.21%	95.76%	88.55%
Inicialización Random Selectiva [-0.01,0.01]	45	SI	309	79.34%	96.88%	90.72%
Inicialización Random Selectiva [-0.01,0.01]	46	SI	182	80.17%	95.76%	90.29%
Inicialización Random Selectiva [-0.01,0.01]	47	SI	187	78.51%	95.31%	89.42%
Inicialización Random Selectiva [-0.01,0.01]	48	SI	198	78.51%	95.98%	89.86%
Inicialización Random Selectiva [-0.01,0.01]	49	SI	228	76.45%	95.98%	89.13%
Inicialización Random Selectiva [-0.01,0.01]	50	SI	244	74.79%	96.65%	88.99%
Inicialización Random [-1,1]	1	SI	162	78.93%	96.21%	90.14%
Inicialización Random [-1,1]	2	SI	197	78.93%	95.09%	89.42%
Inicialización Random [-1,1]	3	SI	154	78.10%	96.65%	90.14%

Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random [-1,1]	4	SI	417	79.34%	96.43%	90.43%
Inicialización Random [-1,1]	5	SI	214	77.27%	95.54%	89.13%
Inicialización Random [-1,1]	6	SI	318	80.99%	97.32%	91.59%
Inicialización Random [-1,1]	7	SI	147	80.17%	96.43%	90.72%
Inicialización Random [-1,1]	8	SI	157	76.86%	95.98%	89.28%
Inicialización Random [-1,1]	9	SI	175	77.69%	96.65%	90.00%
Inicialización Random [-1,1]	10	SI	340	76.86%	96.43%	89.57%
Inicialización Random [-1,1]	11	SI	270	76.45%	96.21%	89.28%
Inicialización Random [-1,1]	12	SI	281	79.34%	96.65%	90.58%
Inicialización Random [-1,1]	13	SI	169	76.86%	96.21%	89.42%
Inicialización Random [-1,1]	14	SI	239	80.17%	97.10%	91.16%
Inicialización Random [-1,1]	15	SI	191	80.58%	96.43%	90.87%
Inicialización Random [-1,1]	16	SI	239	76.86%	96.88%	89.86%
Inicialización Random [-1,1]	17	SI	216	79.34%	96.21%	90.29%
Inicialización Random [-1,1]	18	SI	255	79.75%	97.10%	91.01%
Inicialización Random [-1,1]	19	SI	302	78.93%	97.32%	90.87%
Inicialización Random [-1,1]	20	SI	230	79.34%	95.98%	90.14%
Inicialización Random [-1,1]	21	SI	485	78.10%	92.86%	87.68%
Inicialización Random [-1,1]	22	SI	154	75.62%	95.76%	88.70%
Inicialización Random [-1,1]	23	SI	163	79.75%	96.43%	90.58%
Inicialización Random [-1,1]	24	SI	166	79.34%	94.87%	89.42%
Inicialización Random [-1,1]	25	SI	269	79.75%	96.88%	90.87%
Inicialización Random [-1,1]	26	SI	232	80.58%	95.98%	90.58%
Inicialización Random [-1,1]	27	SI	152	79.75%	97.32%	91.16%
Inicialización Random [-1,1]	28	SI	174	80.17%	96.65%	90.87%
Inicialización Random [-1,1]	29	SI	265	80.17%	96.88%	91.01%
Inicialización Random [-1,1]	30	SI	138	78.93%	96.65%	90.43%

Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random [-1,1]	31	SI	151	78.51%	96.65%	90.29%
Inicialización Random [-1,1]	32	SI	121	78.51%	95.09%	89.28%
Inicialización Random [-1,1]	33	SI	184	76.86%	95.76%	89.13%
Inicialización Random [-1,1]	34	SI	163	77.27%	94.64%	88.55%
Inicialización Random [-1,1]	35	SI	162	79.34%	96.65%	90.58%
Inicialización Random [-1,1]	36	SI	138	76.45%	96.43%	89.42%
Inicialización Random [-1,1]	37	SI	342	75.62%	95.31%	88.41%
Inicialización Random [-1,1]	38	SI	209	77.69%	96.21%	89.71%
Inicialización Random [-1,1]	39	SI	151	81.41%	96.88%	91.45%
Inicialización Random [-1,1]	40	SI	137	76.03%	95.76%	88.84%
Inicialización Random [-1,1]	41	SI	197	78.51%	96.43%	90.14%
Inicialización Random [-1,1]	42	SI	171	79.75%	96.88%	90.87%
Inicialización Random [-1,1]	43	SI	163	75.62%	96.21%	88.99%
Inicialización Random [-1,1]	44	SI	144	78.93%	97.10%	90.72%
Inicialización Random [-1,1]	45	SI	150	78.51%	95.54%	89.57%
Inicialización Random [-1,1]	46	SI	229	76.86%	96.21%	89.42%
Inicialización Random [-1,1]	47	SI	205	76.03%	96.43%	89.28%
Inicialización Random [-1,1]	48	SI	166	78.10%	96.21%	89.86%
Inicialización Random [-1,1]	49	SI	168	79.75%	96.88%	90.87%
Inicialización Random [-1,1]	50	SI	211	76.86%	96.21%	89.42%
Inicialización Random [-0.01,0.01]	1	SI	207	78.93%	95.98%	90.00%
Inicialización Random [-0.01,0.01]	2	SI	236	75.62%	93.97%	87.54%
Inicialización Random [-0.01,0.01]	3	SI	227	81.82%	95.98%	91.01%
Inicialización Random [-0.01,0.01]	4	SI	254	74.79%	95.09%	87.97%
Inicialización Random [-0.01,0.01]	5	SI	302	78.10%	96.43%	90.00%
Inicialización Random [-0.01,0.01]	6	SI	379	77.27%	96.88%	90.00%
Inicialización Random [-0.01,0.01]	7	SI	240	76.86%	95.98%	89.28%

Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random [-0.01,0.01]	8	SI	215	79.34%	96.65%	90.58%
Inicialización Random [-0.01,0.01]	9	SI	222	78.10%	96.21%	89.86%
Inicialización Random [-0.01,0.01]	10	SI	198	78.10%	96.21%	89.86%
Inicialización Random [-0.01,0.01]	11	SI	185	78.10%	96.43%	90.00%
Inicialización Random [-0.01,0.01]	12	SI	184	80.17%	95.54%	90.14%
Inicialización Random [-0.01,0.01]	13	SI	246	75.62%	95.98%	88.84%
Inicialización Random [-0.01,0.01]	14	SI	204	79.34%	96.65%	90.58%
Inicialización Random [-0.01,0.01]	15	SI	161	75.62%	95.09%	88.26%
Inicialización Random [-0.01,0.01]	16	SI	155	77.27%	96.43%	89.71%
Inicialización Random [-0.01,0.01]	17	SI	323	81.82%	96.65%	91.45%
Inicialización Random [-0.01,0.01]	18	SI	202	78.93%	96.43%	90.29%
Inicialización Random [-0.01,0.01]	19	SI	280	79.75%	97.10%	91.01%
Inicialización Random [-0.01,0.01]	20	SI	186	73.97%	95.98%	88.26%
Inicialización Random [-0.01,0.01]	21	SI	227	78.93%	95.31%	89.57%
Inicialización Random [-0.01,0.01]	22	SI	461	73.97%	96.43%	88.55%
Inicialización Random [-0.01,0.01]	23	SI	215	78.10%	96.88%	90.29%
Inicialización Random [-0.01,0.01]	24	SI	273	80.58%	96.88%	91.16%
Inicialización Random [-0.01,0.01]	25	SI	150	74.79%	96.21%	88.70%
Inicialización Random [-0.01,0.01]	26	SI	244	78.51%	95.54%	89.57%
Inicialización Random [-0.01,0.01]	27	SI	240	80.58%	96.65%	91.01%
Inicialización Random [-0.01,0.01]	28	SI	233	78.51%	96.65%	90.29%
Inicialización Random [-0.01,0.01]	29	SI	214	77.69%	94.87%	88.84%
Inicialización Random [-0.01,0.01]	30	SI	185	76.86%	95.76%	89.13%
Inicialización Random [-0.01,0.01]	31	SI	211	79.34%	96.21%	90.29%
Inicialización Random [-0.01,0.01]	32	SI	377	79.34%	97.10%	90.87%
Inicialización Random [-0.01,0.01]	33	SI	453	77.27%	96.43%	89.71%
Inicialización Random [-0.01,0.01]	34	SI	232	79.75%	96.21%	90.43%



Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random [-0.01,0.01]	35	SI	236	73.55%	95.98%	88.12%
Inicialización Random [-0.01,0.01]	36	SI	315	76.45%	95.54%	88.84%
Inicialización Random [-0.01,0.01]	37	SI	190	78.10%	96.88%	90.29%
Inicialización Random [-0.01,0.01]	38	SI	259	76.03%	96.65%	89.42%
Inicialización Random [-0.01,0.01]	39	SI	172	76.86%	96.65%	89.71%
Inicialización Random [-0.01,0.01]	40	SI	194	78.51%	96.65%	90.29%
Inicialización Random [-0.01,0.01]	41	SI	179	77.69%	96.65%	90.00%
Inicialización Random [-0.01,0.01]	42	SI	324	76.86%	96.65%	89.71%
Inicialización Random [-0.01,0.01]	43	SI	424	77.69%	96.88%	90.14%
Inicialización Random [-0.01,0.01]	44	SI	224	76.45%	96.88%	89.71%
Inicialización Random [-0.01,0.01]	45	SI	235	78.10%	96.88%	90.29%
Inicialización Random [-0.01,0.01]	46	SI	261	76.03%	95.31%	88.55%
Inicialización Random [-0.01,0.01]	47	SI	351	78.51%	96.21%	90.00%
Inicialización Random [-0.01,0.01]	48	SI	262	78.51%	96.43%	90.14%
Inicialización Random [-0.01,0.01]	49	SI	205	79.75%	95.98%	90.29%
Inicialización Random [-0.01,0.01]	50	SI	299	78.51%	96.65%	90.29%
Inicialización Random Selectiva [-0.1,0.1]	1	SI	237	75.21%	95.98%	88.70%
Inicialización Random Selectiva [-0.1,0.1]	2	SI	290	79.75%	96.88%	90.87%
Inicialización Random Selectiva [-0.1,0.1]	3	SI	299	80.17%	95.98%	90.43%
Inicialización Random Selectiva [-0.1,0.1]	4	SI	154	78.10%	95.98%	89.71%
Inicialización Random Selectiva [-0.1,0.1]	5	SI	239	80.58%	95.76%	90.43%
Inicialización Random Selectiva [-0.1,0.1]	6	SI	222	79.75%	96.43%	90.58%
Inicialización Random Selectiva [-0.1,0.1]	7	SI	159	75.21%	95.54%	88.41%
Inicialización Random Selectiva [-0.1,0.1]	8	SI	202	78.51%	95.98%	89.86%
Inicialización Random Selectiva [-0.1,0.1]	9	SI	154	78.10%	95.98%	89.71%
Inicialización Random Selectiva [-0.1,0.1]	10	SI	238	79.75%	97.10%	91.01%
Inicialización Random Selectiva [-0.1,0.1]	11	SI	180	78.93%	95.31%	89.57%

Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random Selectiva [-0.1,0.1]	12	SI	162	76.03%	95.09%	88.41%
Inicialización Random Selectiva [-0.1,0.1]	13	SI	145	76.03%	96.88%	89.57%
Inicialización Random Selectiva [-0.1,0.1]	14	SI	175	76.45%	96.21%	89.28%
Inicialización Random Selectiva [-0.1,0.1]	15	SI	207	78.93%	96.21%	90.14%
Inicialización Random Selectiva [-0.1,0.1]	16	SI	197	75.21%	96.21%	88.84%
Inicialización Random Selectiva [-0.1,0.1]	17	SI	139	80.17%	96.43%	90.72%
Inicialización Random Selectiva [-0.1,0.1]	18	SI	209	76.45%	95.54%	88.84%
Inicialización Random Selectiva [-0.1,0.1]	19	SI	172	81.41%	97.10%	91.59%
Inicialización Random Selectiva [-0.1,0.1]	20	SI	222	76.03%	96.43%	89.28%
Inicialización Random Selectiva [-0.1,0.1]	21	SI	164	78.10%	95.31%	89.28%
Inicialización Random Selectiva [-0.1,0.1]	22	SI	217	77.69%	96.43%	89.86%
Inicialización Random Selectiva [-0.1,0.1]	23	SI	231	76.45%	96.43%	89.42%
Inicialización Random Selectiva [-0.1,0.1]	24	SI	256	79.34%	96.65%	90.58%
Inicialización Random Selectiva [-0.1,0.1]	25	SI	208	78.93%	95.76%	89.86%
Inicialización Random Selectiva [-0.1,0.1]	26	SI	160	76.86%	95.09%	88.70%
Inicialización Random Selectiva [-0.1,0.1]	27	SI	329	76.45%	95.98%	89.13%
Inicialización Random Selectiva [-0.1,0.1]	28	SI	273	75.62%	95.09%	88.26%
Inicialización Random Selectiva [-0.1,0.1]	29	SI	183	80.99%	96.88%	91.30%
Inicialización Random Selectiva [-0.1,0.1]	30	SI	322	73.14%	95.76%	87.83%
Inicialización Random Selectiva [-0.1,0.1]	31	SI	191	77.27%	96.43%	89.71%
Inicialización Random Selectiva [-0.1,0.1]	32	SI	182	76.86%	95.09%	88.70%
Inicialización Random Selectiva [-0.1,0.1]	33	SI	185	78.51%	96.43%	90.14%
Inicialización Random Selectiva [-0.1,0.1]	34	SI	252	76.45%	95.31%	88.70%
Inicialización Random Selectiva [-0.1,0.1]	35	SI	303	78.51%	95.98%	89.86%
Inicialización Random Selectiva [-0.1,0.1]	36	SI	345	78.93%	95.98%	90.00%
Inicialización Random Selectiva [-0.1,0.1]	37	SI	187	80.58%	96.43%	90.87%
Inicialización Random Selectiva [-0.1,0.1]	38	SI	278	78.51%	95.76%	89.71%

Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random Selectiva [-0.1,0.1]	39	SI	194	78.51%	96.43%	90.14%
Inicialización Random Selectiva [-0.1,0.1]	40	SI	186	77.27%	95.09%	88.84%
Inicialización Random Selectiva [-0.1,0.1]	41	SI	277	78.93%	96.65%	90.43%
Inicialización Random Selectiva [-0.1,0.1]	42	SI	197	78.10%	96.43%	90.00%
Inicialización Random Selectiva [-0.1,0.1]	43	SI	254	76.03%	95.54%	88.70%
Inicialización Random Selectiva [-0.1,0.1]	44	SI	205	81.82%	96.21%	91.16%
Inicialización Random Selectiva [-0.1,0.1]	45	SI	278	80.58%	96.43%	90.87%
Inicialización Random Selectiva [-0.1,0.1]	46	SI	202	79.75%	96.21%	90.43%
Inicialización Random Selectiva [-0.1,0.1]	47	SI	252	80.17%	95.98%	90.43%
Inicialización Random Selectiva [-0.1,0.1]	48	SI	239	78.51%	96.88%	90.43%
Inicialización Random Selectiva [-0.1,0.1]	49	SI	267	75.62%	97.10%	89.57%
Inicialización Random Selectiva [-0.1,0.1]	50	SI	287	80.17%	95.98%	90.43%
Inicialización Random Selectiva [-5,5]	1	SI	183	78.10%	95.31%	89.28%
Inicialización Random Selectiva [-5,5]	2	SI	550	77.27%	96.88%	90.00%
Inicialización Random Selectiva [-5,5]	3	SI	255	75.21%	96.43%	88.99%
Inicialización Random Selectiva [-5,5]	4	NO	775	78.93%	97.32%	90.87%
Inicialización Random Selectiva [-5,5]	5	SI	311	76.45%	95.98%	89.13%
Inicialización Random Selectiva [-5,5]	6	SI	219	78.93%	96.88%	90.58%
Inicialización Random Selectiva [-5,5]	7	SI	252	78.93%	96.43%	90.29%
Inicialización Random Selectiva [-5,5]	8	NO	775	79.75%	95.98%	90.29%
Inicialización Random Selectiva [-5,5]	9	SI	362	76.45%	96.21%	89.28%
Inicialización Random Selectiva [-5,5]	10	NO	774	78.51%	89.73%	85.80%
Inicialización Random Selectiva [-5,5]	11	SI	236	78.51%	95.98%	89.86%
Inicialización Random Selectiva [-5,5]	12	SI	302	76.45%	96.21%	89.28%
Inicialización Random Selectiva [-5,5]	13	SI	335	75.21%	97.32%	89.57%
Inicialización Random Selectiva [-5,5]	14	SI	441	82.23%	96.88%	91.74%
Inicialización Random Selectiva [-5,5]	15	SI	159	77.69%	96.65%	90.00%

Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random Selectiva [-5,5]	16	SI	333	79.75%	97.32%	91.16%
Inicialización Random Selectiva [-5,5]	17	NO	775	81.82%	96.43%	91.30%
Inicialización Random Selectiva [-5,5]	18	NO	774	77.69%	96.65%	90.00%
Inicialización Random Selectiva [-5,5]	19	NO	773	81.41%	96.43%	91.16%
Inicialización Random Selectiva [-5,5]	20	NO	776	80.58%	96.65%	91.01%
Inicialización Random Selectiva [-5,5]	21	NO	774	82.64%	96.21%	91.45%
Inicialización Random Selectiva [-5,5]	22	SI	234	79.34%	95.09%	89.57%
Inicialización Random Selectiva [-5,5]	23	SI	433	77.27%	96.65%	89.86%
Inicialización Random Selectiva [-5,5]	24	SI	302	75.62%	97.10%	89.57%
Inicialización Random Selectiva [-5,5]	25	SI	362	80.17%	96.21%	90.58%
Inicialización Random Selectiva [-5,5]	26	SI	232	79.75%	95.98%	90.29%
Inicialización Random Selectiva [-5,5]	27	SI	145	78.93%	96.43%	90.29%
Inicialización Random Selectiva [-5,5]	28	SI	232	83.06%	96.21%	91.59%
Inicialización Random Selectiva [-5,5]	29	NO	774	80.17%	96.65%	90.87%
Inicialización Random Selectiva [-5,5]	30	SI	249	79.34%	97.10%	90.87%
Inicialización Random Selectiva [-5,5]	31	SI	551	78.51%	96.43%	90.14%
Inicialización Random Selectiva [-5,5]	32	SI	267	76.45%	95.98%	89.13%
Inicialización Random Selectiva [-5,5]	33	NO	774	80.99%	96.65%	91.16%
Inicialización Random Selectiva [-5,5]	34	SI	205	78.93%	96.88%	90.58%
Inicialización Random Selectiva [-5,5]	35	SI	716	76.03%	96.65%	89.42%
Inicialización Random Selectiva [-5,5]	36	SI	233	82.23%	96.43%	91.45%
Inicialización Random Selectiva [-5,5]	37	SI	360	74.79%	94.87%	87.83%
Inicialización Random Selectiva [-5,5]	38	NO	773	80.17%	95.31%	90.00%
Inicialización Random Selectiva [-5,5]	39	SI	607	77.27%	97.54%	90.43%
Inicialización Random Selectiva [-5,5]	40	SI	763	79.34%	96.65%	90.58%
Inicialización Random Selectiva [-5,5]	41	SI	159	79.75%	95.98%	90.29%
Inicialización Random Selectiva [-5,5]	42	SI	229	78.93%	94.64%	89.13%

Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random Selectiva [-5,5]	43	SI	710	80.17%	96.65%	90.87%
Inicialización Random Selectiva [-5,5]	44	SI	280	80.99%	95.98%	90.72%
Inicialización Random Selectiva [-5,5]	45	SI	469	75.21%	95.31%	88.26%
Inicialización Random Selectiva [-5,5]	46	SI	402	76.03%	95.76%	88.84%
Inicialización Random Selectiva [-5,5]	47	NO	776	75.62%	90.85%	85.51%
Inicialización Random Selectiva [-5,5]	48	SI	340	80.58%	97.32%	91.45%
Inicialización Random Selectiva [-5,5]	49	SI	162	78.51%	96.65%	90.29%
Inicialización Random Selectiva [-5,5]	50	SI	236	76.86%	95.54%	88.99%
Inicialización Random [-0.1,0.1]	1	SI	200	79.34%	96.43%	90.43%
Inicialización Random [-0.1,0.1]	2	SI	225	75.21%	93.97%	87.39%
Inicialización Random [-0.1,0.1]	3	SI	252	80.99%	96.65%	91.16%
Inicialización Random [-0.1,0.1]	4	SI	310	78.93%	95.76%	89.86%
Inicialización Random [-0.1,0.1]	5	SI	177	79.34%	96.88%	90.72%
Inicialización Random [-0.1,0.1]	6	SI	185	78.51%	96.43%	90.14%
Inicialización Random [-0.1,0.1]	7	SI	204	80.58%	96.88%	91.16%
Inicialización Random [-0.1,0.1]	8	SI	251	78.10%	96.21%	89.86%
Inicialización Random [-0.1,0.1]	9	SI	184	76.03%	96.43%	89.28%
Inicialización Random [-0.1,0.1]	10	SI	280	79.75%	96.88%	90.87%
Inicialización Random [-0.1,0.1]	11	SI	151	76.03%	94.64%	88.12%
Inicialización Random [-0.1,0.1]	12	SI	184	79.75%	95.54%	90.00%
Inicialización Random [-0.1,0.1]	13	SI	185	77.27%	95.54%	89.13%
Inicialización Random [-0.1,0.1]	14	SI	245	78.10%	97.32%	90.58%
Inicialización Random [-0.1,0.1]	15	SI	336	78.51%	95.98%	89.86%
Inicialización Random [-0.1,0.1]	16	SI	153	77.27%	95.31%	88.99%
Inicialización Random [-0.1,0.1]	17	SI	252	76.03%	95.31%	88.55%
Inicialización Random [-0.1,0.1]	18	SI	734	80.17%	96.88%	91.01%
Inicialización Random [-0.1,0.1]	19	SI	175	78.10%	95.98%	89.71%

Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random [-0.1,0.1]	20	SI	315	80.99%	96.65%	91.16%
Inicialización Random [-0.1,0.1]	21	SI	298	78.10%	96.43%	90.00%
Inicialización Random [-0.1,0.1]	22	SI	229	75.62%	94.87%	88.12%
Inicialización Random [-0.1,0.1]	23	SI	232	76.45%	95.98%	89.13%
Inicialización Random [-0.1,0.1]	24	SI	223	76.86%	95.98%	89.28%
Inicialización Random [-0.1,0.1]	25	SI	195	78.93%	96.43%	90.29%
Inicialización Random [-0.1,0.1]	26	SI	311	79.34%	95.31%	89.71%
Inicialización Random [-0.1,0.1]	27	SI	217	78.51%	96.21%	90.00%
Inicialización Random [-0.1,0.1]	28	SI	214	77.27%	95.98%	89.42%
Inicialización Random [-0.1,0.1]	29	SI	216	80.17%	96.88%	91.01%
Inicialización Random [-0.1,0.1]	30	SI	216	77.69%	95.98%	89.57%
Inicialización Random [-0.1,0.1]	31	SI	309	78.93%	95.09%	89.42%
Inicialización Random [-0.1,0.1]	32	SI	181	78.10%	96.21%	89.86%
Inicialización Random [-0.1,0.1]	33	SI	196	74.79%	95.54%	88.26%
Inicialización Random [-0.1,0.1]	34	SI	236	77.27%	93.75%	87.97%
Inicialización Random [-0.1,0.1]	35	SI	345	78.51%	97.10%	90.58%
Inicialización Random [-0.1,0.1]	36	SI	213	76.03%	96.65%	89.42%
Inicialización Random [-0.1,0.1]	37	SI	255	81.82%	95.76%	90.87%
Inicialización Random [-0.1,0.1]	38	SI	158	80.17%	94.87%	89.71%
Inicialización Random [-0.1,0.1]	39	SI	256	79.75%	96.88%	90.87%
Inicialización Random [-0.1,0.1]	40	SI	183	80.17%	96.65%	90.87%
Inicialización Random [-0.1,0.1]	41	SI	175	75.21%	95.09%	88.12%
Inicialización Random [-0.1,0.1]	42	SI	263	74.38%	95.09%	87.83%
Inicialización Random [-0.1,0.1]	43	SI	285	77.27%	96.43%	89.71%
Inicialización Random [-0.1,0.1]	44	SI	236	77.69%	96.88%	90.14%
Inicialización Random [-0.1,0.1]	45	SI	226	76.86%	96.65%	89.71%
Inicialización Random [-0.1,0.1]	46	SI	226	77.69%	96.43%	89.86%

Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random [-0.1,0.1]	47	SI	233	79.75%	96.65%	90.72%
Inicialización Random [-0.1,0.1]	48	SI	189	71.90%	94.87%	86.81%
Inicialización Random [-0.1,0.1]	49	SI	223	77.69%	96.43%	89.86%
Inicialización Random [-0.1,0.1]	50	SI	217	79.34%	96.88%	90.72%
Inicialización Random [-5,5]	1	NO	776	80.17%	95.98%	90.43%
Inicialización Random [-5,5]	2	NO	776	80.58%	94.64%	89.71%
Inicialización Random [-5,5]	3	NO	774	78.51%	94.64%	88.99%
Inicialización Random [-5,5]	4	NO	773	77.69%	92.63%	87.39%
Inicialización Random [-5,5]	5	NO	782	74.79%	93.97%	87.25%
Inicialización Random [-5,5]	6	NO	776	72.73%	96.43%	88.12%
Inicialización Random [-5,5]	7	NO	774	74.79%	95.98%	88.55%
Inicialización Random [-5,5]	8	NO	772	78.51%	93.30%	88.12%
Inicialización Random [-5,5]	9	NO	780	76.45%	95.54%	88.84%
Inicialización Random [-5,5]	10	NO	775	81.82%	96.65%	91.45%
Inicialización Random [-5,5]	11	NO	775	78.93%	94.87%	89.28%
Inicialización Random [-5,5]	12	NO	775	74.79%	94.87%	87.83%
Inicialización Random [-5,5]	13	NO	777	74.38%	90.85%	85.07%
Inicialización Random [-5,5]	14	NO	775	79.34%	93.08%	88.26%
Inicialización Random [-5,5]	15	NO	776	75.21%	93.30%	86.96%
Inicialización Random [-5,5]	16	NO	777	78.51%	96.21%	90.00%
Inicialización Random [-5,5]	17	NO	774	77.27%	96.88%	90.00%
Inicialización Random [-5,5]	18	NO	779	72.31%	87.72%	82.32%
Inicialización Random [-5,5]	19	NO	774	82.64%	93.97%	90.00%
Inicialización Random [-5,5]	20	NO	774	82.23%	96.65%	91.59%
Inicialización Random [-5,5]	21	NO	775	76.03%	96.43%	89.28%
Inicialización Random [-5,5]	22	NO	777	79.75%	97.10%	91.01%
Inicialización Random [-5,5]	23	NO	776	80.17%	95.98%	90.43%

Función Inicialización	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Inicialización Random [-5,5]	24	NO	775	77.27%	96.88%	90.00%
Inicialización Random [-5,5]	25	NO	774	78.51%	96.65%	90.29%
Inicialización Random [-5,5]	26	NO	779	76.45%	92.86%	87.10%
Inicialización Random [-5,5]	27	SI	644	82.23%	97.10%	91.88%
Inicialización Random [-5,5]	28	NO	779	80.99%	95.76%	90.58%
Inicialización Random [-5,5]	29	NO	773	78.93%	95.54%	89.71%
Inicialización Random [-5,5]	30	NO	774	78.51%	93.30%	88.12%
Inicialización Random [-5,5]	31	NO	775	79.75%	94.42%	89.28%
Inicialización Random [-5,5]	32	NO	780	78.51%	95.54%	89.57%
Inicialización Random [-5,5]	33	NO	776	79.75%	94.42%	89.28%
Inicialización Random [-5,5]	34	NO	775	79.34%	96.43%	90.43%
Inicialización Random [-5,5]	35	NO	775	73.97%	87.72%	82.90%
Inicialización Random [-5,5]	36	NO	777	79.34%	95.09%	89.57%
Inicialización Random [-5,5]	37	NO	775	75.62%	97.32%	89.71%
Inicialización Random [-5,5]	38	NO	780	76.45%	95.76%	88.99%
Inicialización Random [-5,5]	39	NO	775	75.62%	93.08%	86.96%
Inicialización Random [-5,5]	40	NO	774	78.93%	93.08%	88.12%
Inicialización Random [-5,5]	41	NO	775	80.99%	94.64%	89.86%
Inicialización Random [-5,5]	42	NO	773	79.34%	95.31%	89.71%
Inicialización Random [-5,5]	43	NO	775	79.34%	97.32%	91.01%
Inicialización Random [-5,5]	44	SI	641	78.10%	97.32%	90.58%
Inicialización Random [-5,5]	45	NO	780	79.75%	94.87%	89.57%
Inicialización Random [-5,5]	46	NO	776	78.93%	95.31%	89.57%
Inicialización Random [-5,5]	47	NO	777	79.34%	95.54%	89.86%
Inicialización Random [-5,5]	48	NO	774	76.45%	96.21%	89.28%
Inicialización Random [-5,5]	49	NO	774	76.86%	96.21%	89.42%
Inicialización Random [-5,5]	50	NO	773	77.69%	95.98%	89.57%



**A.7. Comparación de ambos métodos**

Tipo Algoritmo Entrenamiento	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Algoritmo genético	1	SI	711	83.33%	95.29%	92.90%
Algoritmo genético	2	SI	815	83.33%	95.47%	93.04%
Algoritmo genético	3	SI	498	86.96%	94.38%	92.90%
Algoritmo genético	4	SI	271	78.99%	91.49%	88.99%
Algoritmo genético	5	SI	351	79.71%	92.21%	89.71%
Algoritmo genético	6	SI	737	84.78%	95.47%	93.33%
Algoritmo genético	7	SI	301	80.43%	92.93%	90.43%
Algoritmo genético	8	SI	287	81.16%	92.75%	90.43%
Algoritmo genético	9	SI	316	84.06%	94.57%	92.46%
Algoritmo genético	10	SI	474	82.61%	94.75%	92.32%
Algoritmo genético	11	SI	478	78.26%	94.20%	91.01%
Algoritmo genético	12	SI	291	84.06%	94.02%	92.03%
Algoritmo genético	13	SI	508	80.43%	94.20%	91.45%
Algoritmo genético	14	SI	389	84.78%	95.11%	93.04%
Algoritmo genético	15	SI	485	84.06%	94.57%	92.46%
Algoritmo genético	16	SI	468	84.06%	94.57%	92.46%
Algoritmo genético	17	SI	327	84.78%	94.75%	92.75%
Algoritmo genético	18	SI	275	84.06%	94.57%	92.46%
Algoritmo genético	19	SI	458	85.51%	95.47%	93.48%
Algoritmo genético	20	SI	294	82.61%	93.30%	91.16%
Algoritmo genético	21	SI	318	84.78%	93.12%	91.45%
Algoritmo genético	22	SI	365	83.33%	93.84%	91.74%
Algoritmo genético	23	SI	352	78.99%	94.38%	91.30%
Algoritmo genético	24	SI	318	82.61%	94.20%	91.88%
Algoritmo genético	25	SI	260	87.68%	93.84%	92.61%

Tipo Algoritmo Entrenamiento	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Algoritmo genético	26	SI	556	80.43%	93.66%	91.01%
Algoritmo genético	27	SI	637	84.06%	95.29%	93.04%
Algoritmo genético	28	SI	819	85.51%	95.65%	93.62%
Algoritmo genético	29	SI	408	84.06%	93.12%	91.30%
Algoritmo genético	30	SI	207	81.88%	94.57%	92.03%
Algoritmo genético	31	SI	460	81.16%	94.02%	91.45%
Algoritmo genético	32	SI	524	83.33%	92.39%	90.58%
Algoritmo genético	33	SI	424	76.09%	92.75%	89.42%
Algoritmo genético	34	SI	275	80.43%	94.38%	91.59%
Algoritmo genético	35	SI	403	79.71%	93.48%	90.72%
Algoritmo genético	36	SI	279	80.43%	93.84%	91.16%
Algoritmo genético	37	SI	313	82.61%	93.84%	91.59%
Algoritmo genético	38	SI	248	84.78%	94.20%	92.32%
Algoritmo genético	39	SI	275	83.33%	92.93%	91.01%
Algoritmo genético	40	SI	392	78.99%	94.93%	91.74%
Algoritmo genético	41	SI	454	84.06%	94.93%	92.75%
Algoritmo genético	42	SI	226	84.78%	94.02%	92.17%
Algoritmo genético	43	SI	339	85.51%	94.20%	92.46%
Algoritmo genético	44	SI	888	81.88%	93.84%	91.45%
Algoritmo genético	45	SI	436	83.33%	94.20%	92.03%
Algoritmo genético	46	SI	338	81.88%	93.48%	91.16%
Algoritmo genético	47	SI	352	84.78%	95.29%	93.19%
Algoritmo genético	48	SI	410	79.71%	94.02%	91.16%
Algoritmo genético	49	SI	274	81.88%	93.30%	91.01%
Algoritmo genético	50	SI	599	84.78%	94.57%	92.61%
Red Neuronal	1	SI	568	84.06%	93.48%	91.59%
Red Neuronal	2	SI	520	81.88%	92.93%	90.72%

Tipo Algoritmo Entrenamiento	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Red Neuronal	3	SI	571	84.06%	93.66%	91.74%
Red Neuronal	4	SI	573	84.78%	94.75%	92.75%
Red Neuronal	5	SI	531	84.78%	95.29%	93.19%
Red Neuronal	6	SI	894	81.16%	94.75%	92.03%
Red Neuronal	7	SI	686	84.78%	95.47%	93.33%
Red Neuronal	8	SI	481	84.06%	94.38%	92.32%
Red Neuronal	9	SI	528	84.06%	94.02%	92.03%
Red Neuronal	10	SI	704	78.26%	95.11%	91.74%
Red Neuronal	11	SI	791	84.78%	94.57%	92.61%
Red Neuronal	12	SI	436	84.78%	94.57%	92.61%
Red Neuronal	13	SI	703	79.71%	94.20%	91.30%
Red Neuronal	14	SI	883	78.99%	93.30%	90.43%
Red Neuronal	15	SI	527	85.51%	94.75%	92.90%
Red Neuronal	16	SI	482	81.88%	94.57%	92.03%
Red Neuronal	17	SI	450	84.06%	94.57%	92.46%
Red Neuronal	18	SI	571	84.78%	94.93%	92.90%
Red Neuronal	19	SI	763	85.51%	94.57%	92.75%
Red Neuronal	20	SI	366	86.96%	94.93%	93.33%
Red Neuronal	21	SI	459	82.61%	94.93%	92.46%
Red Neuronal	22	SI	448	76.81%	92.93%	89.71%
Red Neuronal	23	SI	881	84.78%	94.57%	92.61%
Red Neuronal	24	SI	763	83.33%	94.20%	92.03%
Red Neuronal	25	SI	395	84.06%	94.02%	92.03%
Red Neuronal	26	SI	482	84.06%	94.75%	92.61%
Red Neuronal	27	SI	392	83.33%	94.93%	92.61%
Red Neuronal	28	SI	1,192	81.88%	92.93%	90.72%
Red Neuronal	29	SI	378	82.61%	94.57%	92.17%

Tipo Algoritmo Entrenamiento	Corrida	Ent?	Tiempo Prom. Ent.	Aciertos Prom. Eval.	Aciertos Prom. Ent.	Aciertos Prom. Total
Red Neuronal	30	SI	683	85.51%	94.93%	93.04%
Red Neuronal	31	NO	2,917	84.78%	95.47%	93.33%
Red Neuronal	32	SI	474	84.78%	93.66%	91.88%
Red Neuronal	33	SI	338	84.78%	94.02%	92.17%
Red Neuronal	34	SI	841	80.43%	95.29%	92.32%
Red Neuronal	35	SI	682	85.51%	95.29%	93.33%
Red Neuronal	36	SI	410	81.88%	92.21%	90.14%
Red Neuronal	37	SI	1,560	84.06%	93.48%	91.59%
Red Neuronal	38	SI	571	80.43%	93.84%	91.16%
Red Neuronal	39	SI	575	82.61%	94.38%	92.03%
Red Neuronal	40	SI	463	83.33%	94.75%	92.46%
Red Neuronal	41	SI	633	83.33%	94.75%	92.46%
Red Neuronal	42	SI	1,046	81.88%	93.84%	91.45%
Red Neuronal	43	SI	444	84.06%	94.75%	92.61%
Red Neuronal	44	SI	663	84.06%	92.57%	90.87%
Red Neuronal	45	SI	439	84.06%	95.29%	93.04%
Red Neuronal	46	SI	927	85.51%	94.93%	93.04%
Red Neuronal	47	SI	489	83.33%	92.57%	90.72%
Red Neuronal	48	SI	656	84.06%	93.84%	91.88%
Red Neuronal	49	SI	1,531	81.88%	95.65%	92.90%
Red Neuronal	50	SI	570	81.88%	93.66%	91.30%

## **Apéndice B: Metodología de software**

El presente anexo tiene como objetivo describir el proceso de ingeniería de software aplicado en este trabajo para obtener los resultados y conclusiones que se expusieron en los capítulos 7 y 8.

En la sección B.1 se describen los requerimientos en los cuales se ha basado el desarrollo de este software.

En la sección B.2 se presenta el resultado del análisis de los requerimientos a través de casos de uso.

En la sección B.3 se presenta el diseño de la solución detallando los programas realizados, tomando como base lo expuesto en la sección de análisis y adjuntando las estructuras de datos y las estructuras de los archivos generados por estos programas.

En la sección B.4 se muestra la pantalla del sistema y se describen los comandos implementados mediante los cuales el usuario interactúa con éste. También se presentan cada una de las pruebas realizadas para verificar que el software entrega los resultados esperados.

Finalmente, en la sección B.5 se detallan algunas tareas de mantenimiento y se hace una breve reseña sobre la gestión de configuración del software.

### ***B.1. Requerimientos***

Como objetivo principal del trabajo se ha determinado que el sistema debe poder entrenar redes neuronales con conexión hacia delante. Además deberá dar al usuario la posibilidad de decidir qué algoritmo de entrenamiento utilizar. Finalmente deberá generar estadísticas detalladas sobre el proceso de entrenamiento que permitan realizar comparaciones entre ambos métodos.

### ***B.2. Análisis***

A continuación se presentan los casos de uso [Booch et al., 2000] más relevantes obtenidos durante el análisis del software. Los mismos corresponden a la interacción con el usuario del sistema, quien selecciona las condiciones en que se realizan los distintos experimentos.

#### **B.2.1. Seleccionar topología**

En este caso de uso, como se observa en la Figura B.1, el actor que inicia el proceso es el usuario del sistema, quien decide la topología de la red que se utilizarán durante los próximos experimentos.



**Figura B.1.** Seleccionar topología

### **B.2.2. Seleccionar tipo de entrenamiento**

También en este caso de uso, el actor que inicia el proceso es el usuario del sistema, quien selecciona uno de los posibles algoritmos de entrenamiento para utilizar en los próximos experimentos. También selecciona la cota máxima de error admisible. Esto se observa en la Figura B. 2.



**Figura B.2.** Seleccionar tipo de entrenamiento

### **B.2.3. Configurar algoritmo de entrenamiento**

Nuevamente en este caso de uso, el actor que inicia el proceso es el usuario del sistema, como se muestra en la Figura B.3. El usuario del sistema selecciona los distintos parámetros con los que desea ejecutar el algoritmo de entrenamiento, los cuales variarán de acuerdo al algoritmo de entrenamiento seleccionado en el caso de uso anterior.



**Figura B.3.** Configurar algoritmo de entrenamiento

### **B.2.4. Cargar ejemplos de entrenamiento**

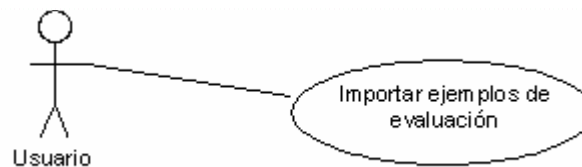
También en este caso de uso, el actor que inicia el proceso es el usuario del sistema, quien dará de alta el conjunto de ejemplos de entrenamiento que se utilizarán para entrenar la red neuronal. Para ello dispone de dos opciones, dar de alta manualmente cada uno de los ejemplos o importar el listado de ejemplos desde un archivo. La Figura B.4 muestra gráficamente lo dicho anteriormente.



**Figura B.4.** Cargar ejemplos de entrenamiento

### B.2.5. Cargar ejemplos de evaluación

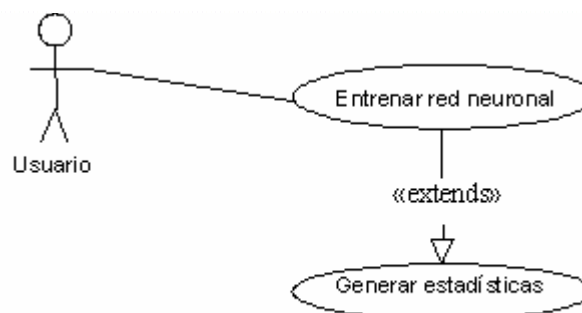
En este caso de uso, el actor que inicia el proceso es el usuario del sistema, quien dará de alta el conjunto de ejemplos de evaluación que se utilizarán para evaluar la calidad de la red neuronal una vez entrenada. Para ello dispone de dos opciones, dar de alta manualmente cada uno de los ejemplos o importar el listado de ejemplos desde un archivo. Esta situación se representa gráficamente en la Figura B.5.



**Figura B.5.** Cargar ejemplos de evaluación

### B.2.6. Entrenar red neuronal

Finalmente, como se muestra en la Figura B.6, en este caso de uso el actor (usuario del sistema) inicia el proceso de entrenamiento de la red neuronal. Dicho entrenamiento se realizará utilizando el método de entrenamiento seleccionado mediante los casos de uso “Seleccionar tipo de entrenamiento” y “Configurar algoritmo de entrenamiento”. A su vez, se generan las estadísticas sobre el desempeño del método a lo largo del proceso.



**Figura B.6.** Entrenar Red Neuronal

### **B.3. Diseño**

En esta sección se describen las pautas seguidas para el diseño del sistema de manera tal de cumplir con las funcionalidades establecidas en la etapa de análisis.

#### **B.3.1. Plataforma**

El sistema se ha implementado utilizando el lenguaje de programación C++. El programa utilizado para realizar todo el desarrollo del sistema fue Visual C++ 6.0.

Para desarrollar el sistema y realizar los experimentos, se seleccionó un equipo Intel Pentium IV, 2.8 GHz con 512 MB de memoria corriendo un Sistema Operativo Windows 2000 SP4.

##### **B.3.1.1. Estructura de directorios**

El sistema se desarrollo utilizando la siguiente estructura de directorios, la cual deberá ser respetada para instalar y compilar los archivos fuentes.

- Tesis
  - Debug
  - MySources
  - Release
  - Res

##### **B.3.1.2. Instalación de los archivos fuente del sistema**

El sistema desarrollado para esta tesis se encuentra distribuido en varios archivos fuente. Cada uno de ellos implementa un módulo particular. Para su compilación se deben instalar los siguientes archivos y directorios, de acuerdo a lo indicado en la tabla B.1.

Directorio	Archivo
Tesis	Diálogos.cpp
	Diálogos.h
	EasyTrain.cpp
	EasyTrain.dsp
	EasyTrain.dsw
	EasyTrain.h
	EasyTrain.rc
	EasyTrain.reg
	EasyTrainDoc.cpp
	EasyTrainDoc.h
	EasyTrainView.cpp
	EasyTrainView.h
	MainFrm.cpp
	MainFrm.h
	resource.h



Directorio	Archivo
MySources	AlgoritmoEntrenamiento.cpp
	AlgoritmoEntrenamiento.h
	AlgoritmoGenetico.cpp
	AlgoritmoGenetico.h
	Capa.cpp
	Capa.h
	Cromosoma.cpp
	Cromosoma.h
	Cruza.cpp
	Cruza.h
	Ejemplo.cpp
	Ejemplo.h
	Evaluacion.cpp
	Evaluacion.h
	Include.cpp
	Include.h
	Inicializar.cpp
	Inicializar.h
	Lista.cpp
	Lista.h
	ListaCromosomas.cpp
	ListaCromosomas.h
	ListaEjemplos.cpp
	ListaEjemplos.h
	Mutacion.cpp
	Mutacion.h
	Objetos.h
	Rrnn.cpp
	Rrnn.h
	Seleccion.cpp
	Seleccion.h
	StdAfx.cpp
	StdAfx.h
Res	EasyTrain.ico
	EasyTrainDoc.ico
	Toolbar.bmp

**Tabla B.1.** Estructura de archivos

Los directorios Debug y Release inicialmente se encuentran vacíos, y son utilizados para guardar archivos intermedios durante el proceso de compilación de los archivos fuente.

### B.3.2. Archivos y bloques constitutivos del sistema

El sistema está formado por varias capas. Esta división en capas o módulos se realizó para facilitar el desarrollo del sistema. Se utilizó un enfoque

ascendente, desarrollando inicialmente capas con funcionalidades básicas, las cuales se integran en capas superiores para cumplir con las funcionalidades necesarias. Cada uno de estos módulos está contenido en un archivo fuente individual, que se suma a los archivos que se generan automáticamente en todo proyecto de Visual C++. El paradigma de programación utilizado fue la programación orientada a objetos.

### **B.3.2.1. Archivos generados por Visual C++**

El siguiente conjunto de archivos se genera automáticamente durante la generación de un nuevo proyecto. Estos archivos se modificaron a través de la interfaz gráfica (GUI) del sistema así como también modificando el código fuente de los mismos.

Las funciones principales de estos archivos están relacionadas con la interacción entre el usuario y el sistema. Dado que las clases que implementan no están directamente relacionadas con el objetivo de esta tesis, se explican brevemente la función de cada archivo y cuando se considere necesario se comentarán los métodos más importantes. A continuación se hace una referencia a las mismas:

- **Diálogos.cpp:** En este archivo se implementan las clases relacionadas con los cuadros de diálogos de la aplicación. En cada clase se implementa el comportamiento de un cuadro de diálogo frente a cada una de las acciones del usuario. Los cuadros de diálogo del sistema permiten al usuario el ingreso de datos y la configuración de los métodos de entrenamiento.
- **Diálogos.h:** Es en este archivo donde se definen las clases relacionadas con cada uno de los cuadros de diálogo que se implementan en el sistema. En esta definición se declara para cada clase cual es la definición gráfica que se utilizará (y que se define en el archivo EasyTrain.rc)
- **EasyTrain.cpp:** Incluye la implementación de la clase *CEasyTrainApp*. El método más importante, *InitInstance*, contiene la inicialización de la aplicación.
- **EasyTrain.dsp:** Este archivo define el proyecto en Visual C++. Contiene la estructura de archivos y carpetas utilizadas en el proyecto.
- **EasyTrain.dsw:** Este archivo define el espacio de trabajo (Workspace) donde se encontrará el proyecto. El espacio de trabajo es el objeto de mayor nivel en Visual C++ y permite agrupar uno o más proyectos relacionados en un ambiente común.
- **EasyTrain.h:** Se realiza la definición de la clase *CEasyTrainApp*. Esta clase es muy importante ya que hereda las características de la clase *CWinApp* (define el funcionamiento básico de una aplicación en el entorno Windows)

- **EasyTrain.rc:** Este archivo contiene la definición gráfica de cada uno de los objetos que hacen a la GUI del sistema. Se definen los cuadros de diálogo, las barras de herramienta, etc.
- **EasyTrain.reg:** Este archivo se utiliza para definir las entradas que se ingresan al registro de Windows. En este caso en particular las entradas que se ingresarán al registro definen la extensión de archivos default y permiten abrir archivos mediante doble clicks.
- **EasyTrainDoc.cpp:** En este archivo se implementa la clase *CEasyTrainDoc*. Entre sus métodos más importantes se encuentran la serialización de documentos (guardar y abrir documentos) y la generación de nuevos documentos.
- **EasyTrainDoc.h:** En este archivo se define la clase *CEasyTrainDoc*. Esta clase será la encargada de la manipulación de los datos de la aplicación.
- **EasyTrainView.cpp:** En este archivo se realiza la implementación de la clase *CEasyTrainView*. Su función principal es desplegar gráficamente la información en pantalla (muestra el conjunto de ejemplos, la representación gráfica de la topología de la red neuronal, etc.). Otros métodos muy importantes definidos en esta clase son los referentes al manejo de los mensajes generados a través del menú. Para cada una de las opciones del menú se generó un método encargado de procesar el mensaje.
- **EasyTrainView.h:** Este archivo contiene la definición de la clase *CEasyTrainView*. Esta clase se utiliza para el despliegue de datos en pantalla. A su vez se definen en esta clase los manejadores de menú.
- **MainFrm.cpp:** Contiene la implementación de la clase *CMainFrame*. El método más importante es *OnCreate* donde se definen las características de la ventana principal.
- **MainFrm.h:** Contiene la definición de la clase *CMainFrame*. Esta clase define el marco donde se dibuja la ventana principal del sistema.
- **Resource.h:** En este archivo se encuentran todas las constantes referentes a la GUI del sistema desarrollado. Se definen los identificadores de los objetos gráficos del sistema.
- **StdAfx.h:** Archivo de cabeceras. Permite incluir cabeceras comunes.
- **StdAfx.cpp:** Incluye al archivo *StdAfx.h* para poder utilizar todas las cabeceras precompiladas de Visual C++.

### **B.3.2.2. Archivos generados para este trabajo**

El siguiente conjunto de archivos definen las funcionalidades más importantes desarrolladas en este trabajo, y se describen a continuación:

- **AlgoritmoEntrenamiento.cpp:** En este archivo se realiza el desarrollo de la clase *CAgoritmoEntrenamiento*, utilizando el soporte provisto por las clases *CRRNN* y *CAgoritmoGenetico*. El desarrollo de este conjunto de métodos es sencillo, ya que gran parte de la funcionalidad es provista por las clases de menor nivel, y que se describen a continuación.
- **AlgoritmoEntrenamiento.h:** El objetivo de la clase *CAgoritmoEntrenamiento*, que se define en este archivo, es proveer de una interfaz común para el entrenamiento de redes neuronales, independientemente del método de entrenamiento utilizado. Dentro de esta clase se encuentran encapsulados objetos de las clases *CRRNN*, *CAgoritmoGenetico* y *CListaEjemplos*, que dan soporte a esta clase para llevar a cabo sus funcionalidades. Existen también en esta clase atributos para cada uno de los parámetros variables de los algoritmos de entrenamiento, y que se utilizan para personalizar las características de los objetos mencionados previamente, que son quienes realizan efectivamente el entrenamiento de la red neuronal.
- **AlgoritmoGenetico.cpp:** En el archivo *AlgoritmoGenetico.cpp* se implementa la clase *CAgoritmoGenetico*. Entre los diversos métodos que se implementan se destaca el método *step*, que produce una nueva generación a partir de la existente mediante la aplicación de los operadores genéticos. Luego, el método *run* realiza una sesión de entrenamiento completa mediante sucesivas llamadas al método *step*, hasta que el nivel de entrenamiento sea aceptable o la cantidad de generaciones supere el máximo admisible. Entre los atributos más importantes encontramos encapsulados objetos de las siguientes clases: *CPoblacion*, *CListaEjemplos*, *CListaEjemplos*, *CRRNN*, *CFuncionInicializar*, *CFuncionEvaluacion*, *CFuncionSeleccion*, *CFuncionCruza* y *CFuncionMutacion*. Mediante llamados a distintos procedimientos de estas clases de soporte se realiza el entrenamiento de una red neuronal basado en el uso de algoritmos genéticos. Los objetos de las clases *CFuncionInicializar*, *CFuncionEvaluacion*, *CFuncionSeleccion*, *CFuncionCruza* y *CFuncionMutacion* realizan las operaciones genéticas necesarias sobre los cromosomas pertenecientes a la población (*CPoblacion*) hasta alcanzar el nivel de entrenamiento adecuado.
- **AlgoritmoGenetico.h:** En este archivo se define una de las clases más importantes de este trabajo, la clase *CAgoritmoGenetico*. Esta clase realiza todos los procesos necesarios para entrenar una red neuronal, utilizando como soporte varias de las clases que se describen a continuación. La interfaz de esta clase es muy rica en procedimientos,

ya que existen diversos parámetros ajustables del método de entrenamiento.

- **Capa.cpp:** La clase *CCapa* y sus clases derivadas se implementan en este archivo. Entre los métodos más importantes se destacan los asociados al entrenamiento de la red neuronal. Entre ellos encontramos al método *evaluar*, que toma como parámetro un conjunto de entradas a la capa y produce la salida correspondiente. Otro método importante, y que varía de acuerdo al tipo de red neuronal, es el método *calcular\_delta*, que tomando como parámetros la salida deseada y la capa siguiente calcula cual es la variación correspondiente a cada capa de acuerdo a su incidencia en el error total. El método *calcular\_variacion\_pesos*, calcula en base al error producido por la red neuronal la variación a realizar sobre cada uno de los pesos sinápticos. Por último, el método *propagar\_error* realiza la variación de los pesos sinápticos correspondientes de acuerdo a su incidencia sobre el error total obtenido. Además de las funciones descritas previamente se encuentran implementadas otras funciones auxiliares que permiten la manipulación de los datos encapsulados en la clase, y el método *funcion\_activacion*, que variará para cada una de las clases derivadas y que, tomando como parámetro un número, devuelve el valor de la función de activación para dicho número.
- **Capa.h:** El objetivo de la clase *CCapa*, que se define en este archivo, es proveer una interfaz común para las capas de los distintos tipos de redes neuronales con conexión hacia delante. En esta capa se definen las funciones comunes que realizan todas las capas de las redes neuronales con conexión hacia delante, y que se heredan en las clases derivadas, como por ejemplo *CCapaBackpropagation*. La clase *CCapa* es una clase virtual, por lo que no se pueden instanciar objetos de esta clase. Para ello es necesario definir las clases derivadas que implementen la lógica propia de cada tipo de red neuronal.
- **Cromosoma.cpp:** En este archivo se implementan los métodos para manipular los valores de los atributos de la clase *CCromosoma*. Uno de los métodos más importantes implementados es el método *init\_random*, que recibe como parámetro dos valores e inicializa la cadena de valores del cromosoma con valores aleatorios en el rango definido por estos dos valores. Otros métodos muy utilizados son aquellos que permiten obtener o asignar el valor de adaptación del individuo tras ser evaluado y los que permiten obtener o asignar los valores de error producido tras ser evaluado el individuo.
- **Cromosoma.h:** La clase *CCromosoma* es una de las clases fundamentales para el entrenamiento de redes neuronales con conexión hacia delante basado en algoritmos evolutivos, y se define en este archivo. Entre sus atributos más importantes se encuentran el vector donde se almacenan los valores del cromosoma y el atributo *fitness* que indica el valor de adaptación del cromosoma. En esta clase

se definen también métodos para modificar los valores del objeto, inicialización del mismo con valores aleatorios, y para guardar sus valores en disco.

- **Cruza.cpp:** Es en este archivo donde se implementan cada una de las clases definidas en el archivo *Cruza.h*. Específicamente, se implementan cuatro variantes del operador de cruce, donde cada una pertenece a una de las variantes descriptas en la sección 6.1.5.
- **Cruza.h:** En este archivo se definen las clases necesarias para experimentar con el uso de diferentes operadores de cruce. Se define la clase abstracta *CFuncionCruza*. Es en esta clase donde se definen diferentes métodos, comunes a todas las clases de operadores de cruce. También se define un método abstracto llamado *cruzar*, que tomando como parámetros dos cromosomas padres realiza la cruce de ellos y devuelve los descendientes. Luego, se definen diferentes clases de operadores de cruce, cada una de las cuales sobrescribe el método *cruzar*. Sin duda los dos atributos más importantes de esta clase son la topología de la red neuronal, que se utiliza para calcular los distintos puntos de cruce posibles de acuerdo a la definición dada en la sección 6.1.5, y el vector que contiene todos los puntos de cruce posibles, que se utiliza en cada uno de los operadores.
- **Ejemplo.cpp:** En este archivo se encuentran implementadas todas las funcionalidades definidas en el archivo *Ejemplo.h*. El ejemplo se almacena en un vector de tipo de dato *double*. Se mantienen también en memoria la cantidad de entradas y de salidas que contiene el ejemplo. De esta manera, el vector contiene una cantidad de posiciones equivalentes a la cantidad de entradas y salidas. Dentro de esta estructura, las primeras posiciones corresponden a las entradas, seguidas por los valores de salida esperados.
- **Ejemplo.h:** Este archivo contiene la definición de la clase *CEjemplo*, la cual se utiliza para mantener en memoria cada uno de los ejemplos, ya sean de entrenamiento o de validación. Se definen los métodos para obtener y asignar valores al ejemplo, así como también para almacenar y obtener los ejemplos en archivos.
- **Evaluacion.cpp:** En este archivo se implementan los servicios definidos en el archivo *Evaluacion.h*. El método más importante de la clase *CFuncionEvaluacion* es *evaluar*, que calcula el valor de adaptación de todos los individuos de una población, la cual recibe como parámetro.
- **Evaluacion.h:** La clase *CFuncionEvaluacion* que se define en este archivo se utilizará para realizar la evaluación, durante una sesión de entrenamiento, de los individuos que componen la población.

- **Include.cpp:** En este archivo se implementan las funciones comunes que se definen en el archivo descripto a continuación y que se utilizan en los distintos módulos que componen el sistema.
- **Include.h:** En este archivo se definen todas las constantes que se utilizan a lo largo del proyecto. En este archivo también se definen un conjunto de funciones simples que se utilizan en los diferentes módulos del proyecto. Por último se definen en este archivo las funciones para almacenar las estadísticas del proyecto.
- **Inicializar.cpp:** En este archivo se implementan las funcionalidades que provee el módulo de inicialización. Para la clase *CFuncionInicializar* se implementan funcionalidades que afectan las características de la población resultante como por ejemplo la cantidad de individuos, la longitud de los cromosomas que la componen, y el rango de valores de inicialización de cada cromosoma. Luego, se implementan para cada una de las clases el método *inicializar*, de acuerdo a las características explicadas en la sección 6.1.7.
- **Inicializar.h:** En este archivo se definen las clases encargadas de generar la población inicial de un algoritmo genético. La clase *CFuncionInicializar*, define una clase abstracta con las características comunes de las funciones de inicialización implementadas. Las clases *CFuncionInicializarRandom* y *CFuncionInicializarRandomSelectiva* sobrescriben el método *inicializar* de acuerdo a las características de cada implementación, retornando en ambos casos un objeto de la clase *CPoblacion*.
- **Lista.cpp:** En este archivo se realiza la implementación de las clases *CNodo* y *CLista*. En la clase *CNodo* se implementa toda la lógica para insertar datos y modificarlos. También se implementa la lógica para obtener los nodos que se encuentran situados a su derecha y a su izquierda. Por su parte, para la clase *CLista* se implementan todas las funcionalidades relacionadas con altas, bajas y modificaciones de registros, así como también las relacionadas con el movimiento a través de la estructura de datos. Por último, ya que estas clases utilizan la funcionalidad de *Templates* provistas por el lenguaje C++, se hace una instanciación explícita de la clase *CLista* para las clases *CEjemplo* y *CCromosoma*.
- **Lista.h:** En este archivo se definen la interfaz de dos clases distintas utilizadas para mantener información en memoria, la clase *CNodo* y la clase *CLista*. La primera se utiliza para almacenar datos dentro de una lista doblemente enlazada, que se define en la clase *CLista*. En ambos casos se definen los atributos (datos) y los métodos asociados, que definen la lógica con que funciona la clase.
- **ListaCromosomas.cpp:** Es en este archivo donde se implementan los métodos definidos en el archivo *ListaCromosomas.h*. La mayor parte

de las funcionalidades que provee esta clase se implementan utilizando como soporte un objeto de la clase *CLista* definida previamente.

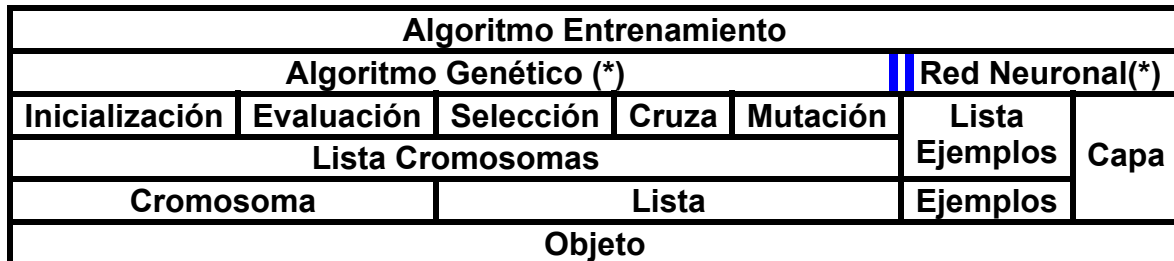
- **ListaCromosomas.h:** Esta clase define la estructura de datos que contendrá el conjunto de cromosomas (población) que se utilizará durante una sesión de entrenamiento de una red neuronal basada en algoritmos evolutivos. Esta clase se apoya en las clases *CCromosoma* y *CLista* para implementar sus funcionalidades, las cuales incluyen las altas, bajas y modificaciones de cromosomas, y otros métodos relacionados con el movimiento a lo largo de la estructura. En este archivo se define *CPoblacion* como sinónimo de *CListaCromosomas*. El único atributo que se define en esta clase es un objeto de la clase *CLista*.
- **ListaEjemplos.cpp:** Es en este archivo donde se implementan los métodos definidos en el archivo *ListaEjemplos.h*. La mayor parte de las funcionalidades que provee esta clase se implementan utilizando como soporte un objeto de la clase *CLista* definida previamente.
- **ListaEjemplos.h:** Esta clase define la estructura de datos que contendrá el conjunto de ejemplos que el usuario ingresa para el entrenamiento de una red neuronal. Esta clase se apoya en las clases *CEjemplo* y *CLista* para implementar sus funcionalidades, las cuales incluyen las altas, bajas y modificaciones de ejemplos, el movimiento a lo largo de la estructura, y métodos para persistir la información en disco. El único atributo que definido en esta clase es un objeto de la clase *CLista*.
- **Mutacion.cpp:** Es en este archivo donde se implementan cada una de las clases definidas en el archivo *Mutacion.h*. Específicamente, se implementan tres variantes del operador de mutación, donde cada una pertenece a una de las variantes descritas en la sección 6.1.6. La principal diferencia entre los operadores es la manera en que actualizan la probabilidad de mutación. El resto de las funcionalidades se encuentran implementadas para la clase *CFuncionMutacion*, y entre los métodos más importantes se destacan *mutar\_numerica* y *mutar\_bit*, que tomando como parámetro un cromosoma (objeto de la clase *CCromosoma*) realizan la mutación numérica o la mutación binaria del cromosoma, como se describe en la sección 6.1.6. El método *mutar* toma un objeto de la clase *CPoblacion* y realiza la mutación de todos los cromosomas que pertenecen a la misma.
- **Mutacion.h:** En este archivo se define el conjunto de clases que permiten realizar experimentos sobre varios tipos de operadores de mutación. Dado que entre los diferentes operadores de mutación a estudiar se encuentran gran cantidad de similitudes, la clase abstracta *CFuncionMutacion* contiene gran cantidad de métodos, los cuales heredan sus clases derivadas.



- **Objetos.h**: En este archivo se define la clase *CObjeto*. Esta clase es una clase virtual que sólo se utiliza para aplicar el concepto de polimorfismo sobre todas las clases que quieran utilizar listas.
- **Rrnn.cpp**: La implementación de la clase *CRRNN* y sus clases derivadas se lleva a cabo en este archivo. La mayoría de los métodos se implementan a nivel de la clase *CRRNN* y se heredan en todas las clases derivadas, las cuales deben sobrescribir el método *entrenar*. Entre los atributos más importantes de la clase se encuentran la topología, donde se definen la cantidad de capas (y neuronas por capa) de la red neuronal, el conjunto de ejemplos de entrenamiento y de evaluación que se utilizarán durante una sesión de entrenamiento (ambos son objetos de la clase *CListaEjemplos*), y un vector de objetos de tipo *CCapa*, que mediante polimorfismo implementan la lógica de cada capa. También se definen algunos atributos extras, que se utilizan para ajustar las características de la sesión de entrenamiento. Entre los diversos métodos que se implementan encontramos aquellos que permiten la manipulación de estos atributos, y los que permiten persistir en disco la información de los objetos. También se encuentran diversos métodos para evaluar ejemplos y patrones de entrada. Por último, para cada clase se implementan el método *entrenar*, que tomando como parámetro el máximo error admisible entrena una red neuronal.
- **Rrnn.h**: En este archivo se define una clase abstracta, llamada *CRRNN*, que define una interfaz común para todas las redes neuronales con conexión hacia delante. Como se mencionó previamente gran parte de las funcionalidades provistas por este grupo de clases se definidas en la clase *CRRNN*, dejando para las clases derivadas las implementaciones específicas de cada red neuronal. Dado que se utilizan como soporte las clases derivadas de *CCapa* y el polimorfismo que proporciona el paradigma de programación utilizado, cada clase derivada deberá sobrescribir únicamente el método *entrenar*, para ajustarlo a las características propias de la topología. De la clase *CRRNN* derivarán las clases específicas para cada topología como por ejemplo *CBackPropagation*.
- **Seleccion.cpp**: En este archivo se implementan cada una de las clases definidas en el archivo *Seleccion.h*. Específicamente, se implementan cuatro variantes del operador de selección, donde cada una pertenece a una de las variantes descritas en la sección 6.1.4.
- **Seleccion.h**: En este archivo se definen las clases necesarias para experimentar con el uso de diferentes operadores de selección. Se define la clase abstracta *CFuncionSeleccion*, que contiene un único método, llamado *seleccionar*. Luego, se definen diferentes clases de operadores de selección, cada una de las cuales sobrescribe el método *seleccionar*, que tomando como parámetro la población actual (objeto de la clase *CPoblacion*), devuelve la población de individuos que sobreviven y pasan a la próxima generación.

### B.3.2.3. Esquema de interacción de bloques

En la figura B.7 se muestra un esquema que representa cómo interactúan los módulos vistos en la sección anterior. Cada módulo está representado por un archivo fuente. No se representa el archivo `include.h` que posee funciones comunes a todos los bloques o funciones que no son específicas de un bloque (no se muestra para facilitar la comprensión del diagrama ya que todos los bloques se relacionan con él).



**Figura B.7.** Diagrama de interacción de módulos

(\*) La doble separación entre los módulos Red Neuronal y Algoritmo Genético representa una iteración lateral entre ambas clases.

De esta manera, el módulo *Algoritmo de Entrenamiento* define una capa abstracta para entrenar redes neuronales. Luego, dependiendo de las opciones de entrenamiento que seleccione el usuario invoca al módulo *Algoritmo Genético* o *Red Neuronal*.

El módulo *Algoritmo Genético* utiliza los módulos *Inicialización*, *Evaluación*, *Selección*, *Cruza* y *Mutación* como soporte para entrenar la red neuronal. Estos a su vez hacen uso de una población que está representada en el módulo *Lista Cromosomas*, quien a su vez dispone de las funcionalidades prestadas por los módulos *Lista* y *Cromosoma*. En la figura se muestra que existe una interacción lateral del módulo *Algoritmo Genético* con el módulo *Red Neuronal*. Esta interacción se produce ya que el cromosoma debe ser convertido en una red neuronal para ser evaluado en el módulo *Evaluación*.

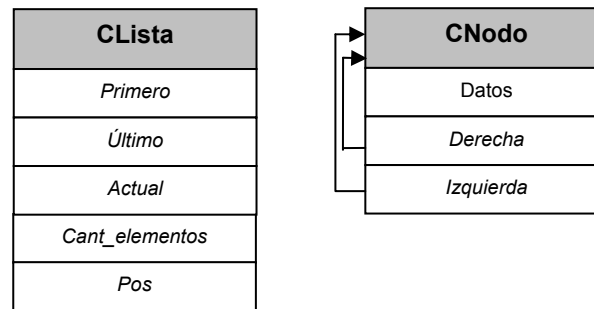
Por su parte, el módulo Red Neuronal requiere de los servicios prestados por los módulos *Capa* y *Lista Ejemplo* para llevar a cabo el entrenamiento de la red neuronal. Este último a su vez está implementado sobre las funcionalidades provistas en los módulos *Lista* y *Ejemplo*.

La principal ventaja de este esquema de capas es que permite, manteniendo la definición, modificar la implementación de una de las capas. De esta manera, el ambiente es apropiado para la evaluación de diferentes algoritmos. A su vez, al utilizar el paradigma de programación orientado a objetos, permite definir varias clases que compartan una definición común, y que cada una de ellas realice una implementación diferente de la misma.

### B.3.3. Estructuras de datos

En esta sección se explica con mayor nivel de detalle cada una de las estructuras de datos utilizadas en el proyecto.

### **B.3.3.1. Lista**



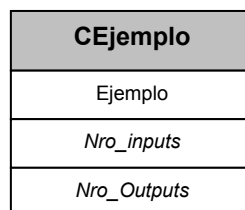
**Figura B.8.** Listas

Esta estructura de datos se utiliza para mantener el listado de individuos (cromosomas) que pertenecen a una determinada población, así como también para mantener en memoria el set de datos de entrenamiento y de evaluación.

La clase CLista, que se muestra en la figura B.8, implementa una lista de datos doblemente enlazada. Contiene un conjunto de punteros (Primero, Último y Actual) a estructuras de datos del tipo CNodo. De esta manera permite moverse en la lista en diferentes posiciones. A su vez almacena la cantidad de elementos y la posición actual dentro de la lista para facilitar el desplazamiento por posiciones (numero de orden).

La clase CNodo contiene un puntero a un objeto de la clase CObjeto (ver figura B.8). De esta manera, y utilizando polimorfismo, esta estructura de datos soporta objetos de cualquier clase, siempre que dicha clase extienda la clase CObjeto. A su vez, la clase CNodo contiene dos punteros (Izquierda y Derecha) a los nodos que se encuentran antes y después de él en la lista.

### **B.3.3.2. Ejemplo**



**Figura B.9.** Ejemplos

Esta clase se utiliza para mantener en memoria cada uno de los ejemplos (ya sean de entrenamiento o de evaluación). Extiende la clase CObjeto para poder ser utilizada en estructuras del tipo Lista. Contiene un vector con el conjunto de valores de entrada y salida deseada, así como también la cantidad de valores que corresponden a entradas y la cantidad que corresponden a las salidas deseadas (ver figura B.9).

### **B.3.3.3. Cromosoma**

CCromosoma
Longi
Cadena
Fitness
Max_error
Min_error

**Figura B.10.** Cromosoma

Esta clase se utiliza para mantener en memoria cada uno de los individuos que componen una determinada población. Extiende la clase CObjeto para poder ser utilizada en estructuras del tipo Lista (que representan a la población). Tal como se observa en la figura B.10, está compuesto por cinco atributos: la longitud del cromosoma, la cadena de genes que representa, el valor de adaptación (fitness) del individuo, y el máximo y mínimo error cometido.

### **B.3.4. Formato de los archivos de estadísticas**

Durante una sesión de entrenamiento se almacenan los datos generados por el tras cada iteración en un archivo de log llamado "step\_result.std". La estructura del mismo se detalla en la Figura B.11.

Step	Max Error	Min Error	Error Train	Error Test	Error Total	Hits Train	Hits Test	Hits Total	Train Time
------	-----------	-----------	-------------	------------	-------------	------------	-----------	------------	------------

**Figura B.11.** Estructura de archivo de estadísticas

El primer campo (step) contiene el número de iteración realizada. El campo **Max Error** contiene el error máximo que produce la red sobre el set de entrenamiento. Corresponde a la máxima diferencia entre la salida deseada y la salida obtenida para algún ejemplo del set de entrenamientos. Análogamente, **Min Error** contiene el mínimo error que produce la red neuronal sobre el set de entrenamiento.

El campo **Error Train** contiene el error cuadrático medio que produce la red sobre el set de entrenamiento. Del mismo modo, el campo **Error Test** contiene el error cuadrático medio que produce la red neuronal con respecto al set de evaluación. El campo Error Total representa el error cuadrático medio sobre el total de ejemplos (set de entrenamiento y set de evaluación).

El campo **Hits Train** contiene el porcentaje de aciertos (cantidad de ejemplos bien clasificados sobre el total), evaluados sobre el set de entrenamiento. Los campos **Hits Test** y **Hits Total** son similares pero se calculan sobre el set de evaluación y sobre el total de ejemplos respectivamente.

Por último, el campo train time contiene el tiempo transcurrido en segundos desde el comienzo de la sesión de entrenamiento hasta el instante en que se

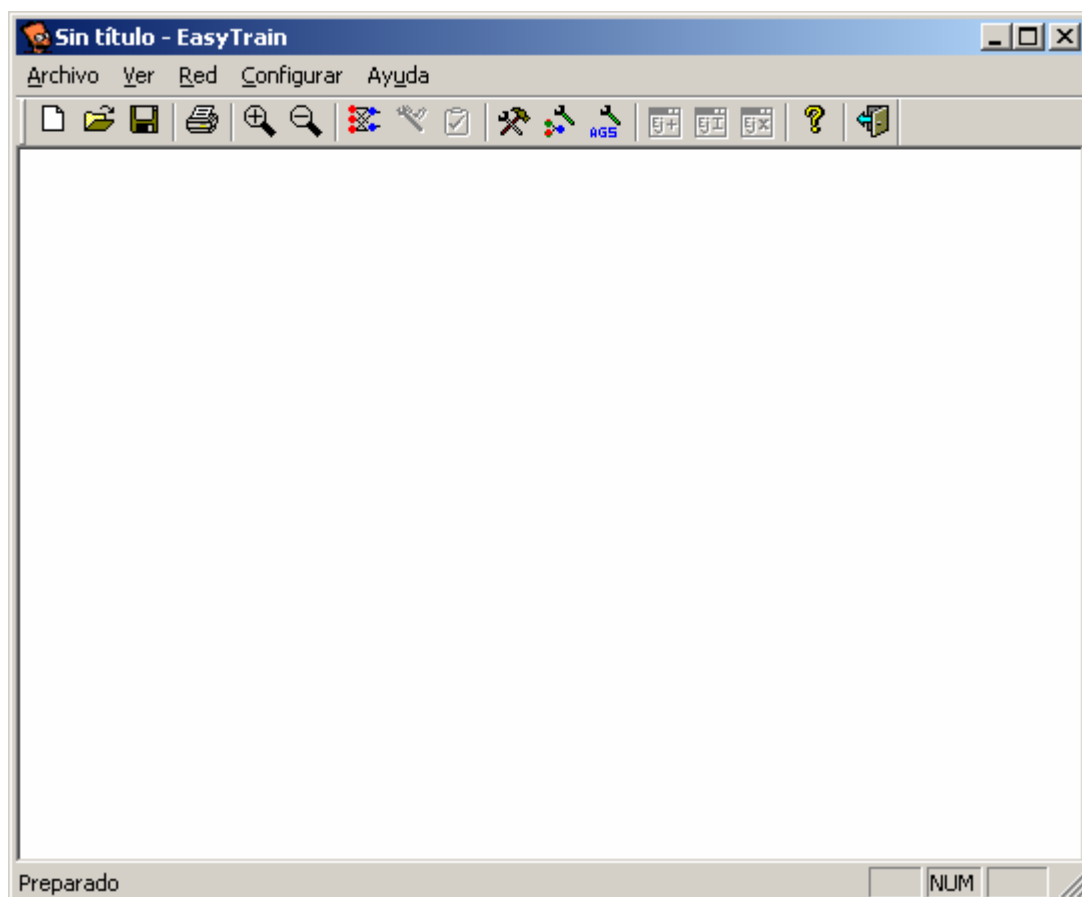
registran las estadísticas. Como se mencionó en el capítulo 7, este campo se utiliza para hacer comparables las estadísticas de los distintos métodos de entrenamiento.

Los valores de una misma línea del archivo de estadísticas corresponden a una misma iteración y se encuentran separados por tabulaciones.

## **B.4. Implementación**

### **B.4.1. Pantallas**

El sistema desarrollado consiste en una aplicación Windows. La pantalla principal del sistema se muestra en la figura B.12. Esta es la única ventana del sistema, ya que el resto de la aplicación se maneja a través de cuadros de diálogo. La información que se despliega en esta pantalla dependerá de la opción seleccionada en el menú *Ver*. Todas las funcionalidades implementadas se encuentran en el menú de la aplicación, y las más importantes contienen un icono dentro de la barra de herramientas de la aplicación.



**Figura B.12.** Pantalla Principal

## B.4.2. Funciones implementadas

A continuación se detalla el conjunto de funciones implementadas, y su organización en el menú de la aplicación (Figura B.13). Luego, se explica en detalle cada una de ellas.

Archivo	<ul style="list-style-type: none"> <li>Nuevo</li> <li>Abrir</li> <li>Guardar</li> <li>Guardar Como</li> <li>Exportar pesos</li> <li>Importar ejemplos de entrenamiento</li> <li>Importar ejemplos de evaluación</li> <li>Imprimir</li> <li>Presentación preliminar</li> <li>Configurar impresora</li> <li>Salir</li> </ul>	
Ver	<ul style="list-style-type: none"> <li>Barra de herramientas</li> <li>Barra de estado</li> <li>Ejemplos de entrenamiento</li> <li>Ejemplos de evaluación</li> <li>Red Neuronal</li> <li>Errores de entrenamiento</li> <li>Errores de evaluación</li> </ul>	
Red	<ul style="list-style-type: none"> <li>Topología</li> <li>Ejemplos de entrenamiento</li> <li>Ejemplos de evaluación</li> <li>Entrenar</li> <li>Evaluar</li> </ul>	<ul style="list-style-type: none"> <li>Agregar</li> <li>Modificar</li> <li>Eliminar</li> <li>Agregar</li> <li>Modificar</li> <li>Eliminar</li> </ul>
Configurar	<ul style="list-style-type: none"> <li>Algoritmo Genético</li> <li>Red Neuronal</li> <li>Tipo de entrenamiento</li> </ul>	
Ayuda	<ul style="list-style-type: none"> <li>Acerca de</li> </ul>	

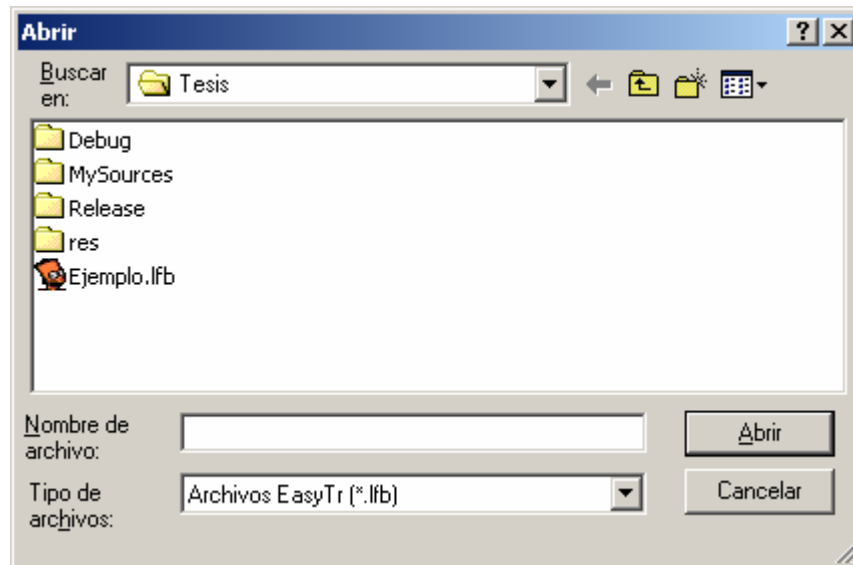
**Figura B.13.** Estructura del menú

### **B.4.2.1. Menú Archivo > Nuevo**

Crea un nuevo documento. El documento se crea totalmente vacío y se resetean todas las opciones a sus valores default.

#### **B.4.2.2. Menú Archivo > Abrir**

Abre un cuadro de diálogo standard de Windows para seleccionar el archivo que se desea abrir, como se muestra en la figura B.14.



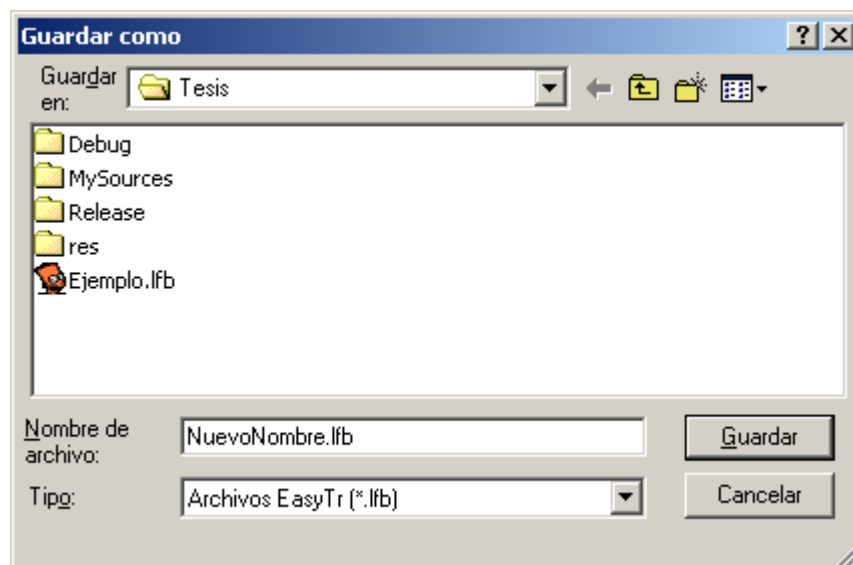
**Figura B.14.** Abrir Archivo

#### **B.4.2.3. Menú Archivo > Guardar**

Guarda las modificaciones realizadas sobre un archivo existente. En caso de que el archivo no haya sido guardado previamente el procedimiento es similar a la opción Guardar como que se explica a continuación.

#### **B.4.2.4. Menú Archivo > Guardar Como**

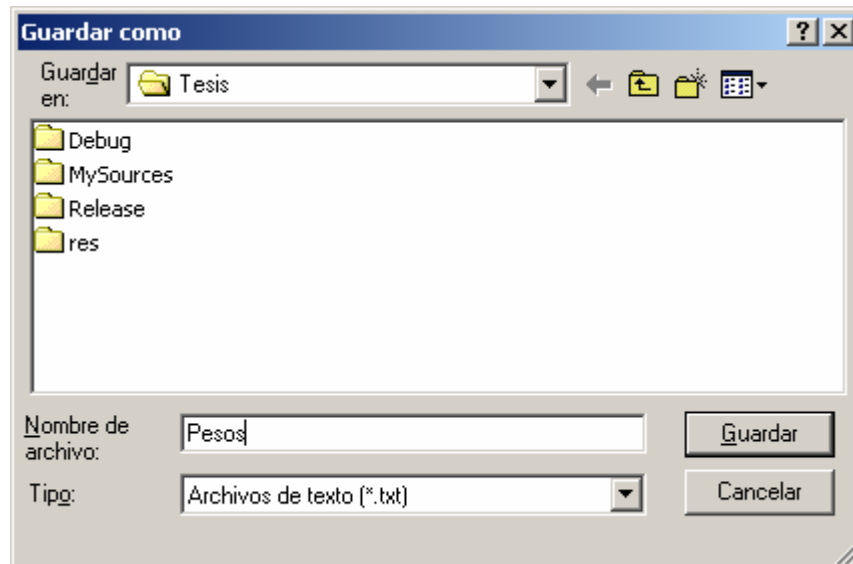
Abre un cuadro de diálogo standard de Windows para ingresar el nombre con el que se desea guardar el archivo, como se muestra en la figura B.15.



**Figura B.15.** Guardar archivo

#### **B.4.2.5. Menú Archivo > Exportar Pesos**

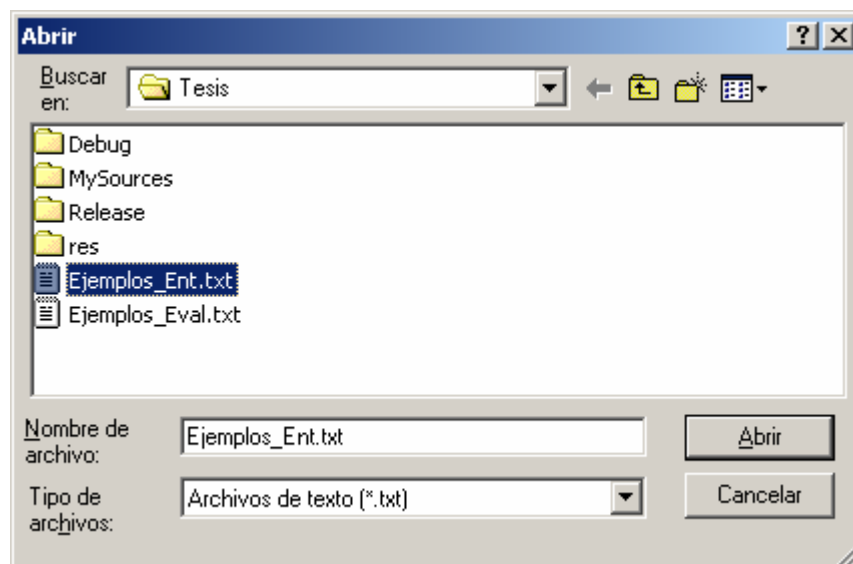
Abre un cuadro de diálogo standard de Windows para ingresar el nombre con el que se desea guardar el archivo que contiene el conjunto de pesos. En la figura B.16 se muestra el diálogo mencionado previamente.



**Figura B.16.** Exportar pesos

#### **B.4.2.6. Menú Archivo > Importar Ejemplos de Entrenamiento**

Abre un cuadro de diálogo standard de Windows para seleccionar el archivo que contiene el conjunto de ejemplos, como se muestra en la figura B.17. Estos ejemplos son importados como ejemplos de entrenamiento.



**Figura B.17.** Importar ejemplos

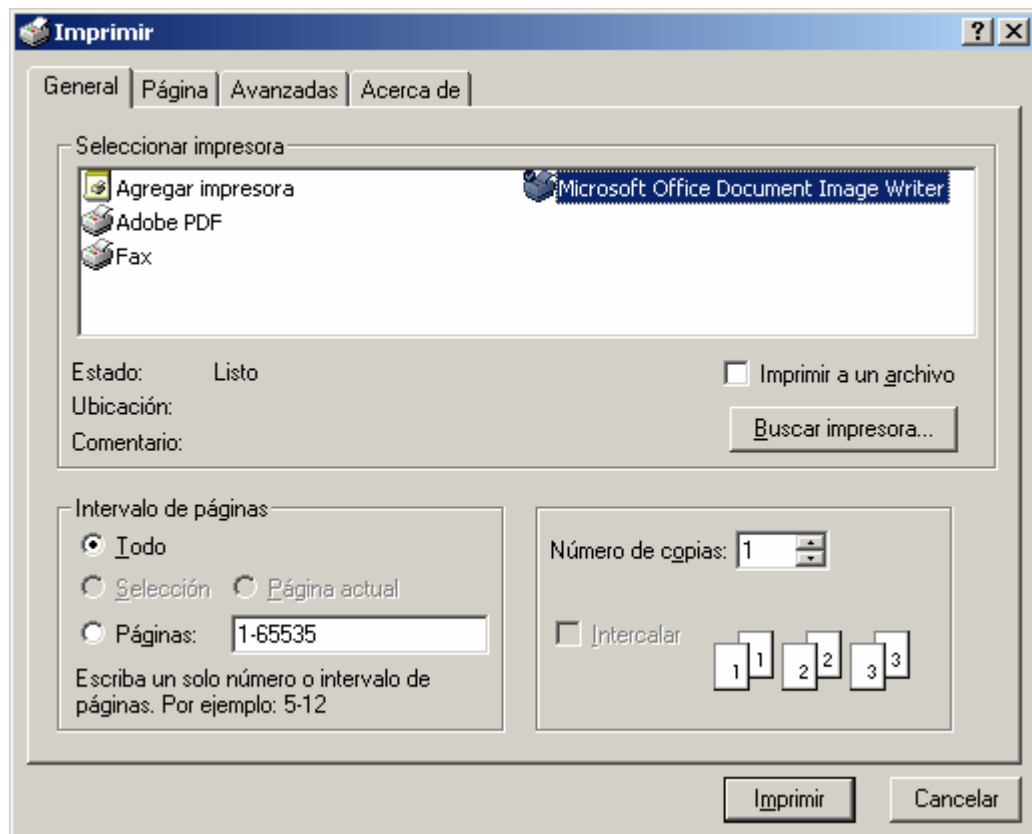


**B.4.2.7. Menú Archivo > Importar Ejemplos de Evaluación**

Esta opción es similar a la anterior sólo que los ejemplos son importados como ejemplos de evaluación.

**B.4.2.8. Menú Archivo > Imprimir**

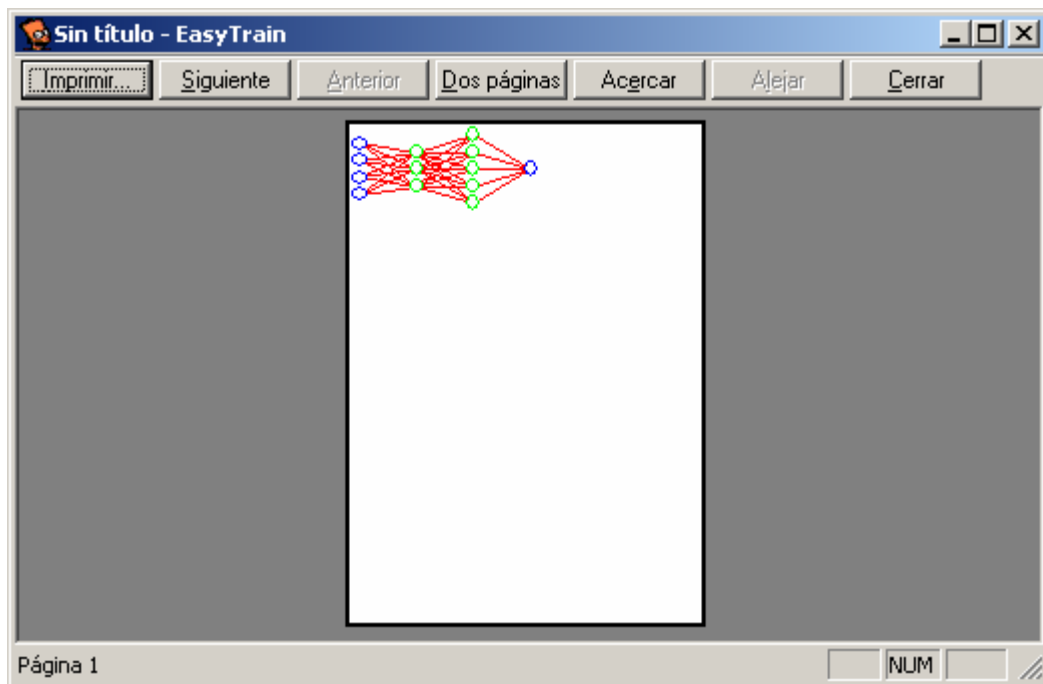
Abre un cuadro de diálogo standard de Windows para imprimir, como el que se muestra en la figura B.18.



**Figura B.18.** Imprimir

**B.4.2.9. Menú Archivo > Presentación Preliminar**

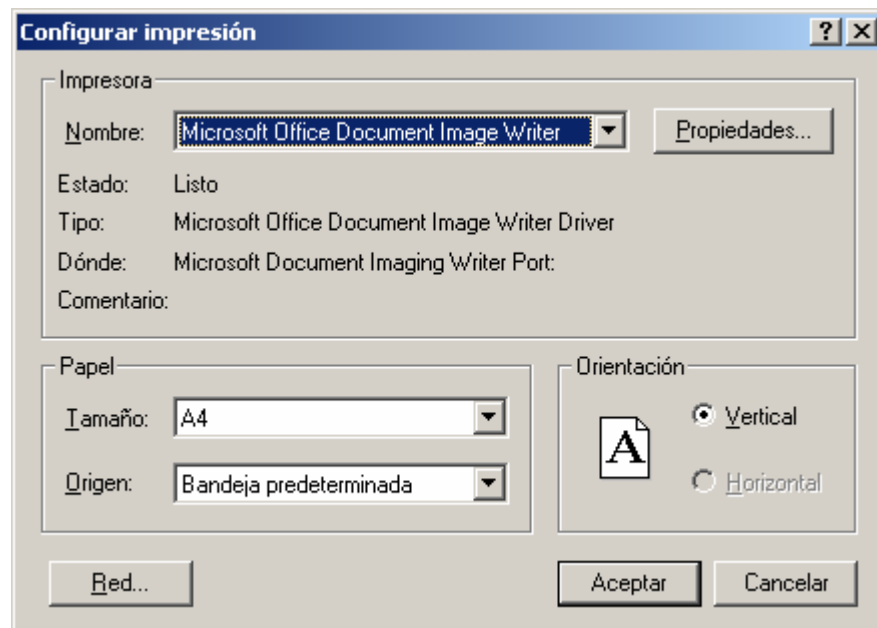
Abre un cuadro de diálogo standard de Windows para obtener una vista previa de la impresión (ver figura B.19). Desde este diálogo se puede imprimir el documento.



**Figura B.19.** Presentación preliminar

#### **B.4.2.10. Menú Archivo > Configurar Impresora**

Abre un cuadro de diálogo standard de Windows para configurar la impresora, como el que se muestra en la figura B.20.



**Figura B.20.** Configurar Impresora

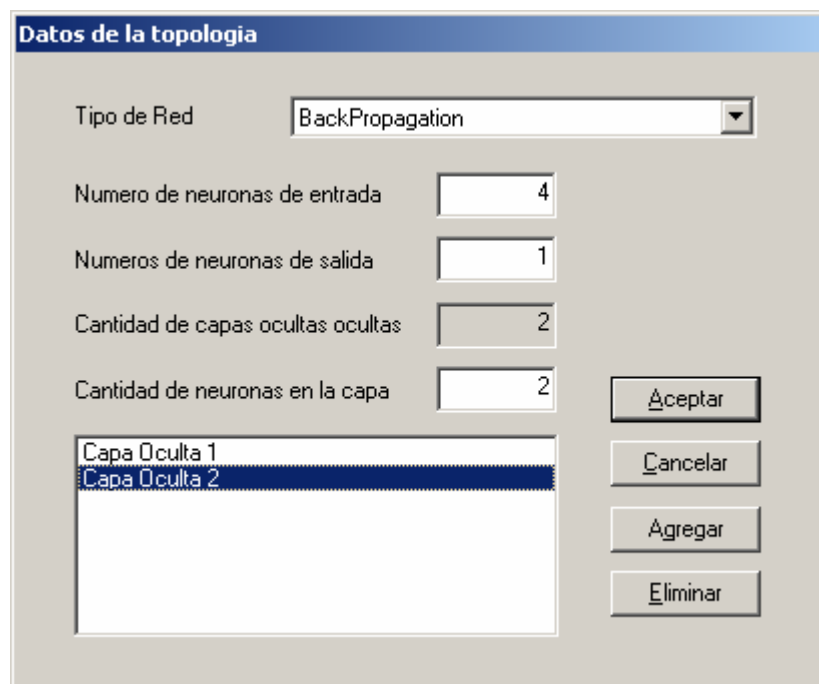
#### **B.4.2.11. Menú Ver**

Este menú contiene varias opciones. Todas estas están relacionadas con la forma en que se visualiza la información en pantalla:

- Barra de herramientas: Habilita/Deshabilita la barra de herramientas.
- Barra de estado: Habilita/Deshabilita la barra de estado.
- Ejemplos de entrenamiento: Despliega en la pantalla principal el set de datos de entrenamiento.
- Ejemplos de evaluación: Despliega en la pantalla principal el set de datos de evaluación.
- Red Neuronal: Muestra gráficamente en la pantalla principal la topología de la Red Neuronal.
- Errores de entrenamiento: Despliega en la pantalla principal el set de datos de entrenamiento junto con el error cometido por la red neuronal al clasificarlo.
- Errores de evaluación: Despliega en la pantalla principal el set de datos de evaluación junto con el error cometido por la red neuronal al clasificarlo.

#### **B.4.2.12. Menú Red > Topología**

Abre un cuadro de diálogo donde el usuario puede definir la topología de la red neuronal que desea entrenar. En la figura B.21 se pueden observar las distintas características de la topología que se pueden configurar.



The image shows a Windows-style dialog box titled "Datos de la topología". It has a light gray background and a blue title bar. Inside, there are several input fields and a list box. The "Tipo de Red" field is a dropdown menu showing "BackPropagation". Below it are four numeric input fields: "Numero de neuronas de entrada" with the value 4, "Numeros de neuronas de salida" with the value 1, "Cantidad de capas ocultas ocultas" with the value 2, and "Cantidad de neuronas en la capa" with the value 2. Below these fields is a list box containing two items: "Capa Oculta 1" and "Capa Oculta 2", with "Capa Oculta 2" selected. To the right of the list box are four buttons: "Aceptar", "Cancelar", "Agregar", and "Eliminar".

**Figura B.21. Topología**

#### **B.4.2.13. Menú Red > Ejemplos de entrenamiento > Agregar**

Abre un cuadro de diálogo donde el usuario puede ingresar los valores de entrada y salida deseada del nuevo ejemplo de entrenamiento, como se muestra en la figura B.22.

Nerurona	Valor	Nuevo valor
Input 00	0.000000	0
Input 01	0.000000	
Input 02	0.000000	
Input 03	0.000000	
Output 00	0.000000	

**Figura B.22.** Agregar Ejemplo

#### **B.4.2.14. Menú Red > Ejemplos de entrenamiento > Modificar**

Permite al usuario modificar uno de los ejemplos de entrenamiento ya ingresados. El usuario debe seleccionar en primer lugar cual es el número de ejemplo a modificar (ver figura B.23).

**Figura B.23.** Seleccionar ejemplo

Una vez seleccionado el ejemplo se despliega un cuadro de diálogo (figura B.24) donde el usuario visualiza los valores actuales e ingresa los nuevos valores.

Nerurona	Valor
Input 00	0.000000
Input 01	0.000000
Input 02	0.000000
Input 03	0.000000
Output 00	0.000000

Nuevo valor:

**Figura B.24.** Modificar ejemplo

#### **B.4.2.15. Menú Red > Ejemplos de entrenamiento > Eliminar**

Permite al usuario eliminar uno de los ejemplos de entrenamiento ya ingresados. El usuario debe seleccionar en primer lugar cual es el número de ejemplo a eliminar, como se muestra en la figura B.23.

Una vez seleccionado el ejemplo se despliega un cuadro de diálogo donde el usuario visualiza los valores actuales y confirma o cancela la operación solicitada, como se muestra en la figura B.25.

Nerurona	Valor
Input 00	0.000000
Input 01	0.000000
Input 02	0.000000
Input 03	0.000000
Output 00	0.000000

**Figura B.25.** Eliminar ejemplo

#### **B.4.2.15. Menú Red > Ejemplos de Evaluación**

Esta opción del menú tiene a su vez tres subopciones (Agregar, Modificar y Eliminar). Cada una de ellas funciona de la misma manera que las descritas en las últimas 3 secciones, sólo que lo hace sobre el conjunto de ejemplos de evaluación

#### **B.4.2.16. Menú Red > Entrenar**

Despliega un cuadro de diálogo donde se visualizan todas las opciones seleccionadas del algoritmo de entrenamiento. Una vez verificado que las opciones sean las deseadas, el usuario puede iniciar una sesión de entrenamiento haciendo click en el botón Run, como se muestra en la figura B.26.

The screenshot shows a software window titled "Entrenamiento" (Training). It contains several sections for configuring a Genetic Algorithm (Algoritmo Genético) for training a neural network (Red Neuronal).

- Algoritmo Genético:**
  - Cantidad de individuos: 100
  - Cantidad máxima de generaciones: 1000
  - Mutación: Minima 0.0001, Maxima 0.0005
  - Función de Selección: Ruleta
  - Función de Cruza: Binomial
  - Valor de inicialización: Minimo -1, Maximo 1
  - Función de Mutación: Uniforme
  - Función de Inicialización: Random-Selectiva
  - ☐ Mutación Numérica ?
  - Probabilidad de cruza: 0.1
- Red Neuronal:**
  - Tasa de Aprendizaje: 0.1
  - Momento: 0.6
  - Cantidad Máxima de Ciclos: 1000
- Entrenar la red neuronal ...**
  - por Algoritmo Genético
  - Error promedio maximo admisible AG: 0.01
  - Error promedio maximo admisible RRNN: 0.01
- Resultados:**
  - Total Generaciones: 0
  - Total Ciclos: 0
  - Error Actual: 0
  - Buttons: Run, Cancelar

**Figura B.26. Entrenar Red Neuronal**

#### **B.4.2.17. Menú Red > Evaluar**

Esta opción permite al usuario evaluar la respuesta que produce una red neuronal previamente entrenada sobre un patrón de entrada determinado (ver figura B.27).

**Evaluar patrones**

Neuronas	Valores
Input 00	0.000000
Input 01	0.000000
Input 02	0.000000
Input 03	0.000000

Valor:

Neuronas	Valores
Output 00	0.000000

**Figura B.27.** Evaluar ejemplo

#### **B.4.2.18. Menú Configurar > Algoritmo Genético**

Despliega un cuadro de diálogo donde el usuario puede ingresar los valores deseados para cada uno de los parámetros que componen el algoritmo de entrenamiento basado en algoritmos genéticos. El conjunto de parámetros configurables se observa en la figura B.28.

**Propiedades del algoritmo genético**

Cantidad de individuos:  Cantidad máxima de generaciones:

Función de Selección:  Función de Cruza:

Función de Mutación:  Función de Inicialización:

Mutación mínima:  Valor de inicialización mínimo:

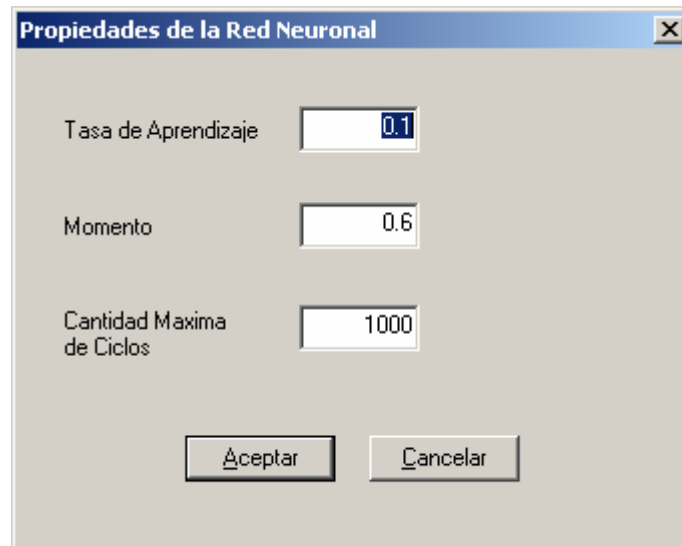
Mutación máxima:  Valor de inicialización máximo:

☐ Mutacion Numerica ? Probabilidad cruza:

**Figura B.28.** Configurar algoritmo genético

#### **B.4.2.19. Menú Configurar > Red Neuronal**

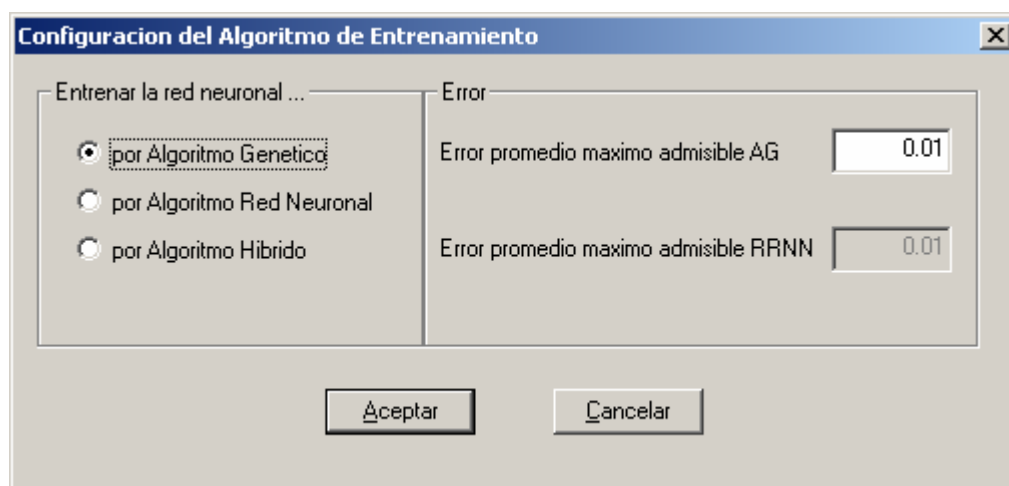
Despliega un cuadro de diálogo donde el usuario puede ingresar los valores deseados para cada uno de los parámetros que componen el algoritmo de entrenamiento propio de la topología de la red neuronal, como se muestra en la figura B.29.



**Figura B.29.** Configurar Red Neuronal

#### **B.4.2.20. Menú Configurar > Tipo Entrenamiento**

Despliega un cuadro de diálogo donde el usuario puede ingresar el tipo de algoritmo de entrenamiento deseado y la cota máxima de error admitido, como se muestra en la figura B.30.



**Figura B.30.** Configurar tipo de entrenamiento



### B.4.3. Pruebas

En esta sección se presenta el detalle de pruebas utilizadas para la validación de la solución; en la misma se presenta el plan de pruebas y posteriormente el desarrollo de las mismas. Solamente se realizan pruebas parciales del sistema, debido a que resulta imposible probar el sistema completo, por tratarse de un simulador cuyos resultados son no determinísticos.

#### B.4.3.1. Plan de pruebas

<b>Objetivo de la prueba</b>	Detectar errores de programación y funcionamiento del algoritmo de entrenamiento.
<b>Objetos a probar</b>	Algoritmo de entrenamiento
<b>Características a probar</b>	Funcionamiento del algoritmo de entrenamiento de la red neuronal, sabiendo cuál debe ser el resultado final. La prueba se hará sobre plataforma Windows.
<b>Cantidad de pruebas a utilizar</b>	12
<b>Características a no probar</b>	Performance (velocidad de ejecución) del algoritmo de entrenamiento.
<b>Método de prueba a utilizar</b>	Prueba por comparación de resultados y adivinación de errores.

**Tabla B.2.** Plan de pruebas

#### B.4.3.2. Documento de diseño de la prueba

<b>Procedimiento de pruebas</b>	Las pruebas serán llevadas a cabo de acuerdo a lo descrito en la sección B.4.3.3 – Especificación de los casos de prueba, registrándose las diferencias que se detecten entre el resultado esperado y el obtenido.
<b>Métodos de prueba a utilizar</b>	Se compararán los resultados obtenidos con los resultados esperados al hacer el cálculo del algoritmo en forma manual.
<b>Criterios para la aprobación de las pruebas</b>	Los criterios para la aprobación de las pruebas se describen a continuación: <i>Excelente</i> : Cuando el resultado obtenido luego de realizada la prueba es idéntico al resultado citado en la sección B.4.3.3 – Especificación de los casos de prueba. <i>Muy bueno</i> : Cuando el resultado obtenido luego de realizada la prueba es parecido al resultado citado en la sección B.4.3.3 – Especificación de los casos de prueba. <i>Bueno</i> : Cuando el resultado obtenido luego de realizada la prueba no fue el resultado citado

	<p>en la sección B.4.3.3 – Especificación de los casos de prueba, pero no ha provocado anomalías en el funcionamiento del programa.</p> <p><i>Regular:</i> Cuando el resultado obtenido luego de realizada la prueba no fue el resultado citado en la sección B.4.3.3 – Especificación de los casos de prueba, pero ha provocado anomalías en el funcionamiento del programa.</p> <p><i>Malo:</i> Cuando el resultado obtenido luego de realizada la prueba no fue el resultado citado en la sección B.4.3.3 – Especificación de los casos de prueba, pero ha provocado anomalías graves en el funcionamiento del programa tales como la salida del sistema o “colgarse”.</p>
--	---

**Tabla B.3.** Diseño de la prueba

**B.4.3.3. Especificación de los casos de prueba**

ÍTEM	OBJETIVO	ACCIÓN	ENTRADA	RESULTADO ESPERADO
1	Transformar un valor del tipo TPeso en un tipo TCromosoma	Ejecutar la función de transformación ToTCromosoma con la entrada especificada	Parámetro value= -15.1	0xDA00C580 (valor equivalente en TCromosoma)
2	Transformar un valor del tipo TCromosoma en un tipo TPeso	Ejecutar la función de transformación FromTCromosoma con la entrada especificada	Parámetro value= 0xDA00C580	-15.1 (valor equivalente en TPeso)
3	Transformar un cromosoma en la red neuronal que representa.	Ejecutar la función de morfogénesis con la entrada especificada	Un objeto de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
4	Cruzar dos cromosomas	Ejecutar el método de cruza de la clase CFuncionCruzaUniforme para dos cromosomas	Dos objetos de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
5	Cruzar dos cromosomas	Ejecutar el método de cruza de la clase CFuncionCruzaBinomial para dos cromosomas	Dos objetos de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
6	Cruzar dos cromosomas	Ejecutar el método de cruza de la clase CFuncionCruzaSimple para dos cromosomas	Dos objetos de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
7	Cruzar dos cromosomas	Ejecutar el método de cruza de la clase CFuncionCruzaMultiPuntos para dos cromosomas	Dos objetos de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.

ÍTEM	OBJETIVO	ACCIÓN	ENTRADA	RESULTADO ESPERADO
8	Inicializar cromosomas con valores aleatorios en un rango	Ejecutar el método <code>init_random</code> de la clase <code>CCromosoma</code>	Topología=2,2,1 Máximo=0.0001 Mínimo=-0.0001	Red neuronal con la topología especificada y con valores de pesos variando dentro del rango.
9	Actualizar los pesos tras la presentación de un patrón de entrenamiento	Ejecutar el método <code>run</code> de la clase <code>CAgoritmoEntrenamiento</code>	Un objeto de tipo <code>CAgoritmoEntrenamiento</code> con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
10	Calcular el error cuadrático medio que produce una red neuronal sobre un set de datos de entrenamiento	Ejecutar el método <code>evaluar_ejemplos</code> de la clase <code>CRRNN</code>	Un objeto de tipo <code>CRRNN</code> con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
11	Entrenar una red neuronal con el método clásico	Ejecutar el método <code>run</code> de la clase <code>CAgoritmoEntrenamiento</code>	Un objeto de tipo <code>CAgoritmoEntrenamiento</code> con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
12	Entrenar una red neuronal con el método basado en algoritmos genéticos	Ejecutar el método <code>run</code> de la clase <code>CAgoritmoEntrenamiento</code>	Un objeto de tipo <code>CAgoritmoEntrenamiento</code> con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.

Tabla B.4. Especificación de los casos de prueba

**B.4.3.4. Especificación del procedimiento de prueba**

ÍTEM	ACCIÓN	ENTRADA	RESULTADO ESPERADO
1	Ejecutar la función de transformación ToTCromosoma con la entrada especificada	Parámetro value= -15.1	0xDA00C580 (valor equivalente en TCromosoma)
2	Ejecutar la función de transformación FromTCromosoma con la entrada especificada	Parámetro value= 0xDA00C580	-15.1 (valor equivalente en TPeso)
3	Ejecutar la función de morfogénesis con la entrada especificada	Un objeto de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
4	Ejecutar el método de cruce de la clase CFuncionCruzaUniforme para dos cromosomas	Dos objetos de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
5	Ejecutar el método de cruce de la clase CFuncionCruzaBinomial para dos cromosomas	Dos objetos de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
6	Ejecutar el método de cruce de la clase CFuncionCruzaSimple para dos cromosomas	Dos objetos de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
7	Ejecutar el método de cruce de la clase CFuncionCruzaMultiPuntos para dos cromosomas	Dos objetos de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
8	Ejecutar el método init_random de la clase CCromosoma	Topología=2,2,1 Máximo=0.0001 Mínimo=-0.0001	Red neuronal con la topología especificada y con valores de pesos variando dentro del rango.

ÍTEM	ACCIÓN	ENTRADA	RESULTADO ESPERADO
9	Ejecutar el método run de la clase CAlgoritmoEntrenamiento	Un objeto de tipo CAlgoritmoEntrenamiento con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
10	Ejecutar el método evaluar_ejemplos de la clase CRRNN	Un objeto de tipo CRRNN con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
11	Ejecutar el método run de la clase CAlgoritmoEntrenamiento	Un objeto de tipo CAlgoritmoEntrenamiento con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.
12	Ejecutar el método run de la clase CAlgoritmoEntrenamiento	Un objeto de tipo CAlgoritmoEntrenamiento con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.

Tabla B.5. Especificación del procedimiento de prueba

**B.4.3.5. Informe de los casos de prueba ejecutados**

ÍTEM	ACCIÓN	ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO
1	Ejecutar la función de transformación ToTCromosoma con la entrada especificada	Parámetro value= -15.1	0xDA00C580	0xDA00C580
2	Ejecutar la función de transformación FromTCromosoma con la entrada especificada	Parámetro value= 0xDA00C580	-15.1	-15.1

ÍTEM	ACCIÓN	ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO
3	Ejecutar la función de morfogénesis con la entrada especificada	Un objeto de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.	Los resultados obtenidos son coincidentes con los esperados y se especifican en la sección B.4.3.7.
4	Ejecutar el método de cruza de la clase CFuncionCruzaUniforme para dos cromosomas	Dos objetos de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.	Los resultados obtenidos son coincidentes con los esperados y se especifican en la sección B.4.3.7.
5	Ejecutar el método de cruza de la clase CFuncionCruzaBinomial para dos cromosomas	Dos objetos de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.	Los resultados obtenidos son coincidentes con los esperados y se especifican en la sección B.4.3.7.
6	Ejecutar el método de cruza de la clase CFuncionCruzaSimple para dos cromosomas	Dos objetos de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.	Los resultados obtenidos son coincidentes con los esperados y se especifican en la sección B.4.3.7.

ÍTEM	ACCIÓN	ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO
7	Ejecutar el método de cruza de la clase CFuncionCruzaMultiPunto para dos cromosomas	Dos objetos de tipo CCromosoma con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.	Los resultados obtenidos son coincidentes con los esperados y se especifican en la sección B.4.3.7.
8	Ejecutar el método init_random de la clase CCromosoma	Topología=2,2,1      Máximo=0.0001 Mínimo=-0.0001	Red neuronal con la topología especificada y con valores de pesos variando dentro del rango.	Los resultados obtenidos son coincidentes con los esperados y se especifican en la sección B.4.3.7.
9	Ejecutar el método run de la clase CAlgoritmoEntrenamiento	Un objeto de tipo CAlgoritmoEntrenamiento con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.	Los resultados obtenidos son coincidentes con los esperados y se especifican en la sección B.4.3.7.
10	Ejecutar el método evaluar_ejemplos de la clase CRRNN	Un objeto de tipo CRRNN con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.	Los resultados obtenidos son coincidentes con los esperados y se especifican en la sección B.4.3.7.



ÍTEM	ACCIÓN	ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO
11	Ejecutar el método run de la clase CAlgoritmoEntrenamiento	Un objeto de tipo CAlgoritmoEntrenamiento con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.	Los resultados obtenidos son coincidentes con los esperados y se especifican en la sección B.4.3.7.
12	Ejecutar el método run de la clase CAlgoritmoEntrenamiento	Un objeto de tipo CAlgoritmoEntrenamiento con los valores especificados en la sección B.4.3.7.	Los resultados esperados se especifican en la sección B.4.3.7.	Los resultados obtenidos son coincidentes con los esperados y se especifican en la sección B.4.3.7.

Tabla B.6. Informe de los casos de prueba ejecutados

**B.4.3.6. Informe de la prueba**

Tras una serie de iteraciones, donde se fueron detectando y corrigiendo errores, las pruebas resultaron satisfactorias para los 12 casos estudiados, ya que en todos los casos se obtuvieron los resultados especificados.

Por lo tanto, se concluye que el resultado de las pruebas es excelente, basándose en los criterios definidos en la sección B.4.3.2.

Se dan por cumplimentadas las pruebas y se anexa en la sección B.4.3.7 el detalle de las 12 pruebas realizadas.

**B.4.3.7. Anexo con pruebas realizadas**

<b>Ítem</b>	1
<b>Acción</b>	Transformar un TPeso en un TCromosoma
<b>Entrada</b>	-15.1
<b>Resultado Esperado</b>	0xDA00C580
<b>Resultado Obtenido</b>	Valor de entrada: -15.100000 Valor obtenido: DA00C580

**Tabla B.7.** Detalle de la prueba N° 1

<b>Ítem</b>	2
<b>Acción</b>	Transformar un TCromosoma en un TPeso
<b>Entrada</b>	0xDA00C580
<b>Resultado Esperado</b>	-15.1
<b>Resultado Obtenido</b>	Valor de entrada: DA00C580 Valor obtenido: -15.100000

**Tabla B.8.** Detalle de la prueba N° 2

<b>Ítem</b>	3
<b>Acción</b>	Ejecutar la función de morfogénesis
<b>Entrada</b>	Cantidad de capas=3 Topología= Backpropagation [2,2,1] Cromosoma=[0.25,0.2,-0.1,-1,0.5,-0.5,1.2,0.001,-1]
<b>Resultado Esperado</b>	<div>Peso 2,0 = 0.25                      Peso 3,0 = 0.2</div> <div>Peso 2,1 = -0.1                      Peso 3,1 = -1</div> <div>Umbral 2= 0.5                      Umbral 3= -0.5</div> <div>Peso 4,2 = 1.2                      Peso 4,3 = 0.001</div> <div>Umbral 4= -1</div>
<b>Resultado Obtenido</b>	<b>NEURONA 0</b> Peso 2,0 = 0.250000 Peso 3,0 = 0.200000  <b>NEURONA 1</b> Peso 2,1 = -0.100000 Peso 3,1 = -1.000000

	<i>Umbral 2= 0.500000</i> <i>Umbral 3= -0.500000</i> <i>NEURONA 2</i> <i>Peso 4,2 = 1.200000</i>  <i>NEURONA 3</i> <i>Peso 4,3 = 0.001000</i>  <i>Umbral 4= -1.000000</i>
--	---

**Tabla B.9.** Detalle de la prueba N° 3

<b>Ítem</b>	4
<b>Acción</b>	Ejecutar el método de cruza de la clase CFuncionCruzaUniforme
<b>Entrada</b>	Cromo. 1= (0.25,0.2,-0.1,-1,0.5,-0.5,1.2,0.001,-1) Cromo. 2= (-1, -1, -1, -1, -1, -1, -1, -1, -1) Topología=(2,2,1) Cantidad de capas=3
<b>Resultado Esperado</b>	Cualquier cruza valida, tomando los puntos de cruza posibles. Los puntos de cruza posibles, para esta topología son los siguientes: Puntos de cruza (0,2,4,6,7,8,9)
<b>Resultado Obtenido</b>	0.25 0.20 -1.00 -1.00 0.50 -0.50 -1.00 -1.00 -1.00 -1.00 -1.00 -0.10 -1.00 -1.00 -1.00 1.20 0.00 -1.00

**Tabla B.10.** Detalle de la prueba N° 4

<b>Ítem</b>	5
<b>Acción</b>	Ejecutar el método de cruza de la clase CFuncionCruzaBinomial
<b>Entrada</b>	Cromo. 1= (0.25,0.2,-0.1,-1,0.5,-0.5,1.2,0.001,-1) Cromo. 2= (-1, -1, -1, -1, -1, -1, -1, -1, -1) Topología=(2,2,1) Cantidad de capas=3
<b>Resultado Esperado</b>	Cualquier cruza valida, tomando los puntos de cruza posibles. Los puntos de cruza posibles, para esta topología son los siguientes: Puntos de cruza (0,2,4,6,7,8,9)
<b>Resultado Obtenido</b>	0.25 0.20 -1.00 -1.00 -1.00 -1.00 0.00 -1.00 -1.00 -1.00 -0.10 -1.00 0.50 -0.50 1.20 -1.00 -1.00

**Tabla B.11.** Detalle de la prueba N° 5

<b>Ítem</b>	6
<b>Acción</b>	Ejecutar el método de cruza de la clase CFuncionCruzaSimple
<b>Entrada</b>	Cromo. 1= (0.25,0.2,-0.1,-1,0.5,-0.5,1.2,0.001,-1)

	Cromo. 2= (-1, -1, -1, -1, -1, -1, -1, -1, -1) Topología=(2,2,1) Cantidad de capas=3
<b>Resultado Esperado</b>	Cualquier cruza valida, tomando los puntos de cruza posibles. Los puntos de cruza posibles, para esta topología son los siguientes: Puntos de cruza (0,2,4,6,7,8,9)
<b>Resultado Obtenido</b>	0.25 0.20 -0.10 -1.00 0.50 -0.50 1.20 -1.00 -1.00 -1.00 -1.00 -1.00 -1.00 -1.00 -1.00 -1.00 0.00 -1.00

Tabla B.12. Detalle de la prueba N° 6

<b>Ítem</b>	7
<b>Acción</b>	Ejecutar el método de cruza de la clase CFuncionCruzaMultiPuntos
<b>Entrada</b>	Cromo. 1= (0.25,0.2,-0.1,-1,0.5,-0.5,1.2,0.001,-1) Cromo. 2= (-1, -1, -1, -1, -1, -1, -1, -1, -1) Topología=(2,2,1) Cantidad de capas=3
<b>Resultado Esperado</b>	Cualquier cruza valida, tomando los puntos de cruza posibles. Los puntos de cruza posibles, para esta topología son los siguientes: Puntos de cruza (0,2,4,6,7,8,9)
<b>Resultado Obtenido</b>	0.25 0.20 -1.00 -1.00 0.50 -0.50 -1.00 -1.00 -1.00 -1.00 -1.00 -0.10 -1.00 -1.00 -1.00 1.20 0.00 -1.00

Tabla B.13. Detalle de la prueba N° 7

<b>Ítem</b>	8
<b>Acción</b>	Ejecutar el método init_random de la clase cromosoma
<b>Entrada</b>	Topología=(2,2,1) Cantidad de capas=3 Valor máximo de inicialización: 0.0001 Valor mínimo de inicialización: -0.0001
<b>Resultado Esperado</b>	Una red neuronal con la topología indicada, cuyos pesos sinápticos iniciales se encuentren dentro del rango determinado.
<b>Resultado Obtenido</b>	<p><i>NEURONA 0</i></p> <p><i>Peso 2,0 = -0.000027</i></p> <p><i>Peso 3,0 = -0.000071</i></p> <p><i>NEURONA 1</i></p> <p><i>Peso 2,1 = -0.000067</i></p> <p><i>Peso 3,1 = 0.000098</i></p> <p><i>Umbral 2= -0.000011</i></p> <p><i>Umbral 3= -0.000076</i></p>

	<p><i>NEURONA 2</i>  <i>Peso 4,2 = -0.000099</i></p> <p><i>NEURONA 3</i>  <i>Peso 4,3 = -0.000098</i></p> <p><i>Umbral 4= -0.000024</i></p>
--	---

Tabla B.14. Detalle de la prueba N° 8

<b>Ítem</b>	9
<b>Acción</b>	Ejecutar el método run de la clase CalgoritmoEntrenamiento
<b>Entrada</b>	<p>Topología=(2,2,1)  Cantidad de capas=3  Ejemplo (1,0,0)  Red inicial:  Peso 2,0 = 0.250000      Peso 3,0 = 0.200000  Peso 2,1 = -0.100000      Peso 3,1 = -1.000000  Umbral 2= 0.500000      Umbral 3= -0.500000  Peso 4,2 = 1.200000      Peso 4,3 = 0.001000  Umbral 4= -1.000000</p>
<b>Resultado Esperado</b>	<p>Red neuronal tras la aplicación del algoritmo backpropagation. Los valores de los pesos tras una aplicación del algoritmo son los siguientes:  Peso 2,0 = 0.250000      Peso 3,0 = 0.200000  Peso 2,1 = -0.126370      Peso 3,1 = -1.000023  Umbral 2= 0.526370      Umbral 3= -0.499977  Peso 4,2 = 1.214637      Peso 4,3 = 0.013386  Umbral 4= -0.971465</p>
<b>Resultado Obtenido</b>	<p><i>NEURONA 0</i>  <i>Peso 2,0 = 0.250000</i>  <i>Peso 3,0 = 0.200000</i></p> <p><i>NEURONA 1</i>  <i>Peso 2,1 = -0.126370</i>  <i>Peso 3,1 = -1.000023</i></p> <p><i>Umbral 2= 0.526370</i>  <i>Umbral 3= -0.499977</i></p> <p><i>NEURONA 2</i>  <i>Peso 4,2 = 1.214637</i></p> <p><i>NEURONA 3</i>  <i>Peso 4,3 = 0.013386</i></p> <p><i>Umbral 4= -0.971465</i></p>

Tabla B.15. Detalle de la prueba N° 9

<b>Ítem</b>	10
<b>Acción</b>	Ejecutar el método evaluar_ejemplos de la clase CRRNN
<b>Entrada</b>	Topología=(2,2,1) Cantidad de capas=3 Ejemplos {(0,0,0); (1,0,0); (0,1,0); (1,1,1)} Red inicial: Peso 2,0 = -1.314117      Peso 3,0 = 0.741078 Peso 2,1 = -1.320710      Peso 3,1 = 0.741409 Umbral 2= -2.658615      Umbral 3= 1.676036 Peso 4,2 = -1.040693      Peso 4,3 = 0.016420 Umbral 4= -1.151829
<b>Resultado Esperado</b>	Error rms = 0.009997
<b>Resultado Obtenido</b>	<i>El error rms es: 0.009997</i>

Tabla B.16. Detalle de la prueba N° 10

<b>Ítem</b>	11
<b>Acción</b>	Ejecutar el método run de la clase CAlgoritmoEntrenamiento
<b>Entrada</b>	Topología=(2,2,1) Cantidad de capas=3 Ejemplos {(0,0,0); (1,0,0); (0,1,0); (1,1,1)} Tipo Entrenamiento: RRNN Máximo error rms: 0.01
<b>Resultado Esperado</b>	Red Neuronal entrenada con error rms<0.01
<b>Resultado Obtenido</b>	<i>Error rms = 0.009879. Red obtenida:</i> <i>NEURONA 0</i> <i>Peso 2,0 = -0.256081</i> <i>Peso 3,0 = -1.286642</i>  <i>NEURONA 1</i> <i>Peso 2,1 = -0.288126</i> <i>Peso 3,1 = -1.296435</i>  <i>Umbral 2= -0.263248</i> <i>Umbral 3= -2.497676</i> <i>NEURONA 2</i> <i>Peso 4,2 = 0.222432</i>  <i>NEURONA 3</i> <i>Peso 4,3 = -1.083380</i>  <i>Umbral 4= -1.120917</i>

Tabla B.17. Detalle de la prueba N° 11

<b>Ítem</b>	12
<b>Acción</b>	Ejecutar el método run de la clase CAlgoritmoEntrenamiento
<b>Entrada</b>	Topología=(2,2,1) Cantidad de capas=3 Ejemplos {(0,0,0); (1,0,0); (0,1,0); (1,1,1)} Tipo Entrenamiento: AGS Máximo error rms: 0.01
<b>Resultado Esperado</b>	Red Neuronal entrenada con error rms<0.01
<b>Resultado Obtenido</b>	<p><i>Error rms = 0.009938. Red obtenida:</i></p> <p><i>NEURONA 0</i>  <i>Peso 2,0 = 0.471516</i>  <i>Peso 3,0 = 8.098954</i></p> <p><i>NEURONA 1</i>  <i>Peso 2,1 = 0.477970</i>  <i>Peso 3,1 = 9.198176</i></p> <p><i>Umbral 2= 0.475004</i>  <i>Umbral 3= 3.927385</i></p> <p><i>NEURONA 2</i>  <i>Peso 4,2 = 2.109279</i></p> <p><i>NEURONA 3</i>  <i>Peso 4,3 = -0.244734</i></p> <p><i>Umbral 4= -0.436489</i></p>

Tabla B.18. Detalle de la prueba N° 12

## B.5. Mantenimiento

En esta sección se exponen algunas cuestiones referentes al mantenimiento de la aplicación. Para ello se describen algunos puntos relacionados con la gestión de configuración.

### B.5.1. Gestión de configuración

Las actividades de Gestión de Configuración que se realizarán en este proyecto son:

- Identificación de la configuración.
- Control de la configuración.
- Generación de informes de estado.

#### B.5.1.1. Identificación de la configuración

- *Nombre de la aplicación:* EasyTrain.
- *Objetivo de la Aplicación:* Entrenar redes neuronales con conexión hacia delante.
- *Ciclo de vida del Software:* Modelo de espiral.

- *Fases del Ciclo de vida*: Se enumeran las fases para este proyecto y se determinan los productos que se espera obtener al final de cada una de ellas:

### 1. *Requerimientos*

- Documento de especificación de requerimientos donde se detallan todas las funcionalidades requeridas para poder cumplir con el objetivo del sistema (Sección B.1).

### 2. *Análisis*

- Casos de uso (Sección B.2).

### 3. *Diseño*

- Especificación de plataforma, programas y diseño de archivos (Sección B.3).

### 4. *Codificación*

- Código fuente de los principales programas diseñados.

### 5. *Implementación*

- Pantallas (Sección B.4.1).
- Comandos implementados (Sección B.4.2).
- Plan de pruebas (Sección B.4.3.1).
- Documento de diseño de la prueba (Sección B.4.3.2).
- Especificación de los casos de prueba (Sección B.4.3.3).
- Especificación del procedimiento de prueba (Sección B.4.3.4).
- Informe de los casos de prueba ejecutados (Sección B.4.3.5).
- Informe de la prueba (Sección B.4.3.6).
- Anexo con pruebas realizadas (Sección B.4.3.7).

### 6. *Mantenimiento*

- Identificación de la Configuración (Sección B.5.1.1).
- Control de la Configuración (Sección B.5.1.2).
- Generación de Informes de Estado (Sección B.5.1.3).

- *Líneas Base establecidas*: Para el desarrollo del presente trabajo se han acotado las líneas base como los elementos de configuración definidos. En este caso el criterio es que, dado que es una sola persona, el tesista, quien realiza la documentación del proyecto y la programación, se ha definido una sola línea base para todo el proceso de Desarrollo, Programación e Implementación cuyas fases y elementos de configuración se detallaron anteriormente.



### **B.5.1.2. Control de configuración**

Utilizando como fuente de información, control y seguimiento una Base de Datos de Cambios, se implementa el siguiente mecanismo para el control de los mismos:

- *Generación de una solicitud de cambio*: Ante el requerimiento de un cambio funcional o un reporte de error, se completa la correspondiente solicitud.
- *Ingreso de la solicitud a la Base de Datos de cambios*: Una vez recibida la solicitud de cambio, se la ingresa en la Base de Datos de cambios.
- *Análisis de la solicitud de cambio*: Cada solicitud de cambio debe ser analizada y se debe decidir si se rechaza o se acepta el cambio. La decisión tomada queda registrada en la Base de Datos de Cambios.
- *Evaluación de la solicitud de cambio*: Si se decide la aceptación de la solicitud de cambio, se debe realizar la evaluación técnica de la misma, emitiendo un informe en donde se exprese el esfuerzo requerido para satisfacer el pedido, las repercusiones que dicho cambio genera en otros elementos, y el costo estimado. La evaluación realizada, queda registrada en la Base de Datos de Cambios.
- *Generación de la orden de cambio*: El informe generado durante la evaluación de la solicitud de cambio, se somete a análisis asignándole la prioridad y los recursos necesarios. Se emite una Orden de Cambio.
- *Realización del cambio*: Se realiza el cambio, se registra y se realiza el control de la modificación.
- *Prueba e implementación del cambio*: Se certifica que el cambio funciona correctamente y se procede a su implementación, a través de la modificación de manuales y documentos que deban reflejar el cambio.

### **B.5.1.3 Generación de informes de estado**

A continuación se detallan algunos de los informes que incluyen la Gestión de Configuración y los datos relevantes que incluye cada uno.

**B.5.1.3.1. Solicitud de cambios**

<b>Sistema:</b> EasyTrain <b>Fecha:</b> dd/mm/aaaa
<b>Pedido por:</b>
<b>Resultado de la evaluación:</b> ACEPTADO   RECHAZADO
<b>Cambio solicitado:</b>
<b>Solución propuesta:</b>
<b>Elementos del producto software afectados por el cambio:</b>
<b>Documentación anexa presentada:</b>
<b>Solución propuesta por el área de desarrollo:</b>
<b>Tiempo estimado en horas (análisis, programación y prueba e implementación):</b>
<b>Cronograma de desarrollo e implementación:</b>
<b>Plan de pruebas:</b>

**Figura B.31.** Solicitud de cambios

#### **B.5.1.3.2. Orden de cambios**

<b>Sistema:</b> EasyTrain <b>Fecha:</b> dd/mm/aaaa
<b>Pedido por:</b>
<b>Nro. solicitud de cambio:</b>
<b>Cambio a realizar:</b>
<b>Elementos del producto software afectados por el cambio:</b>
<b>Tiempo estimado en horas (análisis, programación y prueba e implementación):</b>
<b>Comentarios del cambio:</b>
<b>Fecha de entrega:</b> dd/mm/aaaa

**Figura B.32.** Orden de cambios

#### **B.5.1.3.3. Registro de cambios**

<b>Sistema:</b> EasyTrain <b>Fecha:</b> dd/mm/aaaa
<b>Pedido por:</b>
<b>Nro. solicitud de cambio:</b>
<b>Fecha de solicitud:</b> dd/mm/aaaa
<b>Descripción breve del problema:</b>
<b>Fecha de evaluación:</b> dd/mm/aaaa
<b>Resultado de la evaluación:</b> ACEPTADO   RECHAZADO
<b>Comentarios:</b>
<b>Prioridad asignada:</b> ALTA   MEDIA   BAJA
<b>Estado actual del pedido:</b>
<b>Fecha de terminación:</b> dd/mm/aaaa

**Figura B.33.** Registro de cambios

**B.5.1.3.4. Registro de instalaciones**

<b>Sistema:</b> EasyTrain <b>Fecha:</b> dd/mm/aaaa
<b>Nro. solicitud de cambio:</b>
<b>Descripción breve del problema:</b>
<b>Fecha de evaluación:</b> dd/mm/aaaa
<b>Versión instalada:</b>

**Figura B.34.** Registro de instalaciones

**B.5.1.3.5. Informe de estado de cambios**

<b>Sistema:</b> EasyTrain <b>Fecha:</b> dd/mm/aaaa	
<b>Fecha desde:</b> dd/mm/aaaa	<b>Fecha hasta:</b> dd/mm/aaaa
dd/mm/aaaa	
dd/mm/aaaa	
dd/mm/aaaa	
dd/mm/aaaa	
dd/mm/aaaa	

**Figura B.35.** Informe de estado de cambios

**B.5.1.3.6. Informe de instalación**

<b>Sistema:</b> EasyTrain <b>Fecha:</b> dd/mm/aaaa	
<b>Fecha desde:</b> dd/mm/aaaa	<b>Fecha hasta:</b> dd/mm/aaaa
dd/mm/aaaa	
dd/mm/aaaa	
dd/mm/aaaa	
dd/mm/aaaa	
dd/mm/aaaa	

**Figura B.36.** Informe de instalación