

Búsqueda de la estructura óptima de redes neurales con Algoritmos Genéticos y Simulated Annealing. Verificación con el benchmark PROBEN1

Francisco Yepes Barrera

Phase Difference Srl
Via della Consortia, 2
37127 Verona (Italia)
paco.yepes@godelia.org

Resumen

Este artículo describe el uso de algoritmos genéticos (AG) y simulated annealing (SA) en la búsqueda de configuraciones óptimas de redes neurales artificiales, dentro de una arquitectura software, TSAGANN. El estudio comparativo ha sido realizado con benchmarks consolidados y es ilustrado en detalle. El análisis estadístico de los resultados indica que SA es tan eficiente como AG para este tipo de problemas, permitiendo incluso realizar exploraciones en el espacio del problema con un menor número de evaluaciones de las usadas por el AG para obtener resultados comparables.

Palabras clave: Red Neural, Algoritmo Genético, Simulated Annealing, Backpropagation, Optimización, TSAGANN.

1. Introducción

Las redes neurales artificiales (RNA) han demostrado su eficiencia en problemas de clasificación y aproximación de funciones en general. Sus propiedades de generalización dependen de la estructura y del número de conexiones, de forma que redes poco complejas pueden comportar capacidades de generalización pobres mientras que redes demasiado complejas pueden dar lugar a fenómenos de *overfitting*. En muchos casos, la estructura óptima de una red es encontrada mediante el método de prueba y error, probando algunas y eligiendo aquella que ofrece las mejores oportunidades para

el problema estudiado. Otros métodos más elaborados prevén la progresiva simplificación de una estructura máxima (*pruning*), el aumento progresivo en complejidad de una estructura mínima (*growing*) o una mezcla de ambos métodos (*constructing*) [6, 23].

Entre los métodos más prometedores en el estudio de estructuras óptimas de RNA se encuentran los algoritmos genéticos (AG) y el simulated annealing (SA) [1, 2, 6, 7, 8, 9, 18, 24, 27, 28, 29, 30]. Los resultados expuestos en este artículo han sido obtenidos aplicando TSAGANN (*Time Series Analysis using Genetic Artificial Neural Networks*),¹ un programa proyectado para la búsqueda

¹(C) 2004-2006 Francisco Yepes Barrera. El programa ha sido escrito en C para plataformas Linux/Unix.

²TSAGANN es capaz de realizar tanto análisis y predicciones de series históricas como resolver problemas de clasificación e interpolación, como los tratados en este artículo.

queda de estructuras óptimas de RNA con AG y SA, al benchmark PROBEN1.² Los resultados obtenidos son independientes de la arquitectura software elegida y pueden, por lo tanto, considerarse de carácter general para el problema considerado.

2. Descripción de los algoritmos

2.1. Red neural

Las redes generadas por TSAGANN son de tipo *feed-forward* multi-nivel con *bias*, con interconexión completa entre niveles adyacentes. Redes de este tipo pueden aprender de un conjunto de datos y encontrar una matriz apropiada de pesos de interconexión. El aprendizaje está basado en la siguiente función cuadrática del error entre la salida generada, y , y el conjunto de datos de referencia, t

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^P (y_{i,j} - t_{i,j})^2,$$

el cual es minimizado respecto a los pesos y bias de la red [5, 17]. N y P representan respectivamente el número de unidades de salida y el número de ejemplos usados en el adiestramiento.

Como es sabido, debido al *credit assignment problem*, el adiestramiento de una red multi nivel requiere el uso de una función diferenciable y la aplicación de un algoritmo como el *backpropagation* [21, 22]. Este algoritmo proporciona un modo para el cálculo de la derivadas de la función de error respecto a los pesos, a partir de las expresiones

$$\begin{aligned} \delta_o^p &= (y_o^p - t_o^p) \mathcal{F}'(s_o^p), \\ \delta_h^p &= \mathcal{F}'(s_h^p) \sum_{i=1}^{N_i} \delta_i^p w_{hi}, \end{aligned}$$

que están ligadas a las derivadas de la función de error a través de las ecuaciones

$$\begin{aligned} \delta_k^p &= -\frac{\partial E^p}{\partial s_k^p}, \\ \frac{\partial s_k^p}{\partial w_{jk}} &= y_j^p, \\ \frac{\partial E^p}{\partial w_{jk}} &= \frac{\partial E^p}{\partial s_k^p} \frac{\partial s_k^p}{\partial w_{jk}} = \delta_k^p y_j^p, \end{aligned}$$

donde el superíndice p hace referencia al ejemplo p -ésimo del conjunto de datos y los subíndices h

y o son relativos a unidades presentes respectivamente en un nivel oculto o en el nivel de salida. La función de activación \mathcal{F} es función de la suma pesada de las salidas de las unidades del nivel precedente a las cuales la unidad considerada, k , está conectada

$$s_k^p = \sum_j w_{jk} y_j^p. \quad (1)$$

La suma de la ecuación (1) incluye el *bias*, el término independiente, a través de un peso w_{0k} que conecta la unidad k con una unidad adicional que tiene salida $y_0 = 1$.

Las derivadas del error calculadas son usadas para actualizar los pesos al tiempo $t+1$ a través del método del *gradiente descendiente*

$$\Delta w_{jk}(t+1) = -\eta \frac{\partial E^p}{\partial w_{jk}} + \alpha \Delta w_{jk}(t), \quad (2)$$

donde se ha considerado explícitamente el término de *momento*, usado para reducir las oscilaciones en el espacio de los errores y que depende de la variación del peso calculado en el tiempo precedente. TSAGANN actualiza los pesos en modo secuencial o *pattern-based* [5], es decir, después de la presentación de cada ejemplo.

El error final calculado para el conjunto de datos es dado por el *error cuadrático porcentual* (ECP) [19]:

$$ECP = \frac{100}{NP(y_{max} - y_{min})^2} \sum_{i=1}^N \sum_{j=1}^P (y_{i,j} - t_{i,j})^2, \quad (3)$$

siendo y_{max} y y_{min} los valores máximo y mínimo posibles para la salida en la representación del problema (ver Tabla 3). La ecuación (3) proporciona un valor porcentual para el error que es independiente de la dimensión de los datos y del intervalo de valores permitidos para la salida.

Como criterio de parada para el algoritmo de *backpropagation* se utiliza la parada anticipada (*early stopping*), efectuada cuando el valor de la *pérdida de generalización*

$$G_L^{(k)}(t) = \frac{\sum_{t' \in t-k+1 \dots t} E(t')}{k \cdot E_{min}} \quad (4)$$

supera un cierto valor límite. El valor de G_L es calculado al final de k épocas. E_{min} es el valor mínimo absoluto del error encontrado hasta el momento. Los valores de los parámetros utilizados se encuentran en la Tabla 3.

La función de activación puede ser de dos tipos: la función logística o la tangente hiperbólica, en las formas

$$\mathcal{F}_1 = \frac{1}{1+e^{-\beta s}} \left. \vphantom{\mathcal{F}_1} \right\}, \quad \mathcal{F}_2 = \tanh(\beta x) = \frac{e^{\beta s} - e^{-\beta s}}{e^{\beta s} + e^{-\beta s}} \quad (5)$$

con s dada por la ecuación (1), y donde β es un parámetro a optimizar en función de las características del problema. Todas las unidades de la red tienen la misma función de activación.

2.2. Algoritmo genético

Los AG son métodos adaptativos que pueden ser usados para resolver problemas de búsqueda y optimización. Matemáticamente, son procedimientos estocásticos de orden cero, que involucran el cálculo de los valores de una función dada pero no el de sus derivadas. Una primera distinción importante entre AG de naturaleza diferente puede hacerse en base al *reemplazamiento generacional*, g_g , (*generation gap*) [3, 4], es decir, la fracción de individuos que en cada generación son sustituidos. En este sentido, los AG pertenecen a una de dos categorías extremas: aquéllos con $g_g = 1$ que comprenden los AG estrictamente generacionales, y aquéllos con g_g mínimo, de uno o dos individuos (en base al operador aplicado). El AG usado por TSAGANN es de tipo *steady-state* con g_g mínimo.

2.2.1. Operadores

TSAGANN utiliza cuatro operadores: mutación, crossover a dos puntos, crossover uniforme (*uniform crossover*) e inversión. Los crossover son aplicados a dos elementos de la población cada vez (generando dos descendientes), mientras la mutación y el operador de inversión son aplicados a un único cromosoma. El alfabeto usado para la codificación del genotipo es binario.

El operador mutación (Figura 1a) es el descrito habitualmente en la literatura técnica [3, 4]. Cada bit se cambia de 0 a 1 o viceversa con una probabilidad igual a 0.1.

El crossover [3, 4] consiste en el intercambio de una porción del genotipo entre dos elementos de la población elegidos casualmente. La Figura 1c ilustra el crossover clásico, donde se elige aleatoriamente un punto de corte y son intercambiadas las porciones de cromosoma situadas entre este

punto y un extremo. TSAGANN usa una variante de este operador, llamada crossover a dos puntos, donde se eligen dos puntos de corte y se intercambia la región situada entre ellos. Esta configuración constituye una generalización de la variante clásica, la cual se obtiene cuando uno de los puntos de corte seleccionados coincide con el bit inicial o final del cromosoma.

El crossover uniforme (Figura 1d) consiste en el intercambio de los bits entre dos cromosomas padre en base a un patrón binario de la misma longitud que es generado aleatoriamente. En nuestro caso, son intercambiados los bits cuando en el patrón se encuentra un 1 en la misma posición y dejados inalterados cuando se encuentra un 0. Este operador intercambia en media el 50% de los bits del cromosoma.

El operador de inversión (Figura 1b) consiste en la inversión de los bits dentro de una región situada entre dos puntos de corte elegidos aleatoriamente.

2.2.2. Espacio de búsqueda

El espacio de búsqueda es doble. Por una parte, se explora el espacio de las redes para el problema. En concreto, se minimiza el ECP dado por la ecuación (3) con el objetivo de encontrar un conjunto de redes lo más adherente posible a los datos del problema y con las más altas capacidades de generalización. Por otra parte, se realiza una exploración en el espacio de los operadores de forma que en cada momento del cálculo el algoritmo es capaz de calcular las probabilidades con las cuales deben ser aplicados los cuatro operadores [12, 25]. Eso se realiza añadiendo al final de cada cromosoma una sección de dos bits con los cuales es posible codificar cada una de las cuatro posibilidades. El operador a aplicar en cada momento del cálculo se elige con probabilidad igual a los porcentajes de los cromosomas de la población que codifican para cada operador, mecanismo conocido con el nombre de *adaptación global* [25].

2.2.3. Función objetivo

El valor adaptativo de cada cromosoma (*score*) [12, 13] no es directamente el ECP del fenotipo, sino una función directamente relacionada con este valor. Hemos comprobado que el uso de la varianza media relativa (ARV, *Average Relative Variance*) permite aumentar el intervalo de variabi-

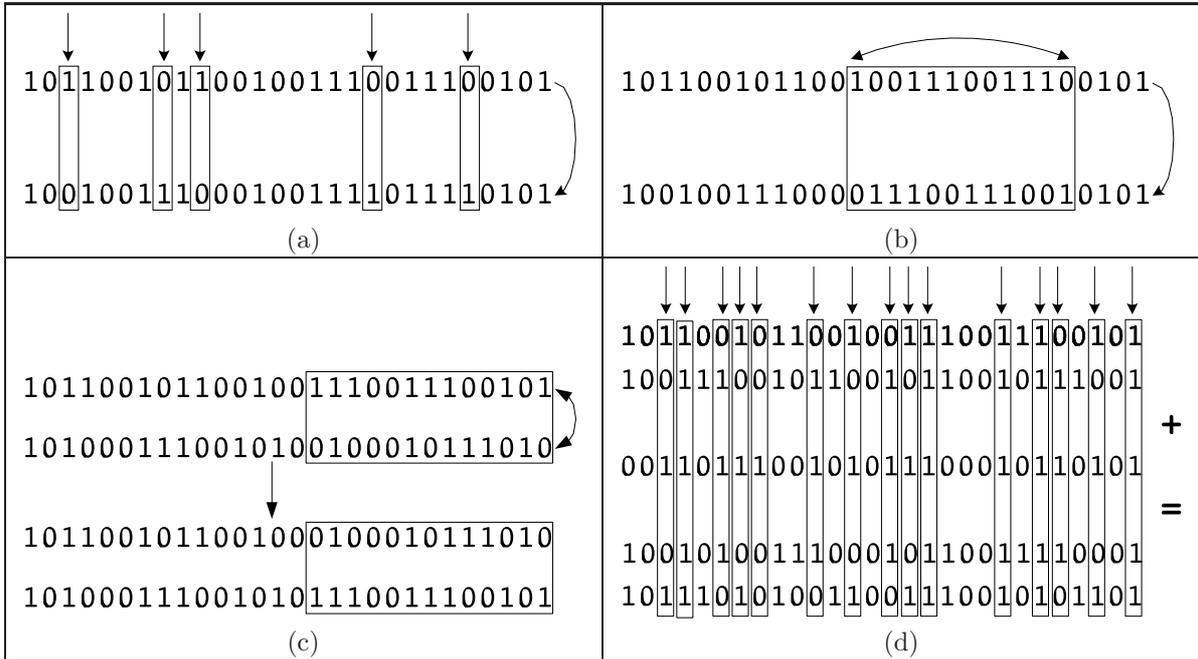


Figura 1: Operadores usados en el algoritmo genético: (a) mutación, (b) inversión, (c) crossover, (d) crossover uniforme. Para la mutación y el crossover uniforme, las flechas en alto indican el bit modificado, en base a la probabilidad de mutación puntual en el primer caso y al patrón binario en el segundo. Para inversión y crossover, el área del recuadro indica la porción de cromosoma a la que se aplica el operador.

lidad y facilitar el trabajo del AG respecto al uso directamente del ECP, reduciendo fenómenos de saturación en el cálculo de los errores.

El ARV, para una única salida, tiene la forma

$$ARV = \frac{\sum_i^P (t_i - y_i)^2}{\sum_i^P (t_i - \bar{t})^2}, \quad (6)$$

con t_i el valor objetivo para el i -ésimo ejemplo, \bar{t} el valor medio del objetivo y y_i el valor calculado. En el caso de un problema con múltiples objetivos (redes con más de una unidad de salida), se usa un *ARV generalizado*, definido de la siguiente forma

$$f = \frac{\sum_i^P \sum_j^N (t_{i,j} - y_{i,j})^2}{\sum_i^P \sum_j^N (t_{i,j} - \bar{t})^2}, \quad (7)$$

con

$$\bar{t} = \frac{1}{NP} \sum_i^P \sum_j^N t_{i,j}.$$

2.2.4. Nichos (*niching*)

Para evitar fenómenos de convergencia prematura de la población, se utiliza un mecanismo de nichos para mantener la diversidad de la población. Los nichos introducen una corrección al valor de la

función objetivo f_k calculada en base a la similitud entre el individuo k y el resto de la población, según la siguiente ecuación:

$$f_k' = \frac{f_k}{\sum_l \phi(d_{kl})}, \quad (8)$$

siendo $\phi(d_{kl})$ una función de la diversidad entre los individuos k y l , y donde d_{kl} es dada por la distancia de Hamming. La función $\phi(d_{kl})$ tiene la forma

$$\phi(d_{kl}) = \begin{cases} 1 - \left(\frac{d_{kl}}{R}\right)^\gamma, & \text{para } d_{kl} < R \\ 0, & \text{para } d_{kl} \geq R \end{cases}, \quad (9)$$

con R el radio del nicho, que es la distancia de Hamming máxima para la cual los individuos k y l son considerados similares. Es decir, todos los elementos de la población que se encuentran dentro del radio R contribuyen a la suma situada en el denominador de la ecuación (8), penalizando la similitud dentro de la población y evitando, por lo tanto, fenómenos de convergencia prematura. Los valores usados de los parámetros relativos al mecanismo de nichos se encuentran en la Tabla 3.

El mecanismo de sustitución consiste en la eliminación del cromosoma con el valor más bajo de f_k' cuando se encuentra una solución que tiene $f_k' > (f_k')_{min}$. TSAGANN sigue un criterio elitista, preservando los cromosomas con el máximo valor de f_k .

2.2.5. Código de Gray

El código binario tiene dos problemas que implican discontinuidades en la codificación de los parámetros y que, por lo tanto, dificultan el trabajo del AG. Estas dos dificultades son las siguientes [10]:

1. a veces es necesario cambiar todos los bits para incrementar o decrementar el número representado de una unidad;³
2. cambios por mutación de los bits más significativos tiene un “impacto” mayor en el fenotipo que los cambios en los bits menos significativos.

Un modo de resolver sobre todo el primer problema es el de usar un código diferente, como el código de Gray, en vez del código binario. El código de Gray asegura que el cambio de una unidad en el fenotipo de un parámetro o gen sea acompañado del cambio de un único bit en su representación binaria o genotipo.

La transformación entre las representaciones binaria, b , y de Gray, g , para el i -ésimo bit, considerando números compuestos por n bits que son numerados de derecha a izquierda, con el bit más significativo, n , situado a la izquierda, es dada por las expresiones

$$g_i = \begin{cases} b_i & \Rightarrow i = n \\ b_{i+1} \otimes b_i & \Rightarrow i < n \end{cases},$$

$$b_i = \begin{cases} g_i & \Rightarrow i = n \\ b_{i+1} \otimes g_i & \Rightarrow i < n \end{cases},$$

siendo \otimes el operador XOR.

2.3. Simulated annealing

El simulated annealing [16, 20] tiene a la base una analogía con la termodinámica, en concreto con el hecho de que la naturaleza, en un proceso de enfriamiento de un sólido suficientemente lento, consigue encontrar el estado de mínima energía, y que es el de cristal perfectamente ordenado. A medida que el sólido se enfría, la energía del sistema tiende a disminuir y se convierte en el estado más probable, como expresado por la función de distribución de Boltzmann.

Desde el punto de vista de esta analogía, es posible definir un “estado energético” y una “temperatura” para el problema considerado en cada momento. El problema, entonces, consiste en el cálculo de la probabilidad de transición del sistema desde un estado de energía E_1 a un estado E_2 , la cual es dada por

$$p = e^{-(E_2-E_1)/kT}. \quad (10)$$

El criterio de Metropolis define los cambios permitidos en el estado del sistema:

1. si $E_2 > E_1$, el nuevo estado es aceptado con una probabilidad igual a p ;
2. si $E_2 < E_1$, la probabilidad es mayor que la unidad, en cuyo caso el cambio es aceptado siempre.

El estado energético se define en función del valor de la función objetivo para la solución examinada, y los diferentes estados se generan por la aplicación repetida del operador mutación a un único cromosoma inicialmente generado aleatoriamente. La variación de temperatura sucede cuando se acepta una nueva solución (se satisface el criterio de Metropolis) o cuando se ha llegado, para una cierta temperatura, a un número límite de ciclos de cálculo. Los valores de los parámetros utilizados para el SA se encuentran en la Tabla 3.

El SA usa, como el AG, el código de Gray en la codificación del genotipo.

3. El benchmark PROBEN1

PROBEN1 [19] es una colección de problemas preparados para el adiestramiento de RNA con el objetivo de poner a prueba algoritmos diferentes y consentir una comparación directa de los resultados. PROBEN1 contiene 15 problemas distribuidos en 12 ámbitos diferentes, y reporta también una serie de reglas y convenciones aconsejadas por cuanto se refiere a la documentación de los resultados obtenidos. Los problemas cubren argumentos dentro de problemas de clasificación e interpolación funcional.

Los problemas de clasificación son los siguientes:

1. **cancer**: diagnosis del cáncer de mama en base a la descripción y al examen al microscopio de células;

³Por ejemplo, el número 3 es codificado en binario como 011, pero el 4 es 100.

2. **card**: predicción de la aprobación o no de una carta de crédito a un cliente;
3. **diabetes**: diagnosis de diabetes en los indios Pima en base a una serie de datos personales y clínicos de los pacientes;
4. **gene**: identificación de fronteras intrón/exón en secuencias de nucleótidos;
5. **glass**: clasificación del tipo de vidrio en base a la composición química de 8 elementos constitutivos;
6. **heart**: predicción de enfermedades del corazón, en concreto si uno de los cuatro mayores vasos es reducido en diámetro de más del 50%, en base a datos personales y hábitos de vida de los pacientes. El problema **heartc** es igual excepto por el hecho que contiene sólo una parte de los datos presentes en **heart**;
7. **horse**: predicción del destino (muerte, supervivencia o eutanasia) de caballos afectados por cólicos en base a un examen veterinario;
8. **mushroom**: discrimina entre hongos venenosos y no venenosos en base a características morfológicas y ambiente. Es el único problema artificial presente en PROBEN1;
9. **soybean**: clasificación de 19 diferentes enfermedades en semillas de soja;
10. **thyroid**: diagnosis de hiper o hipo funcionalidad de la tiroides, en base a una serie de datos provenientes del examen de los pacientes.

La clase es condicionada por la unidad con la salida más elevada (*winner-takes-all*). Los problemas de aproximación son:

1. **building**: predicción del consumo energético en los edificios (consumo horario) en base a la fecha, hora y datos meteorológicos;
2. **flare**: predicción de erupciones solares (número en las sucesivas 24 horas) en base a la actividad registrada en precedencia;
3. **hearta**, **heartac**: como los problemas **heart** y **heartc** con la diferencia que son utilizados para predecir la entidad de la reducción del vaso (sustitución de la salida binaria original por una continua).

Cada conjunto de datos está dividido en tres particiones: datos de adiestramiento (50% de los datos) utilizados para el adiestramiento propiamente dicho, datos de validación (25% de los datos) sobre los que se comprueba la validez del resultado obtenido con los datos de adiestramiento y datos de verificación (25% de los datos) que contiene datos no usados durante la fase de adiestramiento y que da una idea de las capacidades de generalización de la red. Cada problema tiene tres versiones diferentes, que consisten en diferentes permutaciones de los datos que forman los conjuntos de adiestramiento, validación y verificación. Más detalles se encuentran en el trabajo original de Prechelt [19].

Las Tablas 1 y 2 contiene las características del ingreso respectivamente para los problemas de clasificación y aproximación. Los datos de salida son normalizados entre y_{min} y y_{max} (ver Tabla 3) por medio de una transformación lineal, para evitar la saturación de los nodos de salida. El nivel de salida usa siempre la función logística, en modo de obtener valores en el intervalo (0, 1) y usar, de acuerdo con la normalización efectuada, todo el intervalo de variabilidad de la función.

4. Estructura de los tests

Los tests realizados sobre TSAGANN tienen dos partes bien diferenciadas. En la primera parte (Test A) se ha usado el algoritmo de optimización (AG o SA) para realizar una búsqueda automática de la estructura y de los parámetros matemáticos óptimos de la red neural utilizando un número bajo de épocas para el adiestramiento (ver Tabla 3), en modo de permitir la exploración de un número suficientemente grande de posibilidades en tiempos razonables. En la segunda parte (Test B) se ha usado el mejor resultado encontrado en el Test A (la configuración con el valor más bajo del ECP en los datos de validación) con un número de épocas superior, de manera de obtener el mejor resultado posible con las configuraciones óptimas obtenidas con el Test A.

4.1. Test A: búsqueda de la estructura óptima de la red

El algoritmo de optimización ha sido configurado para realizar una búsqueda de la combinación óptima de los siguiente parámetros:

1. **coeficiente del término de momento:** el coeficiente α de la ecuación (2). Intervalo de variabilidad [0:1];
2. **exponente de la función de activación:** el coeficiente β de la ecuación (5). Intervalo de variabilidad [0:1];
3. **razón de aprendizaje:** el coeficiente η de la ecuación (2). Intervalo de variabilidad [0:1];
4. **función de activación:** función logística o tangente hiperbólica;
5. **número de niveles:** situado entre un mínimo de 2 (redes que comprenden exclusivamente los niveles de entrada y salida) y un máximo de 5 (nivel de entrada, nivel de salida y 3 niveles ocultos);
6. **número de unidades en cada nivel oculto:** intervalo de variabilidad [1:10]. El número de unidades en los niveles de entrada y salida es dado por la naturaleza del problema en consideración.

Los parámetros adicionales de funcionamiento del AG, SA y de la RNA se encuentran en la Tabla 3. Para cada problema se han realizado 50 simulaciones. Los resultados del ECP sobre las particiones de validación y de verificación y del error de clasificación se encuentran en la Tabla 5. La Tabla 4 muestra los resultados originales de Prechelt con el algoritmo RPROP que permiten una comparación directa con los resultados obtenidos por TSAGANN,⁴ comparación detallada en la Tabla 10. Se observe que la red neural ha sido entrenada con un bajísimo número de épocas (máximo 20 si no se cumple antes el criterio $G_L^{(5)} > 1,5$, ver Tabla 3).

El problema **mushroom** ha dado ECP = 0% en todas las simulaciones, por lo que ha sido eliminado de los resultados expuestos en las tablas.

4.2. Test B: exploración de las arquitecturas generadas por el Test A con un elevado número de épocas

Para refinar los resultados se han realizado 50 simulaciones adicionales usando las arquitecturas y

los parámetros matemáticos óptimos encontrados en el Test A en base a la configuración con el más bajo valor del ECP de los datos de validación, pero usando como criterio de parada $G_L^{(5)} > 5,0$ o un número máximo de épocas igual a 3000. No se han usado los pesos óptimos encontrados en el Test A, con lo cual el conjunto inicial de los pesos ha sido de nuevo generado aleatoriamente. Los resultados obtenidos se resumen en las Tablas 6 y 7.

5. Resultados y discusión

De una primera comparación entre las Tablas 4, 5, 6 y 7 se puede observar que TSAGANN ha obtenido un ECP medio en la partición de verificación más bajo que RPROP en 21 de los 42 problemas (50%). Este análisis ha considerado el ECP más bajo obtenido en los datos de verificación para todos los resultados presentes en la Tabla 4 (considerando, por tanto, sea redes lineales que multi nivel) y en las Tablas 5, 6 y 7 (el conjunto de los Tests A y B). Obviamente, este análisis sobre los valores medios es sólo indicativo, dado que no se ha hecho ningún análisis estadístico que tenga en cuenta la variabilidad de los resultados individuales, como se ha hecho en la comparación de los resultados entre AG y SA (Tabla 9). Las diferencias en las medias obtenidas pueden ser no significativas.

En relación a la tipología de los problemas, RPROP obtiene resultados mejores para todos los problemas de aproximación excepto que para el problema **flare2**. Teniendo en cuenta la observación sobre la falta de elementos para llegar a una conclusión sobre la significatividad de las diferencias en las medias, podemos plantear algunas hipótesis sobre las razones de este resultado:

1. una mayor eficiencia intrínseca del algoritmo RPROP, como definido en el trabajo de Prechelt [19], respecto a TSAGANN para este tipo de problemas;
2. el uso por parte de TSAGANN de un número de unidades ocultas demasiado bajo (limitado a 10 en cada nivel oculto). Se puede esperar una mejora en la calidad de los resultados extendiendo la búsqueda a configuraciones con un número más elevado de unidades ocultas;

⁴Estos resultados son relativos a redes no recurrentes, con conexiones exclusivamente entre niveles adyacentes, y cuyas arquitecturas están basadas en la estructura pivote identificadas por Prechelt [19].

3. utilización, en la búsqueda de las arquitecturas óptimas del Test A, de un número excesivamente bajo de épocas en el proceso de adiestramiento, igual a 20, que puede condicionar la calidad de los resultados incluso cuando este parámetro es aumentado en el Test B.

TSAGANN se ha demostrado en general más eficiente en la resolución de los problemas de clasificación.

En relación al análisis comparativo entre los dos algoritmos de optimización utilizados, se ha aplicado el t-test para distribuciones a dos colas (*two tailed*) con la intención de verificar si las diferencias observadas en las medias del ECP para las particiones de verificación son significativas. Dado que el ECP tiene una forma aproximadamente lognormal, se ha tomado el logaritmo natural del ECP de forma que sea satisfecho el requisito de normalidad para la aplicación del t-test. Para tener en cuenta la diferencia en las condiciones experimentales, el t-test ha sido aplicado separadamente a los resultados obtenidos en los Tests A (Tabla 5) y B (Tablas 6 y 7), de forma de entender cuál de los dos algoritmos entre AG y SA se ha demostrado más eficiente. Se ha tenido en cuenta la razón entre las varianzas de los datos a través del F-test. El t-test revela diferencias significativas en las medias, con un nivel de confianza del 95 %, en los casos evidenciados en negrita en la Tabla 9; al lado de este valor se indica cuál algoritmo se ha revelado más eficiente. Los resultados, expuestos en las Tablas 11 y 12, se resumen a continuación:

1. los coeficientes de correlación entre los errores de las particiones de validación y de verificación son en general bajos (incluso negativos en algunos casos). Esto no implica necesariamente malos resultados en el valor medio de ECP de la partición de verificación, como se puede observar en las Tablas 5, 6 y 7. Las diferencias en los coeficientes de correlación de las tres permutaciones de cada problema son significativas en algunos casos;
2. la diferencia en los valores medios del ECP de la partición de verificación es significativa sólo en 21 de los 84 casos estudiados (25 %): 7 casos de 42 para el Test A (16.7 %) y 14 de 42 para el Test B (33.3 %). Las diferencias son por lo tanto más significativas entre los resultados relativos a la mejor configuración (Test B);

3. para el Test A, el AG se ha demostrado más eficiente en 3 de los 7 casos con medias estadísticamente significativas (42.9 %), mientras en el Test B lo ha sido en 6 de los 14 casos (42.9 %). En total, se tienen 9 casos de 21;
4. desde el punto de vista de la tipología, de los 4 problemas de aproximación (**building**, **flare**, **hearta** y **heartac**) sólo **building** ha obtenido diferencias significativas en alguna ocasión (en dos casos, resueltos de forma más eficiente por el AG). En los problemas de clasificación se tiene una preferencia del SA, de acuerdo con cuanto se ha discutido precedentemente; en concreto un 66.6 % para el Test A y un 61.6 % para el Test B de los problemas con diferencias significativas en las medias han sido resueltos de forma más eficiente por el SA.

La conclusión de esta discusión es que SA y AG pueden ser considerados aproximadamente igualmente eficientes en la resolución de problemas de clasificación y aproximación, con una preferencia para el SA, sobre todo teniendo en cuenta las consideraciones que siguen en relación al número de soluciones exploradas por ambos algoritmos. La Tabla 8 detalla el mejor resultado obtenido para cada problema en el conjunto de los Test A y B, medido en base al valor más bajo obtenido del ECP para la partición de validación.

Otra consideración importante puede hacerse a raíz de un detalle significativo en los resultados de la Tabla 8, y que consiste en el hecho que no siempre el Test B (que usa las arquitecturas óptimas obtenidas en el Test A con un mayor número de épocas en el proceso de adiestramiento) ha dado el mejor resultado, como se podría esperar. En 15 de los 42 problemas (35.7 %), el mejor resultado del Test A (medido en la partición de validación) ha dado el ECP más bajo. La razón de este resultado puede buscarse en el número de configuraciones que han sido estudiadas en ambos tests. Se puede hacer el siguiente razonamiento sobre el número de estructuras elaboradas en el Test A:

- en el AG, suponiendo que cada de los 4 operadores en media ha sido usado el 25 % de las veces, teniendo en cuenta que son realizados 1000 ciclos de cálculo (ver Tabla 3) y que en cada ciclo se genera 1 nuevo individuo para mutación e inversión y 2 para los crossover, obtenemos que para las 50 simulaciones de cada problema ha sido estudia-

das en media $50 \times 1000 \times (2 \times 0,25 + 2 \times 2 \times 0,25) = 75000$ soluciones;

- en el SA, teniendo en cuenta que $T_0 = 1,0$ K, $T_f = 10^{-6}$ K, que cada variación de temperatura supone una disminución del 10%,⁵ que da lugar a 133 temperaturas diferentes posibles, y que son realizados en media 5 ciclos de cálculo para cada temperatura (el máximo es 10), tenemos $50 \times 133 \times 5 = 33250$ soluciones exploradas.

En el Test B, sin embargo, son exploradas solo 50 configuraciones posibles para cada problema. La diferencia en la cantidad de configuraciones estudiadas justifica el hecho que AG y SA tienen buenas probabilidades de encontrar un conjunto de pesos mejores respecto al obtenido en el Test B, incluso usando parámetros que dan lugar a tiempos de cálculo reducidos. Estas consideraciones refuerzan también el valor de los resultados relativos al SA, que estudia en media sólo el 44.3% de las configuraciones exploradas por el AG, pero que produce resultados comparables en relación al conjunto de los problemas del benchmark PROBEN1.

La última consideración está en relación con los bajos valores de la desviación estándar del ECP para la partición de verificación (Tablas 5, 6 y 7), que constituyen una indicación de la convergencia de los resultados obtenidos por los algoritmos de optimización.

6. Conclusión y trabajo futuro

TSAGANN es un programa de gran flexibilidad para la búsqueda de la estructura y de los parámetros óptimos de RNA. El estudio efectuado demuestra la convergencia de los resultados con los datos presentes en la literatura científica, ya sea usando AG o SA. Estos resultados evidencian que el SA es un algoritmo tan eficiente como el AG, incluso realizando un menor número de evaluaciones, confirmando trabajos de algunos autores sobre su validez en otros problemas de optimización [26].

La comparación entre AG y SA puede ser realizada más en profundidad. Un análisis de sensibilidad de ambos algoritmos respecto a otros pa-

rámetros no considerados explícitamente en este trabajo podría dar indicaciones sobre su funcionamiento, sea nivel teórico que experimental. El estudio de los resultados de estos algoritmos con la variación del exponente de la función $\phi(d_{kl})$ en la ecuación (9) o de la probabilidad de pointmutation, por ejemplo, podría consentir la elaboración y puesta a punto de protocolos más eficientes en la resolución del problema de la búsqueda de estructuras óptimas de RNA. En este cuadro puede también considerarse el efecto que algunos mecanismos de preprocesamiento, sobre todo el análisis en componentes principales (PCA, *Principal Components Analysis*), tienen en los resultados finales. Desde este punto de vista, será objeto de un estudio futuro verificar si el efecto de compresión de los datos que se puede obtener con el PCA, con la consecuente reducción de la cantidad de información a tratar, puede tener un efecto benéfico en algunos de los problemas que han dado los errores más elevados.

Este trabajo no habla de las causas de los resultados obtenidos, y se limita a la exposición y al análisis estadístico de los mismos. Un estudio de las razones a la base de los resultados obtenidos debe pasar por un análisis de la complejidad de los problemas del benchmark PROBEN1 [11]. El primer paso puede ser elaborar algunos índices que permitan correlacionar los valores de los errores con la complejidad intrínseca de los problemas. Si una correlación de este tipo puede hacerse, es posible utilizar este resultado para analizar la complejidad del proceso de exploración del espacio del problema con AG y SA; una complejidad intrínseca menor del SA respecto al AG podría explicar los resultados comparativos que hemos expuesto en la Sección 5. Un candidato para este índice puede ser la correlación fitness-distancia (FDC, *Fitness Distance Correlation*) [14, 15].

Por último, sería interesante incluir en la optimización también los pesos de las redes, además de la estructura y de los parámetros matemáticos de las mismas [24, 27]. Esto tendría la ventaja que la matriz óptima de los pesos sería obtenida por el mismo algoritmo de optimización, por lo que el mecanismo de backpropagation no sería necesario, aumentando la velocidad de evaluación de cada solución candidata; como desventaja sería necesaria la evaluación de un número más elevado de redes. Una alternativa a este programa es la evolución de la estructura de conexiones de la red; en este caso aparecen problemas especiales, como la posible generación de redes inconsistentes

⁵ $T_2 = 0,9 \cdot T_1$.

[28, 29].

El autor está convencido que la acumulación de todas estas evidencias experimentales puede suministrar los datos necesarios para poder confeccionar un marco teórico que permita, al menos de forma cualitativa, dar indicaciones sobre el algoritmo a utilizar en la búsqueda de RNA óptimas para un problema dado, así como la factibilidad en la aplicación de los algoritmos considerados en este artículo a problemas de optimización en general.

Referencias

- [1] Yas Abbas Alsultanny and Musbah M. Aqel. Pattern recognition using multilayer neural-genetic algorithm. *Neurocomputing*, 51:237–247, 2003.
- [2] Karthik Balakrishnan and Vasant Honavar. Properties of genetic representations of neural architectures. In *Proceedings of the World Congress on Neural Networks*, pages 807–813, 1995.
- [3] David Beasley, David R. Bull, and Ralph R. Martin. An overview of genetic algorithms: Part 1 fundamentals. *University Computing*, 15(2):58–69, 1993.
- [4] David Beasley, David R. Bull, and Ralph R. Martin. An overview of genetic algorithms: Part 2 research topics. *University Computing*, 15(4):170–181, 1993.
- [5] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [6] Erick Cantú-Paz. Pruning neural networks with distribution estimation algorithms. In Erick Cantú-Paz et al., editor, *Genetic and Evolutionary Computation Conference, GECCO 2003*, pages 790–800, 2003.
- [7] Erick Cantú-Paz and Chandrika Kamath. Evolving neural networks for the classification of galaxies. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002*, pages 1019–1026, San Francisco, 2002. Morgan Kaufmann Publishers.
- [8] Erick Cantú-Paz and Chandrika Kamath. Evolving neural networks to identify bent-double galaxies in the FIRST survey. *Neural Networks*, 16(3-4):507–517, 2003.
- [9] Erick Cantú-Paz and Chandrika Kamath. An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems. In *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, pages 915–927, 2005.
- [10] Stephanie Forrest. Genetic algorithms: Principles of natural selection applied to computation. *Science*, 261:872–878, August 1993.
- [11] Stephanie Forrest and Melanie Mitchell. What makes a problem hard for a genetic algorithm? Some anomalous results and their explanation. *Machine Learning*, 13:285–319, 1993.
- [12] John H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, first edition, 1992.
- [13] John H. Holland. Algoritmi genetici. *Le Scienze*, 289:50–57, September 1992.
- [14] Terry Jones and Stephanie Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In Larry Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 184–192, San Francisco, CA, 1995.
- [15] Terry Jones and Stephanie Forrest. Genetic algorithms and heuristic search. In *Proceedings of International Joint Conference on Artificial Intelligence*, 1995.
- [16] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [17] Ben Kröse and Patrick van der Smagt. *An introduction to neural networks*. University of Amsterdam, 1996.
- [18] Helmut A. Mayer, Reinhold Huber, and Roland Schwaiger. Lean artificial neural networks - regularization helps evolution. In *Proceedings of the 2NWGA, Second Nordic Workshop on Genetic Algorithms*, 1996.
- [19] Lutz Prechelt. Proben1 - a set of neural network benchmark problems and benchmarking rules. Technical Report 21/94, Fakultät für Informatik, Universität Karlsruhe, 76128 Karlsruhe, Germany, September 1994.

- [20] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C, The Art of Scientific Computing*. Cambridge University Press, second edition, 1995.
- [21] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning internal representations by error propagation*, volume Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Vol. 1. 1986. pp. 318-362.
- [22] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagation errors. *Nature*, 323:533–536, 1986.
- [23] Wolfram Schiffmann. Encoding feedforward networks for topology optimization by simulated evolution. In *Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Systems and Allied Technologies*, volume 1, pages 361–364, 2000.
- [24] Randall S. Sexton, Robert E. Dorsey, and John D. Johnson. Towards global optimization of neural networks: a comparison of the genetic algorithm and backpropagation. *Decision Support Systems*, 22(2):171–185, 1998.
- [25] William M. Spears. Adapting crossover in evolutionary algorithms. In *Proceedings of the Evolutionary Programming Conference*, pages 367–384, 1995.
- [26] Diomidis D. Spinellis and Chrissoleon T. Papadopoulos. Production line buffer allocation: Genetic Algorithms versus Simulated Annealing. In *Second International Aegean Conference on the Analysis and Modelling of Manufacturing Systems*, pages 89–101, Tinos, Greece, 1999. University of the Aegean, Department of Business Administration.
- [27] Xin Yao. A review of evolutionary artificial neural networks. *International Journal of Intelligent Systems*, 8:539–567, 1993.
- [28] Xing Yao. Evolving artificial neural networks. In *Proceedings of the IEEE*, volume 87, pages 1423–1447, September 1999.
- [29] Xing Yao and Yong Liu. Towards designing artificial neural networks by evolution. *Applied Mathematics and Computation*, 91(1):83–90, 1998.
- [30] Ping Zhang, Brijesh Verma, and Kuldeep Kumar. Neural vs. statistical classifier in conjunction with genetic algorithm based feature selection. *Pattern Recognition Letters*, 26(7):909–919, 2005.

Tabla 1: Problemas de clasificación del benchmark PROBEN1 con el número de atributos binarios (b), continuos (c) y nominales (n) de los problemas contenidos, y número de entradas binarias (b) y continuas (c) usadas en la red, así como entradas usadas para representar los valores no presentes (m). Las últimas tres columnas contienen el número de clases, número de ejemplos y entropía de los datos en bits por ejemplo [19].

Problema	Atributos				Entrada				Clases	Ejemplos	E
	b	c	n	tot.	b	c	m	tot.	b		
cancer	0	9	0	9	0	9	0	9	2	699	0.93
card	4	6	5	15	40	6	5	51	2	690	0.99
diabetes	0	8	0	8	0	8	0	8	2	768	0.93
gene	0	0	60	60	120	0	0	120	3	3175	1.50
glass	0	9	0	9	0	9	0	9	6	214	2.18
heart	1	6	6	13	18	6	11	35	2	920	0.99
heartc	1	6	6	13	18	6	11	35	2	303	1.00
horse	2	13	5	20	25	14	19	58	3	364	1.32
mushroom	0	0	22	22	125	0	0	125	2	8124	1.00
soybean	16	6	13	35	46	9	27	82	19	683	3.84
thyroid	9	6	6	21	9	6	6	21	3	7200	0.45

Tabla 2: Problemas de aproximación del benchmark PROBEN1 con el número de atributos binarios (b), continuos (c) y nominales (n) para cada uno de los problemas contenidos, y número de entradas binarias (b) y continuas (c) usadas en la red así como entradas usadas para representar valores no presentes (m). Se especifica, además el número de unidades de salida y de ejemplos [19].

Problema	Atributos				Entrada				Salidas	Ejemplos
	b	c	n	tot.	b	c	m	tot.	c	
building	0	6	0	6	8	6	0	14	3	4208
flare	5	2	3	10	22	2	0	24	3	1066
hearta	1	6	6	13	18	6	11	35	1	920
heartac	1	6	6	13	18	6	11	35	1	303

Tabla 3: Parámetros característicos del funcionamiento de los algoritmos usados en los Tests A y B.

Parámetro	Descripción	Valor
AG		
Población	Número de elementos de la población.	50
R	Radio del nicho.	0.85
γ	Exponente de la ecuación (9).	1.0
Parada	Número de ciclos de cálculo.	1000
P_{mut}	Probabilidad de mutación de cada bit (<i>point-mutation</i>).	0.1
SA		
T_0	Temperatura inicial.	1.0 K
T_f	Temperatura final.	10^{-6} K
Ciclos	Número máximo de ciclos a ejecutar para cada T .	10
Decrecimiento	Fracción de decremento en cada variación de T .	10 %
RNA		
$G_L^{(k)}$	Criterio de parada en el proceso de “parada anticipada”, ec. (4).	1.5 (Test A) 5.0 (Test B)
k	<i>Strip length</i> (ecuación (4)).	5
Épocas	Número máximo de épocas antes de la parada forzada.	20 (Test A) 3000 (Test B)
y_{max}	Valor máximo de la salida normalizada.	0.9
y_{min}	Valor mínimo de la salida normalizada.	0.1
Pesos	Intervalo usado en la generación inicial.	[-0.5:0.5]

Tabla 4: Resultados obtenidos por Prechelt con el algoritmo RPROP sea para redes lineales sin niveles ocultos que para redes multi nivel. Los valore del ECP de las particiones de validación y verificación y de los errores de clasificación son los valores medios de 10 simulaciones para redes lineares y 60 simulaciones para redes multi nivel [19].

Problema	Lineal				Multi nivel			
	Arquit.	Valid.	Verif.	Clas.	Arquit.	Valid.	Verif.	Clas.
building1	14 3	0.920	0.780	-	14 16 3	2.07	1.36	-
building2	14 3	0.370	0.350	-	14 16 8 3	0.30	0.28	-
building3	14 3	0.380	0.380	-	14 16 8 3	0.26	0.26	-
cancer1	9 2	2.910	3.520	2.930	9 4 2 3	1.89	1.32	1.38
cancer2	9 2	3.770	4.770	5.000	9 8 4 2	1.76	3.47	4.77
cancer3	9 2	4.230	4.110	5.170	9 16 8 2	2.83	2.60	3.70
card1	51 2	8.890	10.610	13.370	8 32 2	8.69	10.35	14.05
card2	51 2	10.800	14.910	19.240	8 24 2	10.87	14.94	18.91
card3	51 2	8.390	12.670	14.420	8 16 8 2	8.62	13.47	18.84
diabetes1	8 2	16.300	16.300	25.830	8 32 2	15.93	16.99	24.10
diabetes2	8 2	17.470	17.470	24.690	8 16 8 2	16.94	18.43	26.42
diabetes3	8 2	18.210	18.210	22.920	8 32 2	17.89	16.48	22.59
flare1	24 3	0.340	0.520	-	24 32 3	0.35	0.54	-
flare2	24 3	0.460	0.310	-	24 32 3	0.47	0.32	-
flare3	24 3	0.460	0.350	-	24 24 3	0.47	0.36	-
gene1	120 3	9.580	9.920	13.640	120 4 2 3	8.19	8.66	16.67
gene2	120 3	9.900	9.510	12.300	120 4 2 3	9.46	9.54	18.41
gene3	120 3	9.360	10.610	15.410	120 4 2 3	9.45	10.84	21.82
glass1	9 6	9.700	9.980	46.040	9 16 8 6	9.15	9.24	32.70
glass2	9 6	10.280	10.340	55.280	9 16 8 6	10.03	10.09	55.57
glass3	9 6	9.370	11.070	60.570	9 16 8 2	9.14	10.74	58.40
heart1	35 2	13.280	14.290	20.650	35 32 2	13.10	14.19	19.72
heart2	35 2	12.220	13.520	16.430	35 32 2	12.32	13.61	17.52
heart3	35 2	10.770	16.390	22.650	35 32 2	10.85	16.79	24.08
hearta1	35 1	4.420	4.470	-	35 32 1	4.48	4.55	-
hearta2	35 1	4.280	4.190	-	35 16 1	4.41	4.33	-
hearta3	35 1	4.140	4.540	-	35 32 1	4.46	4.89	-
heartc1	35 2	9.650	16.120	19.730	35 2 2	8.08	16.99	20.82
heartc2	35 2	16.510	6.340	3.200	35 8 4 2	16.86	5.05	5.13
heartc3	35 2	13.880	12.530	14.270	35 16 8 2	14.30	13.79	15.40
heartac1	35 1	4.700	2.690	-	35 16 8 1	4.77	2.47	-
heartac2	35 1	5.210	3.870	-	35 8 8 1	5.16	4.41	-
heartac3	35 1	5.660	5.430	-	35 32 1	5.74	5.55	-
horse1	58 3	15.530	12.930	26.700	58 16 8 3	15.47	13.32	29.19
horse2	58 3	15.990	17.430	34.840	58 16 8 3	16.07	17.68	35.86
horse3	58 3	15.590	15.500	32.420	58 32 3	15.91	15.86	34.16
soybean1	82 19	0.980	1.160	9.470	82 16 8 19	1.94	2.10	29.40
soybean2	82 19	0.810	1.050	4.240	82 32 19	0.59	0.79	5.14
soybean3	82 19	0.960	1.030	7.000	82 16 19	0.93	1.25	11.54
thyroid1	21 3	3.780	3.840	6.560	21 16 8 3	1.01	1.28	2.38
thyroid2	21 3	3.550	3.710	6.560	21 8 4 3	0.89	1.02	1.91
thyroid3	21 3	3.390	4.020	7.230	21 16 8 3	0.98	1.26	2.27

Tabla 5: Resultados obtenidos por TSAGANN para el Test A con AG y SA. La Tabla reporta la media y la desviación estándar del ECP de las particiones de validación y verificación, y del error de clasificación, calculado en base a 50 simulaciones.

Problema	AG						SA					
	Validación		Verificación		Clasificación		Validación		Verificación		Clasificación	
	\bar{E}	σ	\bar{E}	σ	\bar{E}	σ	\bar{E}	σ	\bar{E}	σ	\bar{E}	σ
building1	2.261	0.105	1.599	0.189	-	-	2.264	0.099	1.570	0.222	-	-
building2	1.054	0.011	0.962	0.016	-	-	1.056	0.014	0.966	0.017	-	-
building3	0.966	0.010	0.971	0.013	-	-	0.970	0.010	0.976	0.013	-	-
cancer1	1.581	0.061	0.990	0.114	1.241	0.269	1.590	0.057	0.991	0.160	1.241	0.269
cancer2	1.167	0.050	3.703	0.270	4.195	0.372	1.166	0.074	3.748	0.321	4.276	0.437
cancer3	2.537	0.111	2.347	0.266	2.690	0.562	2.538	0.105	2.372	0.282	2.747	0.496
card1	7.334	0.101	10.220	0.332	14.267	1.320	7.340	0.096	10.186	0.332	14.407	1.250
card2	9.450	0.179	13.554	0.522	18.884	0.790	9.550	0.091	13.465	0.352	18.988	0.594
card3	5.836	0.261	13.662	1.019	18.174	1.480	5.924	0.233	13.269	0.890	17.953	1.753
diabetes1	15.415	0.092	16.312	0.182	23.698	0.723	15.432	0.114	16.340	0.245	23.885	1.190
diabetes2	16.518	0.128	17.392	0.389	25.115	10.69	16.473	0.139	17.331	0.503	25.281	1.424
diabetes3	17.982	0.087	15.882	0.233	22.500	1.153	17.936	0.132	15.778	0.193	22.260	1.258
flare1	0.407	0.006	0.657	0.015	-	-	0.406	0.006	0.652	0.020	-	-
flare2	0.460	0.004	0.296	0.012	-	-	0.462	0.004	0.300	0.012	-	-
flare3	0.529	0.005	0.430	0.011	-	-	0.531	0.006	0.428	0.012	-	-
gene1	5.369	0.118	6.094	0.206	11.624	0.595	5.594	0.193	6.183	0.353	11.579	0.773
gene2	5.549	0.098	5.558	0.274	10.646	0.785	5.746	0.188	5.652	0.325	10.719	0.882
gene3	5.231	0.124	6.533	0.288	12.507	0.766	5.511	0.207	6.683	0.283	12.767	0.634
glass1	9.004	0.125	9.626	0.388	47.623	7.248	9.023	0.127	9.568	0.359	47.736	7.275
glass2	9.051	0.188	9.835	0.216	49.585	4.017	9.090	0.210	9.955	0.285	50.151	3.926
glass3	8.942	0.143	10.163	0.479	48.453	9.410	8.944	0.132	10.231	0.408	50.604	8.723
heart1	12.510	0.095	14.679	0.364	21.870	0.816	12.523	0.113	14.664	0.344	21.670	0.927
heart2	11.641	0.157	13.351	0.394	17.009	1.071	11.638	0.143	13.286	0.366	16.791	0.993
heart3	9.913	0.203	17.027	0.501	23.661	1.047	9.950	0.147	16.943	0.686	23.713	1.560
hearta1	6.509	0.078	7.131	0.242	-	-	6.515	0.061	7.132	0.252	-	-
hearta2	6.335	0.076	6.754	0.220	-	-	6.335	0.079	6.783	0.182	-	-
hearta3	6.143	0.120	7.217	0.201	-	-	6.181	0.097	7.238	0.180	-	-
heartc1	7.844	0.178	17.002	0.732	21.710	1.166	7.839	0.162	16.845	0.856	21.658	1.253
heartc2	14.262	0.188	4.786	0.621	6.158	1.648	14.345	0.347	4.647	0.614	5.789	1.861
heartc3	11.249	0.369	12.998	1.012	16.474	1.903	11.154	0.383	13.131	1.145	16.684	2.032
heartac1	6.467	0.224	3.930	0.316	-	-	6.505	0.174	3.814	0.368	-	-
heartac2	5.384	0.290	9.194	1.071	-	-	5.251	0.291	9.177	0.991	-	-
heartac3	7.759	0.283	8.643	0.458	-	-	7.768	0.178	8.540	0.309	-	-
horse1	14.451	0.180	12.190	0.307	26.435	2.360	14.419	0.158	12.225	0.307	26.674	2.198
horse2	14.813	0.180	16.604	0.398	36.478	1.601	14.813	0.132	16.564	0.451	36.522	1.440
horse3	14.758	0.169	14.658	0.313	32.783	1.634	14.787	0.125	14.706	0.445	33.261	2.025
soybean1	0.637	0.006	0.883	0.018	8.468	0.829	0.635	0.011	0.884	0.028	8.784	0.880
soybean2	0.563	0.010	0.758	0.019	5.205	0.491	0.569	0.006	0.750	0.037	5.322	0.339
soybean3	0.684	0.008	0.814	0.028	6.725	0.768	0.687	0.013	0.837	0.033	6.117	0.749
thyroid1	3.421	0.017	3.413	0.031	6.123	0.086	3.414	0.014	3.401	0.042	6.141	0.115
thyroid2	2.950	0.022	3.243	0.050	5.990	0.140	2.941	0.013	3.225	0.028	5.915	0.124
thyroid3	2.690	0.024	3.369	0.043	6.419	0.049	2.693	0.024	3.351	0.033	6.438	0.098

Tabla 6: Resultados obtenidos por TSAGANN para el Test B con AG. La Tabla contiene la media y la desviación estándar del ECP de las particiones de validación y verificación, y del error de clasificación, calculado en base a 50 simulaciones, así como la arquitectura seleccionada en base al menor ECP de los datos de validación obtenido en el Test A.

Problema	Arquitectura	Validación		Verificación		Clasificación	
		\bar{E}	σ	\bar{E}	σ	\bar{E}	σ
building1	14 3	2.088	0.035	1.505	0.168	-	-
building2	14 8 10 10 3	0.884	0.027	0.845	0.042	-	-
building3	14 8 10 7 3	0.848	0.031	0.914	0.034	-	-
cancer1	9 8 2 10 2	1.584	0.153	1.227	0.392	1.483	0.658
cancer2	9 6 1 2	1.278	0.200	3.912	0.324	4.207	0.375
cancer3	9 10 4 1 2	2.638	0.059	2.364	0.372	2.563	0.482
card1	51 10 9 2	7.630	0.205	10.337	0.282	14.326	1.185
card2	51 4 2	9.924	0.285	13.798	0.571	18.977	0.777
card3	51 6 4 2	6.367	0.353	14.581	1.667	18.965	1.698
diabetes1	8 2	15.280	0.056	16.187	0.214	22.979	0.819
diabetes2	8 4 2	16.038	0.229	18.793	0.918	26.896	1.112
diabetes3	8 9 2	17.386	0.263	15.322	0.486	21.677	1.009
flare1	24 10 5 3 3	0.415	0.007	0.670	0.050	-	-
flare2	24 9 9 10 3	0.472	0.007	0.299	0.016	-	-
flare3	24 3	0.521	0.003	0.431	0.008	-	-
gene1	120 10 8 3	5.399	0.109	6.052	0.256	11.579	0.673
gene2	120 10 9 3	5.497	0.119	5.470	0.245	10.378	0.687
gene3	120 6 3 3	5.312	0.113	6.603	0.203	12.552	0.626
glass1	9 9 9 6	7.670	0.442	9.412	0.795	36.113	5.524
glass2	9 9 7 10 6	7.955	0.543	9.808	0.655	41.698	5.527
glass3	9 6 6	8.782	0.410	9.853	0.748	41.811	9.441
heart1	35 8 2	12.772	0.225	14.628	0.414	20.757	1.413
heart2	35 6 2	11.753	0.256	13.523	0.709	17.235	1.474
heart3	35 8 2	10.085	0.300	17.427	0.888	23.122	1.266
hearta1	35 9 7 1	6.648	0.146	7.034	0.237	-	-
hearta2	35 8 1	6.180	0.155	6.711	0.285	-	-
hearta3	35 10 8 1	6.124	0.192	7.451	0.280	-	-
heartc1	35 3 10 2	7.365	0.586	17.271	1.584	20.763	1.922
heartc2	35 2 3 2	14.811	0.771	5.783	1.490	7.421	2.123
heartc3	35 10 2	12.081	0.667	14.110	2.051	17.316	3.035
heartac1	35 5 1	6.664	0.386	4.028	0.447	-	-
heartac2	35 3 1 2 1	6.033	0.464	9.149	1.110	-	-
heartac3	35 5 9 1	8.365	0.511	8.715	0.735	-	-
horse1	58 3	14.814	0.253	12.323	0.379	26.935	2.806
horse2	58 3	15.169	0.246	16.965	0.553	35.979	1.774
horse3	58 6 10 3	15.590	0.448	16.041	1.328	36.957	3.940
soybean1	82 19	0.590	0.011	0.810	0.024	7.696	0.672
soybean2	82 19	0.538	0.009	0.671	0.019	4.947	0.461
soybean3	82 19	0.638	0.010	0.758	0.022	6.351	0.669
thyroid1	21 6 9 3	1.150	0.136	1.351	0.174	2.436	0.399
thyroid2	21 10 10 3	1.282	0.187	1.489	0.261	2.732	0.565
thyroid3	21 7 7 3	0.817	0.131	1.370	0.197	2.533	0.432

Tabla 7: Resultados obtenidos por TSAGANN para el Test B con SA. La Tabla muestra la media y la desviación estándar del ECP de las particiones de validación y de verificación, y el error cometido en la clasificación, calculado en base a 50 simulaciones, así como la arquitectura seleccionada en base al menor ECP en los datos de validación obtenido en el Test A.

Problema	Arquitectura	Validación		Verificación		Clasificación	
		\bar{E}	σ	\bar{E}	σ	\bar{E}	σ
building1	14 3	2.108	0.041	1.490	0.144	-	-
building2	14 10 8 10 3	0.899	0.030	0.863	0.035	-	-
building3	14 9 9 9 3	0.842	0.030	0.901	0.033	-	-
cancer1	9 4 2 2	1.636	0.089	1.229	0.330	1.563	0.821
cancer2	9 9 1 2	1.345	0.202	3.876	0.371	4.333	0.482
cancer3	9 6 1 2	2.645	0.053	2.414	0.421	2.701	0.496
card1	51 6 2	7.594	0.148	10.260	0.277	14.291	1.210
card2	51 6 10 2	9.740	0.245	13.530	0.480	18.674	0.681
card3	51 4 2	6.383	0.287	13.539	0.802	18.198	1.351
diabetes1	8 2	15.279	0.044	16.136	0.223	22.990	0.780
diabetes2	8 5 2	16.088	0.186	19.026	0.995	27.219	1.407
diabetes3	8 10 2	17.334	0.213	15.369	0.309	21.969	0.934
flare1	24 10 7 3 3	0.413	0.007	0.679	0.050	-	-
flare2	24 4 3	0.472	0.009	0.303	0.015	-	-
flare3	24 3	0.521	0.003	0.431	0.010	-	-
gene1	120 10 7 3	5.495	0.231	6.173	0.257	11.556	0.627
gene2	120 10 7 3	5.717	0.175	5.613	0.266	10.454	0.759
gene3	120 8 10 3	5.547	0.192	6.774	0.304	12.772	0.648
glass1	9 8 9 6	7.637	0.412	9.526	0.961	36.679	4.561
glass2	9 9 8 8 6	7.295	0.508	9.213	0.884	37.472	4.330
glass3	9 10 7 6	8.412	0.458	9.052	0.806	34.566	5.960
heart1	35 2	12.465	0.081	14.279	0.306	20.017	0.950
heart2	35 10 2	11.967	0.247	14.050	0.851	18.061	1.420
heart3	35 8 8 2	9.910	0.498	18.131	1.594	24.217	1.542
hearta1	35 9 9 1	6.596	0.100	7.034	0.210	-	-
hearta2	35 9 1	6.232	0.136	6.653	0.237	-	-
hearta3	35 10 8 2	6.116	0.201	7.470	0.324	-	-
heartc1	35 8 6 2	7.131	0.428	16.780	1.237	20.447	1.787
heartc2	35 4 6 2	14.746	0.580	4.912	1.147	6.447	1.903
heartc3	35 10 2	11.962	0.646	13.848	1.470	16.763	2.221
heartac1	35 5 1	6.703	0.274	4.089	0.506	-	-
heartac2	35 3 10 1 1	5.854	0.553	9.292	1.148	-	-
heartac3	35 7 1	8.109	0.296	8.739	0.528	-	-
horse1	58 3	14.942	0.253	12.481	0.323	27.848	2.195
horse2	58 3	15.091	0.211	16.795	0.565	35.783	1.950
horse3	58 3	15.190	0.210	14.771	0.311	33.413	1.950
soybean1	82 19	0.593	0.012	0.812	0.023	7.614	0.608
soybean2	82 19	0.544	0.009	0.682	0.029	5.041	0.527
soybean3	82 19	0.639	0.012	0.754	0.026	6.456	0.614
thyroid1	21 7 9 3	1.205	0.143	1.449	0.187	2.575	0.356
thyroid2	21 9 10 3	1.095	0.145	1.311	0.163	2.487	0.334
thyroid3	21 10 5 3	0.803	0.124	1.446	0.202	2.630	0.444

Tabla 8: La Tabla muestra los mejores resultados obtenidos por TSAGANN (menor ECP en la partición de validación; en caso de más configuraciones con el mismo valor ha sido seleccionada la que tiene el valor más elevado del ECP en la partición de adiestramiento) considerando los Tests A y B. Se detallan todos los parámetros objeto de la optimización: arquitectura, coeficiente del término de momento (α en la ecuación (2)), exponente de la función de activación (β en la ecuación (5)), razón de aprendizaje (η en la ecuación (2)) y función de activación (SIGM para \mathcal{F}_1 e TANH para \mathcal{F}_2 de la ecuación (5)). La últimas dos columnas indican respectivamente el algoritmo que ha encontrado la configuración óptima y en qué Test (A o B).

Problema	Arquit.	α	β	η	\mathcal{F}	Valid.	Verif.	Clas.	Alg.	A/B
building1	14 3	0.97	0.86	0.97	TANH	2.000	1.419	-	SA	B
building2	14 10 8 10 3	0.76	0.64	0.37	TANH	0.820	0.848	-	SA	B
building3	14 9 9 9 3	0.83	0.34	0.43	TANH	0.765	0.848	-	SA	B
cancer1	9 8 2 10 2	0.68	0.96	0.38	TANH	1.136	1.850	1.724	AG	B
cancer2	9 9 1 2	0.96	0.96	0.84	TANH	1.107	3.615	4.023	SA	B
cancer3	9 10 4 1 2	0.92	0.92	0.44	TANH	2.219	2.310	2.874	AG	A
card1	51 10 9 2	0.40	0.20	0.90	TANH	7.000	10.643	15.116	AG	A
card2	51 4 2	0.80	0.60	0.40	TANH	8.160	13.028	16.860	AG	A
card3	51 6 4 2	0.30	0.50	0.10	TANH	5.020	12.870	17.442	AG	A
diabetes1	8 2	0.88	0.96	0.97	SIGM	15.105	16.087	25.000	AG	A
diabetes2	8 4 2	0.97	0.03	0.87	TANH	15.071	19.286	27.083	AG	B
diabetes3	8 10 2	0.90	0.13	0.87	TANH	16.595	15.306	21.875	SA	B
flare1	24 10 5 3 3	0.85	0.72	0.85	TANH	0.384	0.654	-	AG	A
flare2	24 9 9 10 3	0.82	0.02	0.48	TANH	0.449	0.306	-	AG	A
flare3	24 3	0.70	0.59	0.99	SIGM	0.513	0.449	-	AG	B
gene1	120 10 7 3	0.28	0.35	0.54	TANH	4.919	6.194	11.602	SA	B
gene2	120 10 7 3	0.67	0.29	0.37	SIGM	5.180	5.250	9.962	SA	B
gene3	120 6 3 3	0.21	0.46	0.55	TANH	4.872	6.341	11.349	AG	A
glass1	9 9 9 6	0.92	0.92	0.51	TANH	6.648	8.078	28.302	AG	B
glass2	9 9 8 8 6	1.00	0.31	0.26	TANH	6.227	8.558	41.509	SA	B
glass3	9 6 6	1.00	0.79	0.70	TANH	7.389	8.656	26.415	AG	B
heart1	35 8 2	0.88	0.42	0.14	TANH	12.096	14.324	18.261	AG	B
heart2	35 10 2	0.70	0.32	0.78	TANH	11.227	13.567	16.522	SA	A
heart3	35 8 8 2	0.93	0.01	0.22	TANH	8.568	18.939	24.783	SA	B
hearta1	35 9 9 1	0.60	0.12	0.75	TANH	6.278	7.483	-	SA	A
hearta2	35 8 1	0.62	0.14	0.75	TANH	5.656	7.246	-	AG	B
hearta3	35 10 8 1	0.75	0.85	0.31	TANH	5.680	7.535	-	SA	B
heartc1	35 3 10 2	0.69	0.78	0.39	TANH	5.755	17.158	17.333	AG	B
heartc2	35 2 3 2	0.70	0.30	1.00	TANH	12.227	5.687	5.333	AG	B
heartc3	35 10 2	0.98	0.13	0.70	TANH	9.873	13.643	17.333	AG	A
heartac1	35 5 1	0.96	0.92	0.80	TANH	5.674	5.789	-	AG	B
heartac2	35 3 10 1 1	0.95	0.74	0.81	TANH	4.090	11.008	-	SA	B
heartac3	35 5 9 1	0.85	0.29	0.46	TANH	6.796	8.593	-	AG	A
horse1	58 3	0.44	0.79	0.55	SIGM	13.995	11.677	23.077	SA	A
horse2	58 3	0.81	0.39	0.17	SIGM	14.172	17.195	31.868	AG	A
horse3	58 6 10 3	0.89	0.43	0.07	TANH	13.892	15.262	31.868	AG	A
soybean1	82 19	0.83	0.91	0.73	SIGM	0.568	0.783	6.471t	AG	B
soybean2	82 19	0.77	0.93	0.65	SIGM	0.518	0.686	4.706	SA	B
soybean3	82 19	0.81	0.94	0.67	SIGM	0.614	0.774	5.882	SA	B
thyroid1	21 6 9 3	0.32	0.92	0.96	TANH	0.886	1.230	2.056	AG	B
thyroid2	21 10 10 3	0.83	0.57	0.70	TANH	0.855	1.177	2.111	AG	B
thyroid3	21 10 5 3	0.99	0.57	0.25	TANH	0.593	1.188	2.111	SA	B

Tabla 9: Análisis estadístico de los resultados obtenidos en los Tests A y B. Las primeras 4 columnas contienen los valores de los coeficientes de correlación entre los ECP de las particiones de validación y verificación para los dos algoritmos utilizados. Las 2 columnas sucesivas detallan los valores del F-test, usado para ver si la diferencia de las varianzas de los ECP de las particiones de verificación entre AG y SA son significativas (necesario para poder aplicar sucesivamente el t-test); se ha usado un nivel de confianza del 95 %, con lo cual los valores por debajo de 0.05 indican diferencias estadísticamente significativas en las varianzas. El t-test evalúa diferencias significativas en las medias de los logaritmos del ECP de las particiones de verificación entre AG y SA, considerando distribuciones de dos colas y el resultado del F-test; en negrita se evidencian los resultados para los cuales las diferencias en las medias es estadísticamente significativa con un nivel de confianza del 95 % (valore inferiores a 0.05), con indicación del algoritmo más eficiente.

Problema	Test A		Test B		F-test		t-test	
	r AG	r SA	r AG	r SA	Test A	Test B	Test A	Test B
building1	0.470	0.294	0.233	0.357	0.355	0.291	0.432	0.683
building2	0.497	0.512	0.508	0.616	0.495	0.197	0.164	0.020 (AG)
building3	0.344	0.576	0.643	0.340	0.674	0.877	0.034 (AG)	0.068
cancer1	0.051	0.112	-0.256	-0.217	0.040	0.525	0.866	0.850
cancer2	0.053	0.037	0.098	0.408	0.217	0.360	0.477	0.558
cancer3	-0.018	0.037	0.054	-0.124	0.773	0.428	0.662	0.557
card1	0.129	0.367	0.238	-0.057	0.906	0.882	0.605	0.172
card2	0.479	-0.117	0.466	0.500	0.008	0.258	0.351	0.013 (SA)
card3	0.041	0.056	-0.027	0.106	0.489	0.000	0.046 (SA)	0.000 (SA)
diabetes1	0.228	0.440	0.015	0.018	0.039	0.780	0.544	0.246
diabetes2	0.227	0.103	-0.415	-0.260	0.073	0.735	0.475	0.231
diabetes3	0.325	0.095	0.096	0.096	0.211	0.002	0.017 (SA)	0.521
flare1	0.015	-0.258	-0.136	0.087	0.053	0.876	0.136	0.316
flare2	0.010	-0.050	-0.136	0.252	0.996	0.713	0.150	0.215
flare3	-0.044	-0.389	-0.408	-0.294	0.581	0.305	0.286	0.808
gene1	0.318	0.385	0.115	0.181	0.112	0.037	0.340	0.928
gene2	0.354	0.238	0.307	0.473	0.538	0.193	0.093	0.543
gene3	0.112	0.008	0.259	0.179	0.010	0.671	0.143	0.120
glass1	-0.095	0.297	0.142	0.061	0.632	0.219	0.440	0.575
glass2	0.323	0.135	0.306	0.061	0.074	0.007	0.019 (AG)	0.000 (SA)
glass3	-0.097	0.093	0.230	0.203	0.295	0.228	0.428	0.000 (SA)
heart1	0.058	-0.072	-0.019	0.149	0.709	0.050	0.832	0.000 (SA)
heart2	-0.329	-0.106	-0.137	-0.061	0.645	0.293	0.394	0.001 (AG)
heart3	-0.229	0.051	-0.106	-0.449	0.032	0.001	0.451	0.007 (AG)
hearta1	-0.243	-0.178	-0.044	-0.064	0.692	0.380	0.995	0.985
hearta2	0.305	0.291	-0.142	-0.078	0.165	0.240	0.449	0.286
hearta3	-0.015	0.038	-0.013	0.191	0.413	0.319	0.567	0.769
heartc1	0.153	0.153	0.058	-0.013	0.282	0.242	0.311	0.090
heartc2	-0.133	0.016	0.080	-0.097	0.990	0.313	0.256	0.001 (SA)
heartc3	-0.060	-0.204	0.060	-0.237	0.617	0.052	0.551	0.557
heartac1	0.468	0.364	0.112	0.286	0.164	0.580	0.075	0.528
heartac2	-0.536	-0.443	-0.067	-0.463	0.774	0.996	0.956	0.532
heartac3	0.195	0.358	-0.007	0.126	0.010	0.030	0.219	0.763
horse1	0.184	0.386	0.519	0.071	0.990	0.229	0.569	0.025 (AG)
horse2	-0.175	-0.012	0.213	0.377	0.360	0.883	0.625	0.137
horse3	-0.037	0.148	0.172	0.278	0.022	0.000	0.552	0.000 (SA)
soybean1	0.003	0.233	-0.066	0.317	0.005	0.821	0.928	0.639
soybean2	0.066	0.450	0.114	0.152	0.000	0.007	0.148	0.021 (AG)
soybean3	0.369	0.488	0.425	0.225	0.397	0.249	0.000 (AG)	0.510
thyroid1	0.085	-0.377	0.704	0.688	0.043	0.943	0.103	0.007 (AG)
thyroid2	0.356	0.310	0.834	0.784	0.000	0.008	0.037 (SA)	0.000 (SA)
thyroid3	0.687	-0.021	0.863	0.807	0.068	0.596	0.023 (SA)	0.047

Tabla 10: Tabla resumen que permite comparar los mejores resultados de RPROP [19] y TSAGANN para cada problema. La ultima columna contiene el porcentaje de error cometido por TSAGANN en su mejor resultado (error más bajo en la partición de verificación en todas las simulaciones realizadas, Tablas 5, 6 y 7), E_T , respecto al error más bajo obtenido por RPROP (Tabla 4), E_R . En negrita (valores positivos) se encuentran los resultados que mejoran los del trabajo original. [19].

Problema	RPROP			TSAGANN			$100 \times \frac{E_R - E_T}{E_R}$
	Valid.	Verif.	Clasif.	Valid.	Verif.	Clasif.	
building1	0.92	0.78	-	2.108	1.490	-	-91.0
building2	0.30	0.28	-	0.884	0.845	-	-201.8
building3	0.26	0.26	-	0.842	0.901	-	-246.5
cancer1	1.89	1.32	1.38	1.581	0.990	1.241	25.0
cancer2	1.76	3.47	4.77	1.167	3.703	4.195	-6.7
cancer3	2.83	2.60	3.70	2.537	2.347	2.690	9.7
card1	8.69	10.35	14.05	7.340	10.186	14.407	1.6
card2	10.80	14.91	19.24	9.550	13.465	18.988	9.7
card3	8.39	12.67	14.42	5.924	13.269	17.953	-4.7
diabetes1	16.30	16.30	25.83	15.279	16.136	22.990	1.0
diabetes2	17.47	17.47	24.69	16.473	17.331	25.281	0.8
diabetes3	17.89	16.48	22.59	17.386	15.322	21.677	7.0
flare1	0.34	0.52	-	0.406	0.652	-	-25.4
flare2	0.46	0.31	-	0.460	0.296	-	4.5
flare3	0.46	0.35	-	0.531	0.428	-	-22.3
gene1	8.19	8.66	16.67	5.399	6.052	11.579	30.1
gene2	9.90	9.51	12.30	5.497	5.470	10.378	42.5
gene3	9.36	10.61	15.41	5.231	6.533	12.507	38.4
glass1	9.15	9.24	32.70	7.670	9.412	36.113	-1.9
glass2	10.03	10.09	55.57	7.295	9.213	37.472	8.7
glass3	9.14	10.74	58.40	8.412	9.052	34.599	15.7
heart1	13.10	14.19	19.72	12.465	14.279	20.017	-0.6
heart2	12.22	13.52	16.43	11.638	13.286	16.791	1.7
heart3	10.77	16.39	22.65	9.950	16.943	23.713	-3.4
hearta1	4.42	4.47	-	6.648	7.034	-	-57.4
hearta2	4.28	4.19	-	6.232	6.653	-	-58.8
hearta3	4.14	4.54	-	6.143	7.217	-	-59.0
heartc1	9.65	16.12	19.73	7.131	16.780	20.447	-4.1
heartc2	16.86	5.05	5.13	14.345	4.647	5.789	8.0
heartc3	13.88	12.53	14.27	11.249	12.998	16.474	-3.7
heartac1	4.77	2.47	-	6.505	3.814	-	-54.4
heartac2	5.21	3.87	-	6.033	9.149	-	-136.4
heartac3	5.66	5.43	-	7.768	8.540	-	-57.3
horse1	15.53	12.93	26.70	14.451	12.190	26.435	5.7
horse2	15.99	17.43	34.84	14.813	16.564	36.522	5.0
horse3	15.59	15.50	32.42	14.758	14.658	32.783	5.4
soybean1	0.98	1.16	9.47	0.590	0.810	7.696	30.2
soybean2	0.59	0.79	5.14	0.538	0.671	4.947	15.1
soybean3	0.96	1.030	7.00	0.639	0.754	6.456	26.8
thyroid1	1.01	1.28	2.38	1.150	1.351	2.436	-5.5
thyroid2	0.89	1.02	1.91	1.095	1.311	2.487	-28.5
thyroid3	0.98	1.26	2.27	0.817	1.370	2.533	-8.7

Tabla 11: Tabla resumen de los resultados de significatividad estadística entre AG y SA realizados sobre los valores medios del ECP de la partición de verificación. La Tabla expone los porcentajes de problemas con diferencias significativas entre AG y SA en cada uno de los Test separadamente (referidos a los 42 problemas efectuados en cada Test) y globalmente para cada algoritmo (referidos al conjunto de los 84 problemas).

Comparación realizada	Resultados significativos					
	Test			Algoritmo		
	A	B	Medio	AG	SA	Total
Global	16.7 %	33.3 %	25.0 %	10.7 %	14.3 %	25.0 %
Problemas aproximación	2.4 %	2.4 %	2.4 %	2.4 %	0.0 %	2.4 %
Problemas clasificación	14.3 %	31.0 %	22.65 %	8.3 %	14.3 %	22.6 %

Tabla 12: Tabla resumen de los resultados de eficiencia comparativa entre AG y SA realizados sobre los resultados que han dado diferencias significativas en las medias del ECP de la partición de verificación de los dos algoritmos.

Comparación realizada	AG			SA		
	Test A	Test B	Medio	Test A	Test B	Medio
Global	42.9 %	42.9 %	42.9 %	57.1 %	57.1 %	57.1 %
Problemas aproximación	14.3 %	7.1 %	9.5 %	0.0 %	0.0 %	0.0 %
Problemas clasificación	28.6 %	35.7 %	33.3 %	57.1 %	57.1 %	57.1 %