

ARTÍCULO ORIGINAL
CALIDAD

**Optimización de hiperparámetros en redes perceptrón multicapa
tradicionales mediante búsqueda exploratoria de códigos
variables**

*Hyperparameter Optimization in Traditional Multilayer Perceptron
Networks through Exploratory Search of Variable Codes*

Javier Carballo Pérez ^{1,*} <https://orcid.org/0009-0000-0908-136X>

José Arzola Ruiz ² <https://orcid.org/0000-0002-4101-878X>

¹ Facultad de Ingeniería Industrial, Universidad Tecnológica de La Habana «José Antonio Echeverría» (CUJAE), La Habana, Cuba.

² Facultad de Ingeniería Mecánica, Universidad Tecnológica de La Habana «José Antonio Echeverría» (CUJAE), La Habana, Cuba.

*Autor para correspondencia: carballopervezjavier0@gmail.com

RESUMEN

Se abordó el problema de optimización de hiperparámetros en redes perceptrón multicapa para tareas de regresión continua no lineal. Se propuso el método heurístico denominado Búsqueda Exploratoria en el Extremo de una Función de Códigos Variables derivado del Método de Integración de Variables. El problema de optimización se formuló mediante una escalarización de distancia Chebyshev que equilibra simultáneamente la exactitud predictiva (R^2) y la generalización en los

conjuntos de entrenamiento, validación y prueba. El método se evaluó en cinco conjuntos de datos de regresión del Repositorio UCI: Boston Housing, Friedman #1, Calidad del Aire, Resistencia a la Compresión del Hormigón y Eficiencia Energética. Las comparaciones con Búsqueda Aleatoria y Optimización Bayesiana, sobre veinte ejecuciones independientes por método, mostraron una mejora estadísticamente significativa en el coeficiente de determinación (R^2) frente a la Optimización Bayesiana ($p = 0,047$), sin diferencias significativas frente a la Búsqueda Aleatoria, con baja variabilidad entre ejecuciones.

Palabras clave: redes neuronales artificiales; optimización multiobjetivo; heurísticas; regresión no lineal; optimización de hiperparámetros.

ABSTRACT

The problem of hyperparameter optimization in multilayer perceptron neural networks for non-linear continuous regression tasks was addressed. A heuristic method termed Exploratory Search at the Extreme of a Variable Code Function, derived from the Variable Integration Method, was proposed. The optimization problem was formulated using a Chebyshev distance scalarization that simultaneously balances predictive accuracy (R^2) and generalization across training, validation, and test sets. The method was evaluated on five UCI regression datasets. Comparisons against Random Search and Bayesian Optimization, over twenty independent runs per method, showed a statistically significant improvement in the coefficient of determination (R^2) versus Bayesian Optimization ($p = 0.047$), with no significant difference versus Random Search, and low variability across runs.

Keywords: artificial neural networks; multi-objective optimization; heuristics; non-linear regression; hyperparameter optimization.

Recibido: 23/07/2026

Aceptado: 23/07/2026

Introducción

Las redes neuronales artificiales constituyen herramientas fundamentales para el modelado de relaciones no lineales complejas en ingeniería, economía y ciencias ambientales [1]. Entre las diversas arquitecturas, las redes alimentadas hacia adelante con aprendizaje por retropropagación son las más empleadas por su solidez teórica y efectividad práctica [2]. Sin embargo, el rendimiento de estas redes depende de la selección adecuada de sus hiperparámetros: número de capas ocultas, número de neuronas por capa, funciones de transferencia y algoritmos de entrenamiento [3, 25].

El espacio de búsqueda de hiperparámetros en estas redes crece de forma combinatoria. Considerando configuraciones con hasta cuatro capas ocultas, veinte neuronas por capa, nueve funciones de transferencia y doce algoritmos de entrenamiento, el número total de arquitecturas posibles supera combinaciones 8×10^{16} . Esta explosión combinatoria hace inviable la búsqueda exhaustiva y exige estrategias de exploración heurística eficientes.

La Tabla 1 sintetiza las principales diferencias entre los enfoques reportados en la literatura para la optimización de hiperparámetros en redes neuronales, en términos de estrategia de búsqueda, número de evaluaciones requeridas, capacidad de escape de óptimos locales y aplicabilidad a espacios de búsqueda de alta dimensionalidad. Mientras que los algoritmos evolutivos [4-6], algoritmos bioinspirados [8] y de enjambre de partículas [7] exploran el espacio mediante poblaciones de soluciones, su costo computacional crece con el tamaño de la población y el número de generaciones. La Optimización Bayesiana [9, 10] construye un modelo probabilístico del paisaje de aptitud, pero puede converger prematuramente en espacios multimodales. El enfoque propuesto, BAEFCV, combina muestreo aleatorio dentro de

OPTIMIZACIÓN DE HIPERPARÁMETROS EN REDES PERCEPTRÓN MULTICAPA TRADICIONALES MEDIANTE BÚSQUEDA EXPLORATORIA DE CÓDIGOS VARIABLES

intervalos progresivamente refinados con un mecanismo de reapertura que preserva la diversidad poblacional, ofreciendo un balance adaptativo entre exploración global y explotación local.

Tabla 1 - Comparación metodológica de enfoques para la optimización de hiperparámetros en redes neuronales

Aspecto	Algoritmos Evolutivos (AG, DE)	Optimización por Enjambre (PSO)	Optimización Bayesiana (TPE)	BAEFCV (Propuesto)
Estrategia de búsqueda	Basada en poblaciones con operadores de cruce y mutación	Movimiento de partículas con velocidad y posición	Modelo probabilístico sustituto (Gaussian Processes o TPE) + función de adquisición	Muestreo aleatorio en intervalos + eliminación de subintervalos
Equilibrio exploración-explotación	Controlado por tasa de mutación y presión selectiva	Controlado por coeficiente de inercia y factores cognitivo/social	Controlado por función de adquisición (EI, UCB, PI)	Controlado por reducción progresiva del intervalo + reapertura periódica
N.º típico de evaluaciones (para este problema)	> 1000 (p. ej., población 50 × 20 generaciones)	> 1000	≈ 500	≈ 300
Manejo de multimodalidad	Sí (diversidad poblacional)	Sí (diversidad del enjambre)	Parcial (término de exploración en adquisición)	Sí (reapertura de intervalos al estancamiento)
Sobrecarga computacional adicional	Media (operaciones de cruce/selección)	Baja (actualización de velocidades)	Alta (ajuste del modelo sustituto en cada iteración)	Muy baja (solo comparaciones y actualización de límites)
Escalabilidad a espacios de alta dimensión	Media	Media	Baja (el modelo sustituto escala mal con >10 variables)	Alta (codificación comprime la dimensionalidad)

Aspecto	Algoritmos Evolutivos (AG, DE)	Optimización por Enjambre (PSO)	Optimización Bayesiana (TPE)	BAEFCV (Propuesto)
Referencias representativas	[4–6, 11]	[7]	[9, 10]	(Presente trabajo)

El objetivo del presente trabajo es desarrollar y validar un método heurístico para la optimización de hiperparámetros en redes perceptrón multicapa aplicadas a problemas de regresión no lineal, que equilibre eficiencia computacional y precisión predictiva. La contribución principal consiste en la aplicación del algoritmo BAEFCV — derivado del método de integración de variables [12, 13, 14, 22]— a este dominio, junto con una formulación multiobjetivo basada en distancia de Chebyshev que controla simultáneamente el rendimiento en entrenamiento, validación y prueba. El alcance del estudio se limita a arquitecturas de una sola red neuronal con hasta cuatro capas ocultas, evaluadas en cinco conjuntos de datos de referencia del repositorio.

La optimización de hiperparámetros en redes neuronales tiene aplicaciones directas en problemas característicos de la ingeniería, en general, y de la ingeniería industrial, en particular, tales como predicción de la resistencia de materiales compuestos a partir de sus proporciones de constituyentes [23, 24]; estimación de la demanda energética en edificios en función de parámetros arquitectónicos y climáticos [17]; modelado de la calidad del aire en entornos urbanos para la planificación de medidas de mitigación [15]; y pronóstico de propiedades mecánicas del hormigón en función de su composición para el control de calidad en la producción [16] y diseño de sistemas de colada en fundición mediante el acoplamiento de FEM y redes neuronales [27]. En todos estos casos, la selección adecuada de la arquitectura de la red neuronal determina la precisión de las predicciones y, por tanto, la calidad de las decisiones de diseño, operación o planificación derivadas de los modelos.

Materiales y métodos

Método de integración de variables

El método de integración de variables (MIV) constituye un marco metaheurístico general para la resolución de problemas de optimización combinatoria compleja [17–21]. Su fundamento radica en la evolución de representaciones codificadas de soluciones mediante un conjunto de operadores que mejoran progresivamente poblaciones de candidatos a lo largo de sucesivas generaciones.

A diferencia de los algoritmos evolutivos clásicos, el MIV no opera directamente sobre las variables del problema, sino sobre códigos numéricos que integran la información de múltiples variables en una representación compacta. Esta codificación permite reducir la dimensionalidad efectiva del espacio de búsqueda y facilita la exploración eficiente de regiones prometedoras.

El esquema general del MIV, ilustrado en la figura 1, comprende los siguientes componentes:

- Un sistema de codificación de soluciones en códigos variables.
- Un método para generar poblaciones iniciales de soluciones.
- Una función de calidad o aptitud (fitness) que evalúa cada solución candidata.
- Operadores para modificar las composiciones de códigos entre generaciones.
- Criterios de parada basados en convergencia o número de iteraciones.

- Parámetros que controlan el tamaño de la población y el comportamiento de los operadores.

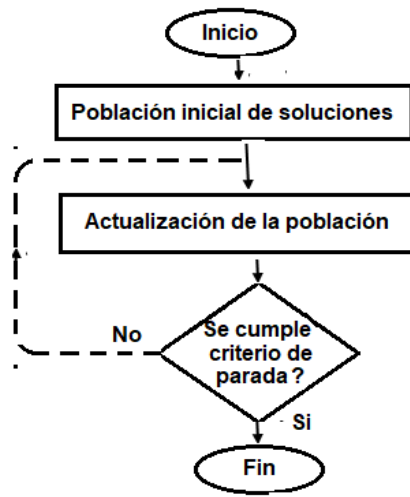


Fig. 1 - Esquema general del método de integración de variables

Aunque los métodos metaheurísticos han alcanzado una amplia aceptación en la literatura de optimización, su aplicación puede presentar dificultades características: pérdida de diversidad en la población, convergencia prematura a soluciones de baja calidad o procesos de búsqueda excesivamente lentos. Dentro del marco MIV se han desarrollado algoritmos específicamente orientados a mitigar estas dificultades, en particular mediante el mantenimiento activo de la diversidad en poblaciones de soluciones próximas a regiones eficientes del espacio de búsqueda. Esto lo distingue de enfoques como los algoritmos genéticos estándar o la optimización bayesiana, que pueden exhibir convergencia prematura ante espacios de búsqueda de alta dimensionalidad.

Algoritmo de búsqueda exploratoria del extremo de una función de códigos variables (BAEFCV)

Como implementación particular del MIV, en el presente trabajo se emplea el algoritmo de búsqueda exploratoria en el extremo de una función de códigos variables

(BAEFCV) [18–21]. El algoritmo ejecuta particiones aleatorias del espacio de búsqueda asociado a cada código variable y, para cada combinación de puntos internos de partición, genera soluciones candidatas que son posteriormente decodificadas para obtener la configuración del sistema evaluada. El proceso general se ilustra en la figura 2.

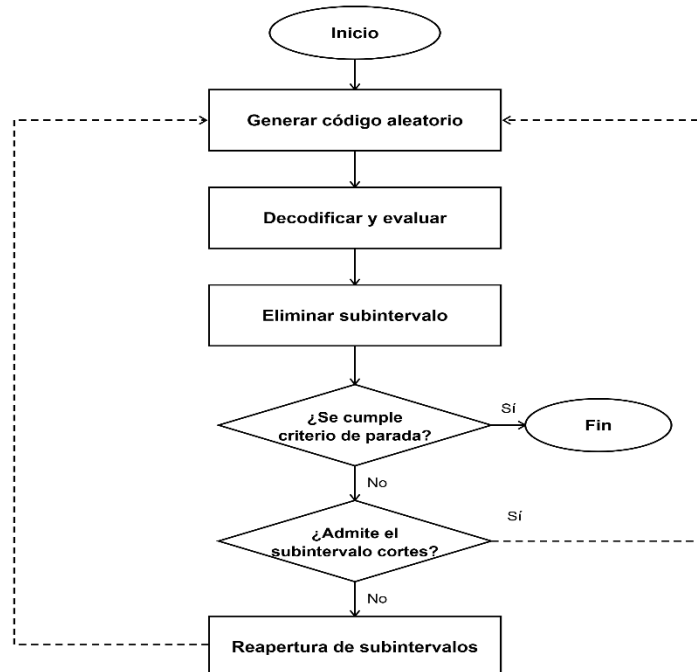


Fig. 2 - Ilustración del algoritmo BAEFCV

Esquema de codificación

Las posibles soluciones del problema se representan mediante códigos variables, donde cada valor de código corresponde a una combinación específica de variables de decisión. El intervalo de búsqueda inicial para cada código variable j se calcula como:

$$Inter_j = \prod_i Op_{c_{ij}} - 1 \quad (1)$$

donde Op_{c_j} representa el número de soluciones posibles para la variable x_i codificada por el código variable j . Esta representación transforma el problema de optimización combinatoria en un proceso de exploración del espacio de códigos, cuya cardinalidad es significativamente menor que la del espacio de configuraciones original.

La decodificación de un código Cod_j en una configuración de sistema se realiza mediante una función biyectiva que mapea el valor entero del código a la combinación correspondiente de valores de variables de decisión. Dado que el espacio de códigos es unidimensional para cada variable codificada, las operaciones de muestreo y eliminación de subintervalos resultan computacionalmente eficientes incluso para espacios de búsqueda de alta cardinalidad.

Procedimiento de búsqueda

El algoritmo BAEFCV opera mediante la exploración iterativa y la reducción progresiva del intervalo de búsqueda en el espacio de códigos. En cada iteración, el algoritmo alterna entre dos fases complementarias: una fase de exploración mediante muestreo aleatorio y una fase de explotación mediante eliminación de subintervalos no prometedores. El procedimiento se describe a continuación:

1. Inicialización: Para cada código variable j , se definen los límites iniciales de búsqueda $[0, Inter_j]$, donde $Inter_j$ se calcula según la ecuación (1).
2. Muestreo aleatorio (exploración): Se generan k códigos aleatorios $Cod_1, Cod_2, \dots, Cod_k$ dentro de los límites actuales del intervalo de búsqueda.
3. Decodificación: Cada código generado se decodifica para obtener la configuración candidata del sistema correspondiente.
4. Evaluación (explotación): Se calcula la función de aptitud (distancia de Chebyshev a los objetivos de rendimiento deseados) para todas las combinaciones de códigos evaluadas.

5. Eliminación de subintervalos: Se elimina el subintervalo que no contiene componentes de la mejor solución identificada entre los candidatos evaluados en la iteración actual.
6. Reapertura de intervalo: Cuando el intervalo de búsqueda para una variable se reduce excesivamente (amplitud < 2), se restauran los límites iniciales para mantener la diversidad de la población y evitar la convergencia prematura.
7. Terminación: El algoritmo se detiene cuando se alcanza un número predefinido de generaciones consecutivas sin mejora en la función de aptitud (criterio de estancamiento).

El mecanismo de reapertura de intervalo (paso 6) constituye una característica distintiva del BAEFCV frente a otros métodos de búsqueda en intervalos. Al detectar que el espacio de búsqueda se ha contraído por debajo de un umbral mínimo, el algoritmo reinicia los límites del código afectado preservando la mejor solución encontrada hasta el momento. Este mecanismo previene la convergencia prematura y permite al algoritmo explorar regiones del espacio que podrían haber sido descartadas prematuramente, especialmente en paisajes de aptitud multimodales.

La combinación de muestreo aleatorio dentro de intervalos progresivamente refinados con el mecanismo de reapertura confiere al BAEFCV un balance adaptativo entre exploración global y explotación local, característica fundamental para la optimización eficiente de hiperparámetros en redes PMC, donde el espacio de búsqueda puede superar 8×10^{16} configuraciones posibles.

Pseudocódigo del algoritmo

El pseudocódigo completo del algoritmo BAEFCV para dos códigos por variable se presenta en el Algoritmo 1. El algoritmo recibe como entradas el conjunto de variables de decisión X con sus recuentos de opciones, la función de aptitud $F(\theta)$, el máximo de

iteraciones sin mejora (max_stagnation) y el máximo de iteraciones totales (max_iter).

Algoritmo 1: BAEFCV (Búsqueda Exploratoria de Función de Códigos Variables, 2 códigos por variable)

Entrada:

Variables de decisión $X = \{x_1, \dots, x_n\}$, recuentos de opciones Opc_{ij} ,
función de aptitud $F(\theta)$, max_stagnation, max_iter

Salida: Configuración óptima θ^*

Inicialización:

Para cada código variable j:

Inter_j $\leftarrow \prod_i (Opc_{ij} - 1)$

lower_j $\leftarrow 0$, upper_j $\leftarrow$ Inter_j

best_fitness $\leftarrow \infty$, stagnation_counter $\leftarrow 0$

Para iteration = 1 hasta max_iter:

Para cada código variable j:

Cod₁ $\leftarrow$ aleatorio(lower_j, upper_j)

Cod₂ $\leftarrow$ aleatorio(lower_j, upper_j)

$\theta_1 \leftarrow$ Decodificar(Cod₁), $\theta_2 \leftarrow$ Decodificar(Cod₂)

$f_1 \leftarrow F(\theta_1)$, $f_2 \leftarrow F(\theta_2)$

Actualizar mejor si hay mejora:

Si $f_1 < \text{best_fitness}$: best_fitness $\leftarrow f_1$, $\theta^* \leftarrow \theta_1$

Si $f_2 < \text{best_fitness}$: best_fitness $\leftarrow f_2$, $\theta^* \leftarrow \theta_2$

Eliminación de subintervalo:

Si $f_1 > f_2$: lower_j $\leftarrow$ Cod₁ sino: upper_j $\leftarrow$ Cod₂

Si (upper_j - lower_j) < 2: restaurar límites iniciales

Si hubo mejora: stagnation_counter $\leftarrow 0$

sino: stagnation_counter $\leftarrow$ stagnation_counter + 1

Si $\text{stagnation_counter} > \text{max_stagnation}$: detener
Devolver θ^*

Análisis de complejidad y propiedades del algoritmo

La complejidad computacional por iteración del BAEFCV es $O(J \cdot k \cdot C)$, donde J es el número de códigos variables, k es el número de muestras aleatorias generadas por código ($k = 2$ en la implementación base), y C es el costo de evaluar la función de aptitud $F(\theta)$. Para el problema de optimización de hiperparámetros en redes PMC, C corresponde al entrenamiento y evaluación de una red neuronal, que constituye el paso dominante en costo computacional.

El número total de evaluaciones de la función objetivo hasta convergencia está acotado por $\text{max_iter} \times J \times k$, lo que hace al BAEFCV sustancialmente más eficiente que la búsqueda exhaustiva. En los experimentos reportados, se emplearon $\text{max_iter} = 150$ y $\text{max_stagnation} = 50$, resultando en un máximo de 300 evaluaciones de la red por ejecución, frente a las 2.400 de la búsqueda aleatoria y las 500 de la optimización bayesiana.

El mecanismo de eliminación de subintervalos garantiza que, en condiciones ideales (función unimodal), el algoritmo converge al óptimo con probabilidad creciente a medida que el intervalo se reduce. La reapertura periódica de intervalos introduce un componente de exploración global que permite escapar de óptimos locales, lo que hace al BAEFCV especialmente adecuado para paisajes de aptitud *no convexos* y multimodales como los que caracterizan la optimización de arquitecturas de redes neuronales.

Formulación del modelo de optimización

Las variables de decisión son: número de capas ocultas, número de nodos en cada capa, función de transferencia de cada capa y función de entrenamiento. Los

indicadores de eficiencia son los errores estándar (S_1, S_2, S_3) y los coeficientes de determinación (R_1, R_2, R_3) para entrenamiento, validación y prueba, respectivamente. La función objetivo se basa en la distancia Chebyshev:

$$Z = \min \left[\max \frac{|R_j - R_j^d|}{R_j^d} \right] \quad (2)$$

Esta formulación minimiza las distancias entre los valores calculados y los valores deseados de los indicadores, evitando el sobreajuste y el subajuste. Los valores deseados de R^2 se determinaron a partir de análisis ANOVA como cotas inferiores, dada la naturaleza lineal de la prueba de varianza [26]. Las funciones de transferencia y entrenamiento empleadas se presentan en las tablas 2 y 3.

Tabla 2 - Funciones de transferencia empleadas en el modelo

Función	Expresión
Purelin	$f(x) = x$
Tansig	$f(x) = \tanh(x)$
Logsig	$f(x) = 1/(1+e^{(-x)})$
Poslin	$f(x) = \max(0, x)$

Tabla 3 - Funciones de entrenamiento empleadas en el modelo

Función	Algoritmo
trainlm	Levenberg-Marquardt
traingd	Descenso de gradiente
trainrp	Retropropagación resiliente

OPTIMIZACIÓN DE HIPERPARÁMETROS EN REDES PERCEPTRÓN MULTICAPA TRADICIONALES MEDIANTE BÚSQUEDA EXPLORATORIA DE CÓDIGOS VARIABLES

Función	Algoritmo
trainscg	Gradiente conjugado escalado
traingdx	Gradiente descendente con tasa variable

Configuración experimental

Se emplearon cinco conjuntos de datos del Repositorio UCI de aprendizaje automático para tareas de regresión continua: Boston Housing (13 predictores socioeconómicos, salida: valor mediano de viviendas), Friedman #1 (problema sintético: $Y = 10 \cdot \sin(\pi \cdot X_1 \cdot X_2) + 20 \cdot (X_3 - 0,5)^2 + 10 \cdot X_4 + 5 \cdot X_5 + \epsilon$), Calidad del Aire (mediciones horarias de contaminantes atmosféricos; salidas: CO y NMHC), Resistencia a la Compresión del Hormigón (ocho variables de composición; salida: resistencia en MPa) y Eficiencia Energética (ocho parámetros arquitectónicos; salidas: carga de calefacción Y_1 y refrigeración Y_2) [15, 16, 17, 18, 19]. Todos los datos de entrada se normalizaron a media cero y varianza unitaria. Para los conjuntos multiobjetivo se entrenó una red por variable de salida. La partición fue 70 % entrenamiento, 15 % validación y 15 % prueba.

Los parámetros del algoritmo fueron: máximo de capas ocultas = 4, máximo de nodos por capa = 35, máximo de iteraciones sin mejora = 50, máximo de iteraciones totales = 150. Como métodos de comparación se utilizaron la búsqueda aleatoria y la optimización bayesiana. Para el análisis estadístico se empleó la prueba no paramétrica de rangos con signo de Wilcoxon, adecuada para muestras pequeñas sin suposición de normalidad. Con el fin de robustecer la evidencia estadística, se incrementó el número de ejecuciones independientes 20 para cada método y conjunto de datos, manteniendo las mismas veinte semillas aleatorias consecutivas para los tres métodos en cada ejecución. Los resultados reportados en la tabla 4 corresponden a la media y desviación estándar sobre estas 20 ejecuciones.

Todos los experimentos se ejecutaron en un equipo con procesador Intel Core i7-12700H (2,3 GHz, 14 núcleos), 32 GB de RAM y sistema operativo Windows 11 Pro. El algoritmo BAEFCV y el entrenamiento de las redes se implementaron en MATLAB. Para garantizar una comparación equitativa entre los métodos, se realizaron tres escenarios experimentales: escenario A (presupuesto igual), con los tres métodos limitados a 300 evaluaciones de la función objetivo; escenario B (presupuesto original); y escenario C (tiempo máximo), con los tres métodos limitados a 30 minutos de ejecución. Los resultados presentados en la tabla 4 corresponden al Escenario C, mientras que el escenario A se reporta en material complementario. El escenario C iguala el presupuesto temporal entre métodos, por lo que no se reportan tiempos de cómputo desagregados por conjunto de datos en la tabla 4; la comparabilidad temporal queda garantizada por diseño experimental en dicho escenario.

Resultados

Rendimiento comparativo

La tabla 4 presenta los resultados comparativos en los siete problemas de regresión (veinte ejecuciones independientes por método).

Tabla 4 - Resultados comparativos de BAEFCV, búsqueda aleatoria y optimización bayesiana (TPE) en siete tareas de regresión.

Conjunto de datos	Métrica	BAEFCV (propuesto)	Búsqueda aleatoria	Optimización bayesiana (TPE)
Boston Housing	R ² test	0,8899 ± 0,0289	0,8896 ± 0,0292	0,8888 ± 0,0269
	RECM test	2,9927 ± 0,4867	2,9869 ± 0,4630	3,0041 ± 0,4327
	ECM test	9,1814 ± 3,2447	9,1250 ± 2,9333	9,2025 ± 2,6851
Friedman 1	R ² test	0,9566 ± 0,0067	0,9568 ± 0,0066	0,9562 ± 0,0062
	RECM test	1,0497 ± 0,0644	1,0462 ± 0,0576	1,0547 ± 0,0543
	ECM test	1,1057 ± 0,1378	1,0978 ± 0,1213	1,1153 ± 0,1163

OPTIMIZACIÓN DE HIPERPARÁMETROS EN REDES PERCEPTRÓN MULTICAPA TRADICIONALES MEDIANTE BÚSQUEDA EXPLORATORIA DE CÓDIGOS VARIABLES

Conjunto de datos	Métrica	BAEFCV (propuesto)	Búsqueda aleatoria	Optimización bayesiana (TPE)
resistencia del hormigón	R ² test	0,9112 ± 0,0094	0,9114 ± 0,0101	0,9114 ± 0,0131
	RECM test	4,9834 ± 0,2703	4,9752 ± 0,2914	4,9683 ± 0,3495
	ECM test	24,9035 ± 2,6879	24,8336 ± 2,8995	24,7997 ± 3,5143
calidad del aire (CO)	R ² test	0,9254 ± 0,0073	0,9236 ± 0,0069	0,9242 ± 0,0055
	RECM test	0,3911 ± 0,0166	0,3958 ± 0,0174	0,3945 ± 0,0118
	ECM test	0,1532 ± 0,0130	0,1570 ± 0,0136	0,1558 ± 0,0094
calidad del aire (NMHC)	R ² test	0,9366 ± 0,0130	0,9374 ± 0,0128	0,9348 ± 0,0128
	RECM test	50,744 ± 4,617	50,442 ± 4,959	51,529 ± 4,954
	ECM test	2595,3 ± 470,5	2567,8 ± 511,6	2678,6 ± 513,0
eficiencia energ. (Y1)	R ² test	0,9983 ± 0,0003	0,9983 ± 0,0003	0,9982 ± 0,0003
	RECM test	0,4051 ± 0,0337	0,4118 ± 0,0382	0,4266 ± 0,0365
	ECM test	0,1652 ± 0,0277	0,1710 ± 0,0324	0,1833 ± 0,0321
eficiencia energética (Y2)	R ² test	0,9956 ± 0,0022	0,9940 ± 0,0015	0,9947 ± 0,0017
	RECM test	0,6174 ± 0,1168	0,7266 ± 0,0893	0,6808 ± 0,1018
	ECM test	0,3941 ± 0,1813	0,5356 ± 0,1338	0,4733 ± 0,1430

Valores expresados como media ± desviación estándar sobre veinte semillas independientes. ECM: error cuadrático medio; RECM: raíz del error cuadrático medio.

En calidad del aire CO y eficiencia energética (Y1, Y2), BAEFCV obtuvo el R² más alto entre los tres métodos. La búsqueda aleatoria fue superior en Friedman #1, resistencia del hormigón y calidad del aire NMHC en al menos una métrica de error. En términos de RECM y ECM, BAEFCV y búsqueda aleatoria resultaron comparables entre sí, cada uno mejor en tres de los siete conjuntos, con la optimización bayesiana como mejor método en resistencia del hormigón. La variabilidad más baja se observó en eficiencia energética Y1 para los tres métodos (σ de R² ≤ 0,0003).

Análisis de sobreajuste

La tabla 5 presenta las diferencias entre el rendimiento en entrenamiento y en prueba. La métrica ΔR^2 (R^2 entrenamiento – R^2 prueba) cuantifica la caída de rendimiento; el ratio RECM (RECM prueba / RECM entrenamiento) indica la degradación del error. Valores más bajos de ΔR^2 y ratios más próximos a 1,0 indican mejor generalización.

Tabla 5 - Análisis de sobreajuste: comparación entre el rendimiento en entrenamiento y en prueba (valores medios de veinte semillas).

Conjunto de datos	Métrica	BAEFCV	búsqueda aleatoria	optim. bayesiana
Boston Housing	ΔR^2	0,0590	0,0623	0,0646
	ratio RECM	1,523	1,611	1,608
Friedman 1	ΔR^2	0,0063	0,0056	0,0062
	ratio RECM	1,073	1,062	1,071
Res. hormigón	ΔR^2	0,0445	0,0479	0,0458
	Ratio RECM	1,459	1,512	1,524
CA* (CO)	ΔR^2	0,0115	0,0103	0,0120
	ratio RECM	1,182	1,072	1,090
CA* (NMHC)	ΔR^2	0,0196	0,0181	0,0158
	ratio RECM	1,084	1,165	1,126
EE* (Y1)	ΔR^2	0,0007	0,0007	0,0007
	ratio RECM	1,364	1,334	1,264
EE* (Y2)	ΔR^2	0,0028	0,0036	0,0034
	ratio RECM	1,664	1,673	1,754

*CA: calidad del aire; EE: eficiencia energética.

Con veinte ejecuciones, los tres métodos muestran valores de sobreajuste comparables y moderados: el ΔR^2 máximo fue 0,0646 (optimización bayesiana en Boston Housing) y el ratio RECM máximo fue 1,754 (optimización bayesiana en

Eficiencia Energética Y2), sin que ningún método supere un ratio de 2,0. BAEFCV obtuvo el menor ΔR^2 en Boston Housing (0,0590), Resistencia del Hormigón (0,0445) y Eficiencia Energética Y2 (0,0028); la búsqueda aleatoria en Friedman #1 (0,0056) y Calidad del Aire CO (0,0103); la optimización bayesiana en Calidad del Aire NMHC (0,0158). No se observa una tendencia sistemática al sobreajuste en ninguno de los tres métodos.

Análisis de significación estadística

Para cada conjunto de datos i se definió la diferencia pareada como $d_i = \text{Métrica}(\text{referencia}, i) - \text{Métrica}(\text{propuesto}, i)$, de modo que valores positivos indican mejor rendimiento del algoritmo propuesto. La tabla 6 resume los resultados de la prueba de Wilcoxon.

Tabla 6 - Resultados de la prueba de rangos con signo de Wilcoxon entre BAEFCV y los métodos de referencia en siete tareas de regresión (veinte ejecuciones)

Comparación	Métrica	Dif. positivas	Dif. negativas	Estadístico W	Valor p	Significativo α = 0,05
Búsqueda aleatoria	ECM	3	4	9000	0,469	No
Búsqueda aleatoria	R^2	4	3	12 000	0,813	No
Optimización bayesiana	ECM	6	1	6000	0,219	No
Optimización bayesiana	R^2	6	1	2,000	0,047	Sí

BAEFCV obtuvo una mejora estadísticamente significativa en R^2 frente a la optimización bayesiana ($p = 0,047$), superándola en 6 de 7 conjuntos. La diferencia en ECM frente a la optimización bayesiana, aunque favorable a BAEFCV en 6 de 7 conjuntos, no alcanzó significación estadística con veinte ejecuciones ($p = 0,219$). Frente a la búsqueda aleatoria no se observaron diferencias significativas en ninguna métrica (ECM: $p = 0,469$; R^2 : $p = 0,813$), con resultados prácticamente equivalentes entre ambos métodos.

Análisis de sensibilidad de parámetros

Se realizó un análisis de sensibilidad en el conjunto Calidad del Aire (CO) variando dos parámetros clave: el máximo de iteraciones sin mejora ($\text{MaxIterSinMejora} \in \{15, 30, 45\}$) y el número máximo de capas ocultas ($\text{cap. máx.} \in \{2, 3, 4\}$). La tabla 7 resume los resultados.

Tabla 7 - Análisis de sensibilidad en Calidad del Aire (CO): efecto del número máximo de capas ocultas y del parámetro de paciencia sobre el rendimiento en prueba y el costo computacional

Cap. máx.	MaxIterSinMejora	R ² test (media ± desv.)	RECM test (media ± desv.)	Tiempo (min)
2	15	0,9755 ± 0,0079	0,2174 ± 0,0396	1,34
2	30	0,9718 ± 0,0058	0,2334 ± 0,0248	2,09
3	30	0,9726 ± 0,0044	0,2132 ± 0,0084	6,49
4	45	0,9747 ± 0,0047	0,2115 ± 0,0292	35,64

La configuración más simple ($\text{cap. máx.} = 2$, $\text{MaxIterSinMejora} = 15$) obtuvo resultados competitivos ($R^2 = 0,9755$, $\text{RECM} = 0,2174$) en solo 1,34 minutos. Incrementar a cuatro capas con paciencia 45 mejoró levemente la RECM (0,2115), pero aumentó el tiempo a 35,64 minutos. Las desviaciones estándar reducidas (σ de $R^2 \leq 0,0079$) confirman la estabilidad del algoritmo ante variaciones moderadas de parámetros.

Discusión

Los resultados obtenidos validan la aplicabilidad del algoritmo propuesto para la optimización de arquitecturas PMC en problemas de regresión no lineal. El método identificó consistentemente configuraciones con altos coeficientes de determinación y bajos errores cuadráticos medios en los cinco conjuntos de datos evaluados. La

formulación multiobjetivo basada en distancia Chebyshev resultó efectiva para controlar simultáneamente el rendimiento en los tres conjuntos (entrenamiento, validación y prueba), reduciendo el riesgo de sobreajuste sin términos de regularización explícitos.

Al comparar con GeNNsem [11], que emplea un conjunto de 50 redes y alcanza $R^2 = 0,9997$ en Friedman #1 y $R^2 = 0,9418$ en Boston Housing, la brecha de rendimiento es esperada dado que el promediado de conjunto reduce la varianza y mejora la generalización. No obstante, el algoritmo propuesto se aproxima a ese nivel ($R^2 = 0,9566$ en Friedman #1) con una única red optimizada, lo que supone ventajas sustanciales en simplicidad estructural, eficiencia computacional, interpretabilidad y facilidad de despliegue en entornos con recursos limitados.

Las principales limitaciones identificadas son: mayor costo computacional respecto a la búsqueda aleatoria (5-10 veces más lento); falta de significación estadística frente a la búsqueda aleatoria con el número actual de conjuntos de prueba, y frente a la optimización bayesiana en la métrica de error (ECM), pese a mostrar significación en R^2 ; dependencia de los valores objetivo de R^2 derivados del ANOVA lineal, que puede constituir una referencia débil en problemas altamente no lineales; y escalabilidad no explorada en problemas de alta dimensionalidad o con millones de muestras.

Conclusiones

El algoritmo propuesto para la optimización de hiperparámetros en redes PMC aplicadas a regresión continua no lineal presenta los principales resultados:

1. El método alcanzó valores de R^2 entre 0,8899 (Boston Housing) y 0,9983 (Eficiencia Energética Y1), superando a la optimización bayesiana en R^2 en 6 de 7 conjuntos con diferencia estadísticamente significativa ($p = 0,047$).

2. El análisis de sobreajuste mostró que los tres métodos mantienen valores moderados de ΔR^2 (0,0007–0,0590) en todos los conjuntos, sin evidencia de sobreajuste sistemático en ninguno de ellos.
3. La baja variabilidad entre ejecuciones independientes (σ de $R^2 = 0,0003$ en Eficiencia Energética Y1; $\sigma = 0,0067$ en Friedman #1) confirma la reproducibilidad del método.
4. Una red única optimizada con este método puede aproximarse al nivel de rendimiento de conjuntos de 50 redes (GeNNsem), manteniendo la simplicidad estructural.
5. El análisis de sensibilidad confirmó la robustez ante variaciones moderadas de parámetros, con configuraciones simplificadas que ofrecen una relación favorable entre precisión y costo computacional.

Como líneas de trabajo futuro, se proponen: implementación de estrategias de paralelización para reducir el tiempo de cómputo; validación en conjuntos de referencia adicionales para robustecer la evidencia estadística frente a la búsqueda aleatoria; integración de selección automática de variables dentro del esquema de códigos variables; y extensión a arquitecturas más profundas (redes convolucionales).

Referencias

- [1] BENARDOS, P. G.; VOSNIAKOS, G. C., «Optimizing feedforward artificial neural network architecture», *Engineering Applications of Artificial Intelligence* [en línea], 2007, vol. 20, no. 3, p. 365–382. Disponible en: <doi:10.1016/j.engappai.2006.06.005>.
- [2] RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J., «Learning internal representations by error propagation», en *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Cambridge, MIT Press, 1986, vol. 1, p. 318–362.

- [3] RAIAN, M. A. K.; et al., «A systematic review of hyperparameter optimization techniques in convolutional neural networks», Decision Analytics Journal [en línea], 2024, vol. 11, art. 100470. Disponible en: <doi:10.1016/j.dajour.2024.100470>.
- [4] EL-HASSANI, F. Z.; et al., «A new optimization model for MLP hyperparameter tuning: modeling and resolution by real-coded genetic algorithm», Neural Processing Letters [en línea], 2024, vol. 56. Disponible en: <doi:10.1007/s11063-024-11578-0>.
- [5] KAUR, B. P.; et al., «A genetic algorithm aided hyper parameter optimization based ensemble model for respiratory disease prediction with explainable AI», PLoS One [en línea], 2024, vol. 19. Disponible en: <doi:10.1371/journal.pone.0308015>.
- [6] NGUYEN-TRONG, K.; NGUYEN, T. T. T., «Optimization of multi-layer perceptron deep neural networks using genetic algorithms for hand gesture recognition», Journal of Computer Science [en línea], 2022, vol. 18, p. 57–66. Disponible en: <doi:10.3844/jcssp.2022.57.66>.
- [7] LI, D.; et al., «Landslide susceptibility prediction using particle-swarm-optimized multilayer perceptron», Applied Sciences [en línea], 2019, vol. 9, no. 18. Disponible en: <doi:10.3390/app9183664>.
- [8] ALI, A.; et al., «An optimized multilayer perceptron-based network intrusion detection using gray wolf optimization», Computers & Electrical Engineering [en línea], 2024, vol. 120, art. 109838. Disponible en: <doi:10.1016/j.compeleceng.2024.109838>.
- [9] TIEP, N. H.; et al., «A new hyperparameter tuning framework for regression tasks in deep neural network: combined-sampling algorithm to search the optimized hyperparameters», Mathematics [en línea], 2024, vol. 12, no. 24. Disponible en: <doi:10.3390/math12243892>.
- [10] MARTÍNEZ-COMESAÑA, M.; et al., «Use of optimised MLP neural networks for spatiotemporal estimation of indoor environmental conditions of existing

- buildings», *Building and Environment* [en línea], 2021, vol. 205, art. 108243. Disponible en: <doi:10.1016/j.buildenv.2021.108243>.
- [11] KRSTIC, L.; et al., «Evolutionary approach for composing a thoroughly optimized ensemble of regression neural networks», *Egyptian Informatics Journal* [en línea], 2024, vol. 28, art. 100581. Disponible en: <doi:10.1016/j.eij.2024.100581>.
- [12] ASGHER, U.; et al., «Novel metaheuristic approach: integration of variables method (IVM) and human-machine interaction for subjective evaluation», en *International Conference on Applied Human Factors and Ergonomics (AHFE 2019), Advances in Intelligent Systems and Computing*, vol. 965, Cham, Springer, 2019, p. 467-478. Disponible en: <doi:10.1007/978-3-030-51041-1_50>.
- [13] ARZOLA-RUIZ, J.; et al., «Using adaptive integration of variables algorithm for analysis and optimization of 2D irregular nesting problem», en *AHFE 2019, Advances in Intelligent Systems and Computing*, vol. 965, Cham, Springer, 2019, p. 479-489. Disponible en: <doi:10.1007/978-3-030-20473-0_47>.
- [14] VALDES, O. M.; ARZOLA-RUIZ, J., «Exploration of variable codes algorithm for linings materials and its thickness selection of steel casting ladles», *IEEE Latin America Transactions* [en línea], 2017, vol. 15, p. 1528-1535. Disponible en: <doi:10.1109/TLA.2017.7994802>.
- [15] VITO, S., *Air Quality* [en línea], UCI Machine Learning Repository, 2008 [consulta: 2026-03-18]. Disponible en: <https://archive.ics.uci.edu/dataset/360/air+quality>.
- [16] YEH, I.-C., *Concrete Compressive Strength* [en línea], UCI Machine Learning Repository, 1998 [consulta: 2026-03-18]. Disponible en: <https://archive.ics.uci.edu/dataset/165/concrete+compressive+strength>.
- [17] TSANAS, A.; XIFARA, A., *Energy Efficiency* [en línea], UCI Machine Learning Repository, 2012 [consulta: 2026-03-18]. Disponible en: <https://archive.ics.uci.edu/dataset/242/energy+efficiency>.

- [18] HARRISON, D.; RUBINFELD, D. L., «Hedonic housing prices and the demand for clean air», *Journal of Environmental Economics and Management* [en línea], 1978, vol. 5, p. 81-102. Disponible en: <doi:10.1016/0095-0696(78)90006-2>.
- [19] FRIEDMAN, J. H., «Multivariate adaptive regression splines», *The Annals of Statistics* [en línea], 1991, vol. 19, no. 1, p. 1-67. Disponible en: <doi:10.1214/aos/1176347963>.
- [20] GLOT, X.; BENGIO, Y., «Understanding the difficulty of training deep feedforward neural networks», en *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, JMLR Workshop and Conference Proceedings, vol. 9, 2010, p. 249-256.
- [21] ASGHER, U.; et al., «Multi-level optimization of reactive power compensation in industrial nets with heuristic modelling techniques», en *AHFE 2020, Advances in Intelligent Systems and Computing*, vol. 1213, Cham, Springer, 2020, p. 429-438. Disponible en: <doi:10.1007/978-3-030-51041-1_57>.
- [22] HECHAVARRÍA HERNÁNDEZ, J. R.; et al., «Novel multi-objective optimization algorithm incorporating decisions factors in design modeling of hydraulic nets», en *AHFE 2018, Advances in Intelligent Systems and Computing*, vol. 781, Cham, Springer, 2018, p. 690-696. Disponible en: <doi:10.1007/978-3-319-94709-9_67>.
- [23] ALHARBI, A. H.; et al., «Forecasting of energy efficiency in buildings using multilayer perceptron regressor with waterwheel plant algorithm hyperparameter», *Frontiers in Energy Research* [en línea], 2024, vol. 12, art. 1393794. Disponible en: <doi:10.3389/fenrg.2024.1393794>.
- [24] ZHANG, Y.; et al., «Predicting the compressive strength of high-performance concrete using an interpretable machine learning model», *Scientific Reports* [en línea], 2024, vol. 14. Disponible en: <doi:10.1038/s41598-024-79502-z>.
- [25] YURSHIN, V. G.; STANOVOV, V. V., «Artificial neural network architecture tuning algorithm», en *EpCT HMMOCS 2022 International Workshop on Hybrid Methods of*

Modeling and Optimization in Complex Systems [en línea], 2023. Disponible en: <doi:10.15405/epct.23021.29>.

[26] WALPOLE, R. E.; MYERS, R. H.; MYERS, S. L., Probabilidad y estadística para ingenieros, México, Pearson Educación, 1999, ISBN 978-9701702642.

[27] KTARI, A.; ELMANSORI, M., «Bridging FEM and artificial neural network in gating system design for smart 3D sand casting», Procedia Manufacturing [en línea], 2020, vol. 52, p. 795-800. Disponible en: <doi:10.1016/j.promfg.2020.10.111>.

Conflicto de intereses

Los autores declaran que no existen conflictos de intereses.

Contribución de cada autor:

Javier Carballo Pérez: Conceptualización de la implementación computacional del algoritmo; desarrollo del software; diseño y ejecución de los experimentos; validación de los resultados; análisis e interpretación de los datos; elaboración de figuras y tablas; redacción del borrador original y revisión del manuscrito.

José Arzola Ruiz: Concepción y desarrollo del método heurístico empleado en la investigación; supervisión científica; formulación metodológica; interpretación de los resultados; revisión crítica del contenido intelectual del manuscrito y aprobación de la versión final.

Se declara que ambos autores participaron en la discusión de los resultados, la revisión y edición del manuscrito, y aprobaron la versión final para su publicación.

Sobre los autores:

Javier Carballo Pérez: Ingeniero Industrial por la Universidad Tecnológica de La Habana «José Antonio Echeverría» (CUJAE), graduado con título de oro, premio al mérito científico. Profesor de la Facultad de Ingeniería Industrial de la CUJAE e investigador en optimización, inteligencia artificial y análisis de datos, con énfasis en la optimización de arquitecturas de redes neuronales artificiales mediante métodos

metaheurísticos. Ha participado en proyectos de optimización de procesos industriales y es autor de publicaciones y ponencias nacionales e internacionales en las áreas de ingeniería industrial e inteligencia artificial. Correo electrónico: carballoperezjavier0@gmail.com.

José Arzola Ruiz: Ingeniero Metalúrgico especializado en Automatización desde 1971. Doctor en Ciencias en Cibernética Técnica desde 1982, en la Universidad Politécnica de San Petersburgo, Rusia. Investiga en el área del análisis y síntesis de sistemas de ingeniería, métodos asociados y sus aplicaciones. Tiene 2 libros y alrededor de 100 artículos publicados, dedicados a su línea de investigación. Profesor de la facultad de Mecánica de la CUJAE. Correo electrónico: josearzolaruiz1945@gmail.com.