



C-18

## OPTIMIZACIÓN DE LA PREDICCIÓN DE DEMANDA DE AGUA MEDIANTE ALGORITMOS NEURO-GENÉTICOS PARA UN CONJUNTO DE DATOS REDUCIDO

González Perea, R.<sup>1</sup>, Camacho Poyato, E.<sup>2</sup>, Montesinos, P.<sup>3</sup>, Rodríguez Díaz, J.A.<sup>4</sup>

<sup>1</sup> Doctorando FPU. Departamento de Agronomía. Universidad de Córdoba. Campus Rabanales, Edif. Da Vinci, 14071. Córdoba. [g72goper@uco.es](mailto:g72goper@uco.es)

<sup>2</sup> Catedrático. Departamento de Agronomía. Universidad de Córdoba. Campus Rabanales, Edif. Da Vinci, 14071. Córdoba. [ecamacho@uco.es](mailto:ecamacho@uco.es)

<sup>3</sup> Profesora Titular. Departamento de Agronomía. Universidad de Córdoba. Campus Rabanales, Edif. Da Vinci, 14071. Córdoba. [pmontesinos@uco.es](mailto:pmontesinos@uco.es)

<sup>4</sup> Contratado Doctor. Departamento de Agronomía. Universidad de Córdoba. Campus Rabanales, Edif. Da Vinci, 14071. Córdoba. [jarodriguez@uco.es](mailto:jarodriguez@uco.es)

### Resumen

La predicción de la demanda de agua es uno de los factores principales en el diseño y gestión de sistemas de abastecimiento y distribución de agua. Recientemente, avanzadas técnicas en inteligencia computacional como las Redes Neuronales Artificiales (RNAs) han sido aplicadas para la predicción de series temporales con importantes resultados. En este trabajo se ha desarrollado una metodología híbrida que combina RNAs y Algoritmos Genéticos multiobjetivo para la predicción a corto plazo de la demanda de agua en una Comunidad de Regantes cuando la disponibilidad de datos es escasa. El modelo fue desarrollado utilizando datos de series temporales del Sector VII de la Zona Regable Bembézar M.D. Tras el proceso de optimización con un algoritmo genético multiobjetivo se obtuvo una RNA de tipo perceptrón multicapa entrenada mediante el algoritmo *Regularización Bayesiana* con 24 neuronas en la primera capa oculta y 21 en la segunda. El modelo desarrollado fue capaz de explicar el 95 % de la varianza total de los datos observados con un Error Estándar de Predicción del 9.38 % (periodo de test).

### 1.- Introducción

El aumento de la población mundial unido a la creciente demanda de alimentos hace que sea necesario incrementar la producción agrícola. Para asegurar este aumento de la producción agrícola, la agricultura de regadío desempeña un papel fundamental.

La agricultura de regadío es el mayor usuario de agua en todo el mundo, presentando el 70% del total de agua dulce. En países como España, el uso del agua en la agricultura representa el 80% de la demanda total de agua, correspondiendo la mayor parte a la agricultura de regadío. Así, con el objetivo de aumentar la eficiencia en el uso del agua, muchas zonas de regadío han sido sometidas a procesos de modernización. En consecuencia, la mayoría de los nuevos sistemas de riego (goteo y aspersión) operan bajo demanda respetando el tiempo, la duración y la frecuencia de riego que lo agricultores deciden. Debido a este aumento de la flexibilidad en la gestión del riego de cada agricultor, se produce una gran variabilidad temporal de la demanda de agua en toda la red de riego. Por esta razón, la predicción de la demanda de agua se ha convertido en uno de los principales problemas en el diseño, gestión y modernización de los sistemas de abastecimiento y distribución de agua.

En los últimos años, ha sido posible una importante evolución en el campo de la predicción gracias a los avances en los modelos de sistemas no lineales conocidos como

Redes Neuronales Artificiales (RNAs). Las características de las RNAs están inspiradas en el sistema nervioso biológico que les da una serie de ventajas como la capacidad de aprendizaje adaptativo, son auto-organizadas, pueden funcionar en paralelo en tiempo real y proporcionan tolerancia a errores por codificación de la información redundante.

Por otra parte, cuando se consideran varios objetivos simultáneamente en la gestión de redes de agua, el proceso de toma de decisiones se mejora de manera significativa con una amplia gama de alternativas por lo general debido a que se identifica y se consigue un modelo más realista del problema. Sin embargo, en este tipo de problemas es difícil encontrar una solución utilizando técnicas de optimización tradicionales. Por esta razón, los enfoques heurísticos son adecuados para resolver este tipo de problemas (Baños et al. 2010). Estas herramientas se basan por lo general en los algoritmos genéticos multiobjetivo (Savic 2002; Farmani et al. 2005), que implican más de una función objetivo a optimizar simultáneamente obteniendo la mejor combinación de los valores de las variables de decisión.

Las RNAs y los algoritmos genéticos son tecnologías de “soft-computing” que pueden ser muy eficaces cuando son usados de forma individual. Sin embargo, cuando se combinan juntos, las fortalezas individuales de cada enfoque pueden ser explotadas de manera sinérgica para la construcción de sistemas de gran alcance, híbridos e inteligentes (See y Openshaw 2000).

Uno de los principales problemas al desarrollar un modelo predictivo es la escasez de información con la que se cuenta que hace que el modelo desarrollado sea poco robusto y/o representativo. Así, el objetivo de este trabajo fue desarrollar un modelo predictivo mediante algoritmos neuro-genéticos multiobjetivo cuando la información que se posee es reducida. El modelo desarrollado se aplicó en el Sector VII de la Zona Regable Bembézar M.D.

## **2.- Materiales y métodos**

### 2.1. Zona de estudio y fuente de datos

La Comunidad de Regantes Bembézar Margen Derecha (BMD) se extiende por 4 municipios que son: Hornachuelos, Palma del Río, Peñaflor y Lora del Río. Los cultivos principales son: Maíz, Algodón, cítricos y frutales. El sector VII, objeto de este trabajo, está situado en el Término municipal de Lora del Río. Después de la modernización, BMD posee un sistema de distribución de agua a presión que suministra a 11 sectores de riego independientes, organizados a la demanda. Dentro de la red, el Sector VII suministra agua de riego a 935 ha aproximadamente.

El Sector VII cuenta con un sistema de telemedida que almacena datos de caudal y altura de presión en la estación de bombeo cada minuto. Los datos de demanda de agua utilizados para la calibración y validación del modelo desarrollado fueron obtenidos a partir del promedio de los valores registrados por el sistema durante las campañas de riego 2010, 2012 y 2013.

### 2.2. Redes Neuronales Artificiales

Una Red Neuronal Artificial (RNA) es un modelo heurístico utilizado para determinar la relación no lineal entre input y output. El modelo de RNA más ampliamente utilizado es el Perceptrón Multicapa (Multilayer Perceptron, MLP) (Rumelhart et al. 1986). Una estructura típica de un MLP está formada por cuatro capas (capa de entrada, primera y segunda capa oculta y capa de salida). Cada capa está formada por un número de neuronas. Así, la anotación típica de un MLP es (g, n, m, s) siendo g, n, m y s los nodos o neuronas en la capa de entrada, la primera y segunda capa oculta y la capa de salida, respectivamente. Las neuronas de cada capa están conectadas a las neuronas de la siguiente capa mediante los pesos sinápticos (w). Todas las conexiones se alimentan hacia delante, es decir, que sólo permiten la transferencia de información desde una capa anterior a una posterior. El número

de neuronas de las capas de entrada y salida son las mismas que el número de variables de entrada y salida, respectivamente. El número de neuronas de la primera y segunda capa oculta fueron dos variables de decisión del algoritmo genético utilizado para el proceso de optimización del modelo predictivo desarrollado.

Las neuronas de cada capa son activadas según una función de activación. Esta función representa simultáneamente la salida de la neurona y su estado de activación. En este trabajo, la función de activación asociada a las neuronas de cada capa fueron tres variables de decisión (capa de entrada, capas oculta y capa de salida). En la tabla 1 se muestran las distintas funciones de activación utilizadas.

**Tabla 1.** Variables de decisión de cada cromosoma.

Número	Nombre de la variable de decisión	Rango de valores de cada variable de decisión
1	Neuronas 1ªcapa oculta	Número entero entre 1 y 50.
2	Neuronas 2ªcapa oculta	Número entero entre 0 y 50.
3	Función de Transferencia de la capa de entrada	Número entero entre 1 y 3: (1) <i>Función de Transferencia Log-Sigmoid</i> (LogST); (2) <i>Función de Transferencia Tan-Sigmoid</i> ; (3) <i>Función de Transferencia Linear</i> (LinT).
4	Función de Transferencia de las capas ocultas	Número entero entre 1 y 3: (1) <i>Función de Transferencia Log-Sigmoid</i> (LogST); (2) <i>Función de Transferencia Tan-Sigmoid</i> ; (3) <i>Función de Transferencia Linear</i> (LinT).
5	Función de Transferencia de la capa de salida	Número entero entre 1 y 3: (1) <i>Función de Transferencia Log-Sigmoid</i> (LogST); (2) <i>Función de Transferencia Tan-Sigmoid</i> ; (3) <i>Función de Transferencia Linear</i> (LinT).

### 2.2.1. Entrenamiento y generalización de una RNA.

Para determinar el valor del conjunto de pesos ( $w$ ) que constituyen la red neuronal, se lleva a cabo un proceso repetitivo-correctivo que es conocido como entrenamiento o aprendizaje. El objetivo de la fase de entrenamiento es definir las interconexiones entre neuronas ajustando los pesos sinápticos a través de patrones de entrenamiento (conjunto de relaciones de entradas y salidas conocidas). Estos patrones de entrenamiento son presentados a la RNA de forma ordenada, ajustando los pesos sinápticos hasta que se alcanzan las salidas deseadas. Para determinar la actualización de los pesos sinápticos en cada una de las iteraciones se utilizan diferentes algoritmos denominados “algoritmos de aprendizaje por corrección de errores”. Los algoritmos de este tipo ajustan los pesos en proporción a la diferencia existente entre la salida actual de la red y la salida deseada, con el objetivo de minimizar el error actual de la red.

Durante la fase de entrenamiento de una red neuronal, el error de predicción del conjunto de datos en esta fase puede llegar a ser muy pequeño pero cuando una serie de nuevos datos son presentados a la red neuronal, este error crece considerablemente. Este fenómeno es conocido como sobreajuste. La red no ha aprendido a generalizar para nuevas situaciones sino que ha memorizado los datos en la fase de entrenamiento. Para mejorar la generalización de la red se suele utilizar un método conocido como “*parada temprana*” que consiste en dividir todos los datos disponibles en tres subconjuntos: conjunto de entrenamiento, conjunto de validación y conjunto de test. El primer conjunto es utilizado para calcular el gradiente de la función de error y actualizar los pesos sinápticos de la red. El conjunto de validación se utiliza para evitar el sobreajuste y el conjunto de test es utilizado para comprobar la robustez y representatividad del modelo predictivo así como para comparar entre diferentes modelos.

Cuando el conjunto de datos disponibles para entrenar la RNA es reducido, los

conjuntos de entrenamiento, validación y test son muy reducidos y el modelo predictivo obtenido resulta poco representativo y/o con un error de predicción alto. Por otro lado, existe un algoritmo de entrenamiento conocido como Regularización Bayesiana (RB) (Mackay 1992) el cual asume que los pesos sinápticos de la RNA son variables aleatorias con distribuciones normales. Cuando el conjunto de datos disponibles es reducido, RB da mejores rendimientos de generalización que la “*parada temprana*” debido a que este algoritmo no requiere dividir el conjunto de datos en sus correspondientes conjuntos de entrenamiento y validación. Por tanto, en este trabajo se ha utilizado BR como función de entrenamiento de la RNA.

La función de rendimiento típica utilizada en redes neuronales alimentadas hacia delante (feedforward) es el error cuadrático medio (mean square error, *mse*) (Ecuación 1).

$$\text{---} \quad (1)$$

donde  $OB$  es el número total de datos utilizados para el entrenamiento de la red neuronal;  $Q_t$  es la demanda de agua observada en el dato número  $t$  ( $L s^{-1}$ ) y  $\hat{Q}_t$  es la demanda de agua estimada para el mismo dato  $t$  ( $L s^{-1}$ ).

El modelo de predicción desarrollado se implementó en Matlab utilizando la herramienta para Redes Neuronales Artificiales (Artificial Neural Network toolbox).

### 2.3. Optimización del modelo RNA mediante un algoritmo genético

#### *2.3.1. Planteamiento del problema*

El objetivo de este trabajo fue desarrollar un modelo de predicción diaria de demanda de agua en una red de riego cuando el conjunto de datos que se disponen es reducido. El problema fue formulado como un problema de optimización multiobjetivo con dos funciones objetivo.

La finalidad de la primera función objetivo ( $F1$ ) fue maximizar la suma de los coeficientes de determinación ( $R^2$ ) del conjunto de test (Ecuación 2). Este coeficiente describe la proporción de varianza total de los datos observados que puede ser explicado por el modelo.

(2)

La finalidad de la segunda función objetivo ( $F2$ ) fue minimizar el RMSE (Average Root Mean Square Error) (Ecuación 3).

$$\text{---} \quad (3)$$

donde  $OB_{\text{test}}$  es el número total de observaciones utilizadas durante el periodo de test de la red neuronal artificial.

$F2$  fue normalizada, con la finalidad de poder comparar las dos funciones objetivo ( $F1$  y  $F2$ ).

Con el objetivo de poder comparar el rendimiento de los diferentes modelos de predicción desarrollados se utilizó el error estándar de predicción (Standard Error of Prediction (SEP)) (Ventura et al. 1995). El SEP es definido según la ecuación 4.

$$\text{---} \quad (4)$$

donde  $D$  es la demanda de agua media observada del conjunto de test ( $L s^{-1}$ ).

### 2.3.2. Método de optimización. NSGA-II.

El algoritmo de optimización NSGA (Deb et al. 2002) fue implementado en MATLAB (Deb 2010) para obtener un modelo óptimo de predicción de demanda de agua en una red de riego. Se utilizó el algoritmo de optimización NSGA-II el cual fue adaptado a este problema (Figura 1). Las etapas iniciales, la población inicial y las funciones objetivo de este algoritmo fueron ligeramente modificadas para adaptarlo al caso particular de este trabajo. El resto de etapas: “cruzamiento”, “mutación” y “selección” no fueron modificadas con respecto al algoritmo original.

En la primera etapa, la población inicial de  $N$  cromosomas (diferentes configuraciones de RNAs) fue generado aleatoriamente. Cada cromosoma representa un modelo de RNA. El número de variables de decisión ( $C$ ) en el cromosoma representa un parámetro a optimizar de la RNA. A continuación,  $F1$  y  $F2$  fueron calculadas para cada cromosoma. Con el objetivo de poder comparar las dos funciones objetivo,  $F2$  fue normalizada previamente utilizando una función de distribución uniforme continua en el intervalo  $[0, 1]$ . Así, los valores de ambas funciones objetivo estaban comprendidos en ese intervalo. En las etapas restantes, cada valor de las variables de decisión de los cromosomas fueron modificados (cruzamiento y mutación) y los  $N$  cromosomas con mejores valores de las funciones objetivo fueron seleccionados. El proceso fue repetido durante varias generaciones (GEN). Finalmente, el conjunto de  $N$  cromosomas óptimas de la última generación es conocida como el Frente de Pareto.

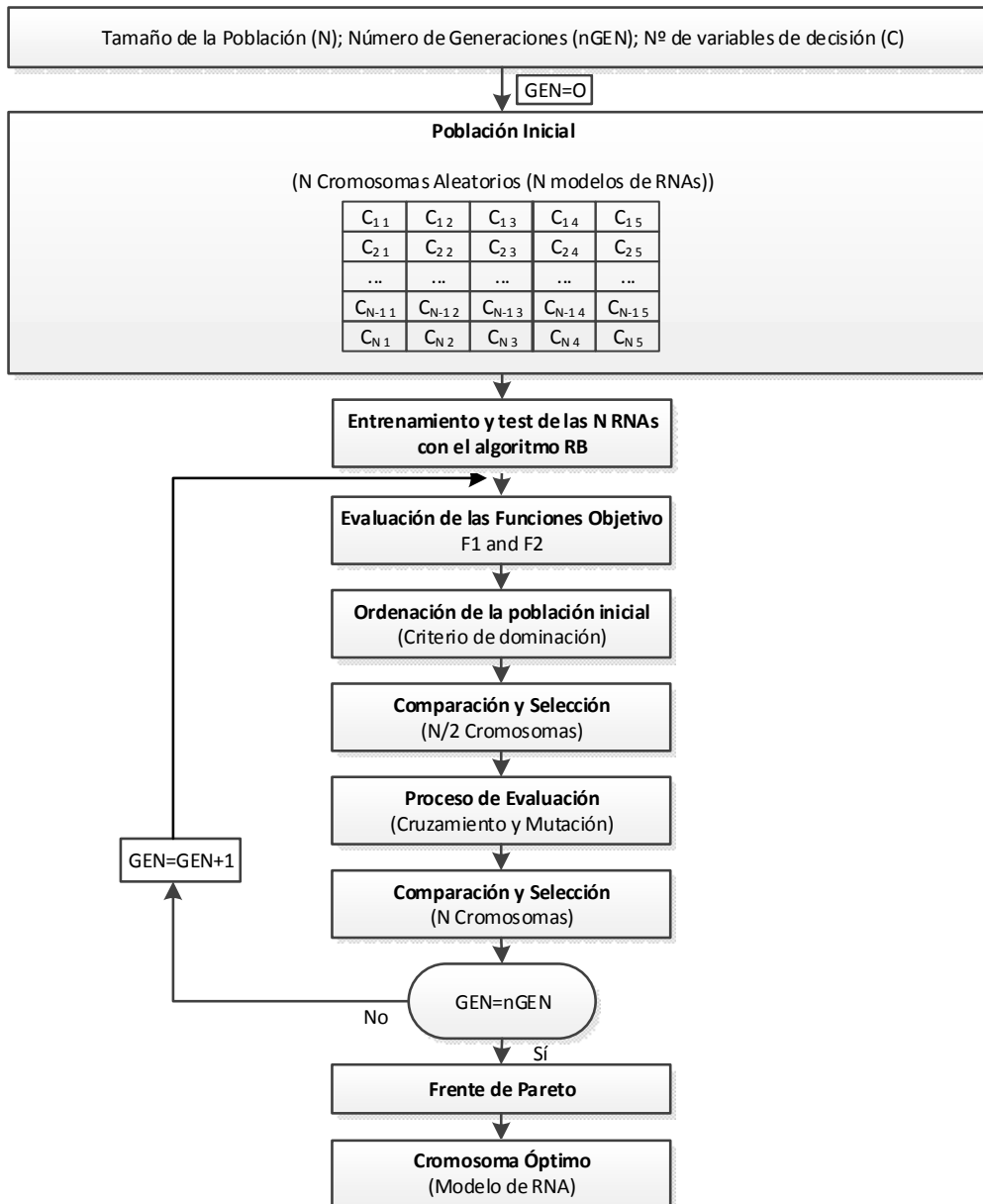


Figura 1. Proceso de optimización utilizando NSGA-II.

### 3.- Resultados y discusión

En este trabajo se consideraron 5 variables de decisión. Los valores que pueden tomar cada una de estas variables de decisión se muestran en la Tabla 1. Los conjuntos de entrenamiento y test fueron elegidos aleatoriamente del total de los datos disponibles en una proporción de 85% para el conjunto de entrenamiento y un 15 % para el conjunto de test.

Con el objetivo de implementar los modelos desarrollados en una Comunidad de Regantes y predecir la demanda de agua en tiempo real, la velocidad de entrenamiento de la red neuronal constituye un factor clave. Así, durante la implementación de las funciones objetivo se programó una restricción de tiempo del entrenamiento de las diferentes RNAs (60 segundos), seleccionando el algoritmo genético de manera indirecta aquellas configuraciones de RNAs más rápidas.

Después de realizar un análisis de correlación de 21 variables potenciales de entrada al modelo de predicción, se seleccionaron las 7 siguientes: demanda de agua del día anterior ( $L s^{-1}$ ), demanda de agua de dos días anteriores ( $L s^{-1}$ ), temperatura media del día a predecir ( $^{\circ}C$ ), radiación solar del día a predecir ( $MJ m^{-2}$ ), radiación solar del día anterior ( $MJ$

$m^{-2}$ ), evapotranspiración de referencia del día a predecir ( $mm\ día^{-1}$ ) y evapotranspiración de referencia del día anterior ( $mm\ día^{-1}$ ).

### 3.1. Evolución de las funciones objetivo en el proceso de optimización

El modelo basado en el algoritmo NSGA-II y descrito en la figura 1 fue aplicado al Sector VII de la Zona Regable Bembézar M.D. y tanto F1 como F2 fueron optimizadas. La población inicial aleatoria fue de 70 cromosomas. Cada cromosoma está compuesto por 5 genes. Cada gen representa una característica diferente de una RNA. La población inicial se desarrolló para 130 generaciones y las probabilidades de cruce y mutación fueron 90 % y 10 %, respectivamente.

En la figura 2 se muestra el Frente de Pareto que se obtuvo tras el proceso de optimización (generación 130). Esta figura muestra claramente aquellos RNAs con una única capa oculta (línea inferior de puntos) y con dos capas ocultas (línea superior de puntos). Debido a la restricción de tiempo anteriormente especificada, aquellas redes neuronales con una capa oculta, tendrán menor número de neuronas y por tanto la fase de entrenamiento es más rápida, es decir, para el mismo tiempo de entrenamiento obtienen valores de F2 más pequeños. Por el contrario, si la RNA tiene dos capas ocultas, ésta es más densa y necesita más tiempo de entrenamiento. Así, para el mismo tiempo de entrenamiento, el error cometido en la predicción (F2) es mayor con respecto a redes con una capa oculta, aunque la variabilidad explicada (F1) es mayor debido a ese aumento de neuronas en la segunda capa oculta.

Los mejores individuos para RNAs con una y dos capas ocultas están representados por los colores rojo y verde, respectivamente.

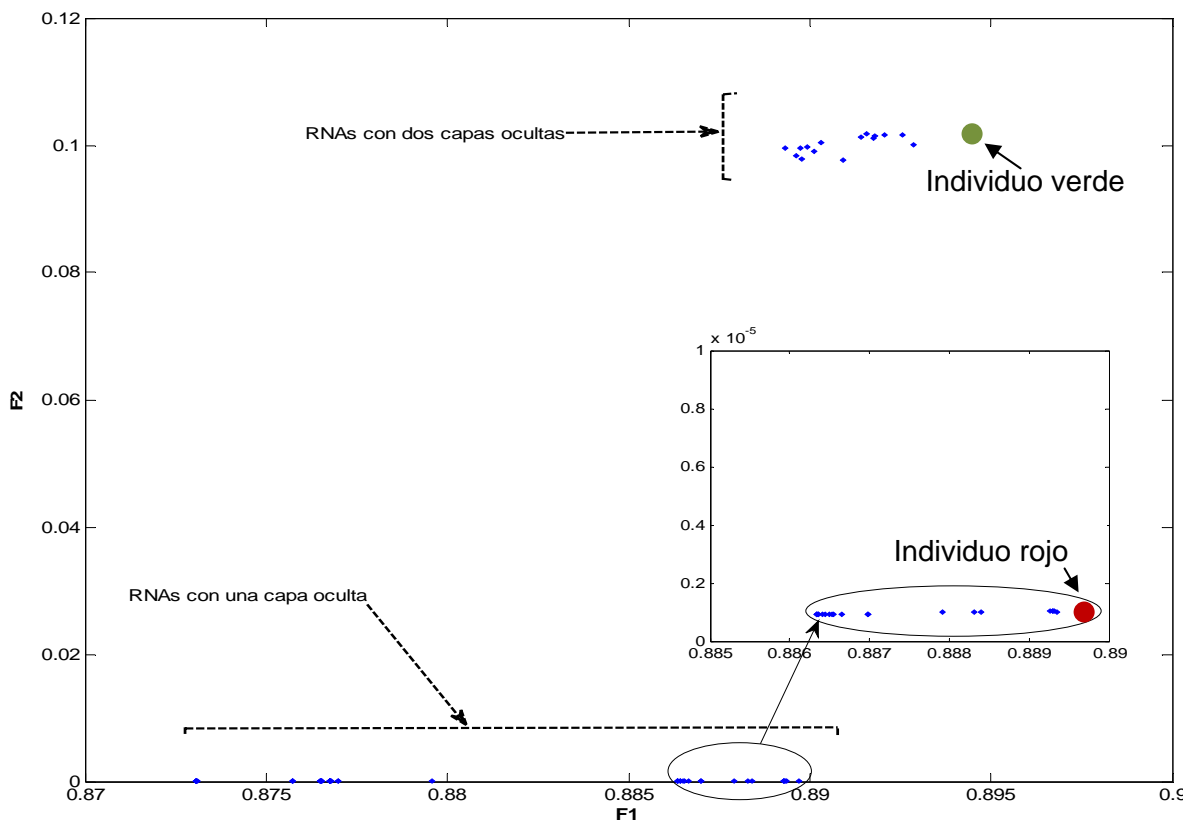


Figura 2. Frente de Pareto para la Generación 130.

### 3.2. Modelos de RNAs óptimos

Cuando el algoritmo RB es utilizado para entrenar RNAs es importante dejar que éste se ejecute hasta que sus parámetros hayan convergido. Así, los dos individuos seleccionados anteriormente fueron entrenados de nuevo con 100 repeticiones para favorecer la convergencia dichos parámetros y disminuir el efecto de la aleatoriedad inicial de los valores de los pesos sinápticos.

El mejor individuo para las RNAs con una sola capa oculta (individuo rojo) y con dos capas ocultas (individuo verde) se muestran en la Tabla 2. Los valores de  $R^2$  y RMSE durante el periodo de test fueron 0.88 y 0.95; 68.06  $L s^{-1}$  y 42.44  $L s^{-1}$ , para los individuos rojo y verde, respectivamente. La configuración de la red neuronal para los dos individuos fueron  $RNA_{rojo}(7, 30, 1)$  y  $RNA_{verde}(7, 24, 21)$ .

En la figura 2, el valor de F2 del individuo rojo fue de varios órdenes de magnitud mayor con respecto al individuo verde mientras que F1 no varió considerablemente entre ambos individuos, siendo estos valores de 0.889 y 0.895, respectivamente. Sin embargo, después de que estos dos individuos se volvieran a entrenar sin restricción de tiempo, claramente, el individuo verde que cuenta con dos capas ocultas obtiene mejores resultados que el individuo rojo. Con restricción de tiempo, aquellas redes con menor número de neuronas (una única capa oculta) obtienen mejores resultados ya que la convergencia del algoritmo de entrenamiento (RB) es más rápida cuando la red neural es más simple. Pero, cuando el algoritmo de entrenamiento RB tiene tiempo suficiente para converger bien, las redes neuronales más complejas son más precisas y representan mejor la variabilidad observada. Así, el error estándar de predicción (SEP) para el individuo verde es del 9.38 % mientras que para el individuo rojo es del 15.05 %, siendo además considerablemente mejor la representatividad de la variabilidad observada del individuo verde con respecto al individuo rojo.

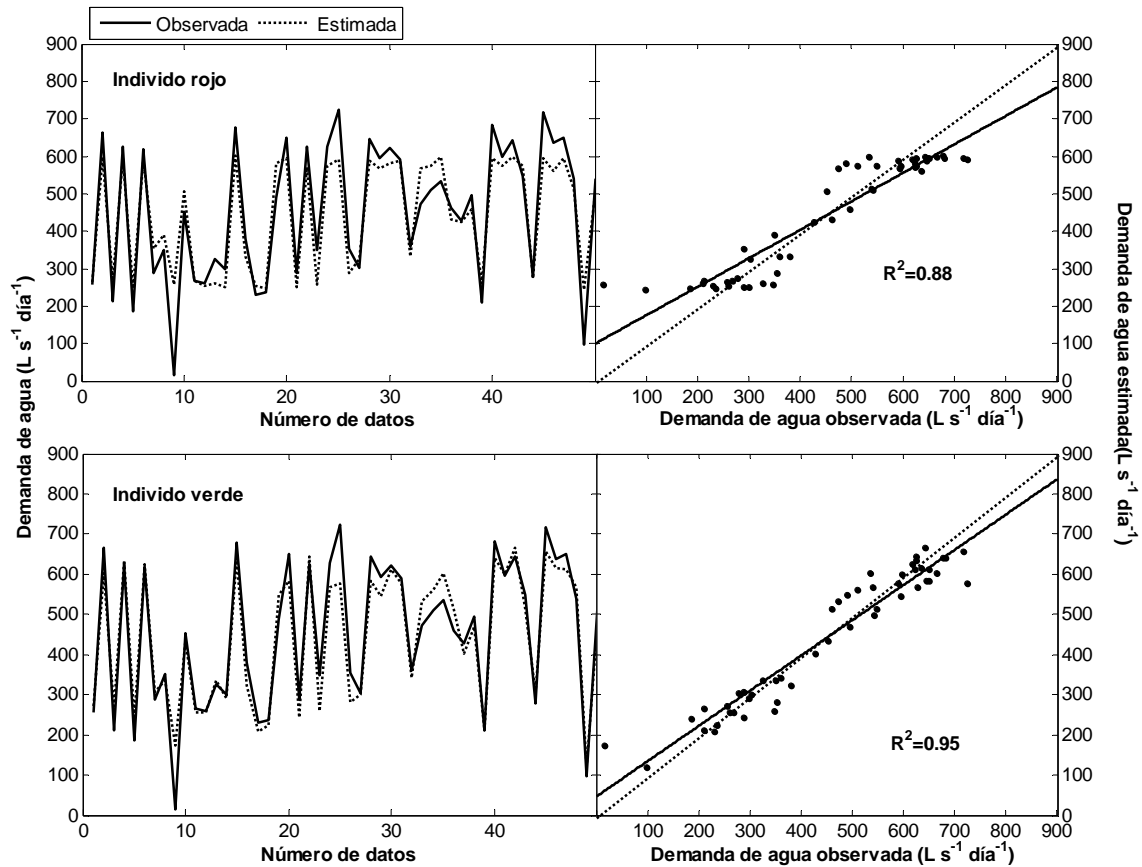
**Tabla 2.** Mejores modelos de RNA obtenidos tras el proceso de optimización.

		Individuo Rojo	Individuo Verde
	<b>Nº Neuronas de la 1ª capa oculta</b>	30	24
	<b>Nº Neuronas de la 2ª capa oculta</b>	0	21
	<b>Función de Transferencia de la capa de entrada</b>	TanST	LinT
	<b>Función de Transferencia de las capas ocultas</b>	TanST	LogST
	<b>Función de Transferencia de la capa de salida</b>	LogST	TanST
	<b>Algoritmo de entrenamiento</b>	RB	RB
<b>Periodo de test</b>	<b><math>R^2</math></b>	0.88	0.95
	<b>RMSE (<math>L s^{-1}</math>)</b>	68.06	42.44
	<b>SEP (%)</b>	15.05	9.38

En la figura 3 se muestra la predicción de los valores del conjunto de test así como el diagrama de dispersión para la  $RNA_{rojo}$  y  $RNA_{verde}$ .

Esta figura muestra como la demanda de agua predicha con el modelo verde [ $RNA_{verde}(7, 24, 21, 1)$ ] se ajusta mejor a la demanda de agua observada para todo el rango de caudales. Tanto en el modelo predictivo representado por el individuo rojo como en el representado por el individuo verde de la figura 3 se puede observar algunos picos de demanda de agua que el modelo no predice correctamente. Este hecho muestra que durante el periodo de entrenamiento no se ha producido un sobreajuste, ya que esos valores máximos pueden ser valores atípicos (outliers) o el ruido propio del sistema.





**Figura 3.** Predicción de demanda de agua para los valores de test y diagrama de dispersión entre la demanda de agua observada y estimada.

Según los valores de  $R^2$  y SEP, la RNA más adecuada para predecir la demanda de agua diaria fue la RNA<sub>verde</sub>. Los valores ( $R^2=0.95$  y  $SEP=9.38\%$ ) fueron superiores a los obtenidos por Pulido-Calvo (2009) y Gonzalez et al. (2015) que obtuvieron valores de  $R^2$  y SEP en sus modelos predictivos de 0.89 y 20.27%; 0.93 y 12.63%, respectivamente.

#### 4.- Conclusiones

El objetivo de este trabajo fue predecir la demanda diaria de agua de una zona de riego utilizando la información que nos facilitaba la estación de bombeo y una metodología híbrida que combina RNAs y algoritmos genéticos cuando el número de datos de los que se disponen es reducido. En este modelo, se utilizó un algoritmo genético multiobjetivo para encontrar la configuración óptima de una RNA que explique la máxima varianza de demanda de agua observada con el mínimo error de estimación. La función de entrenamiento utilizada fue RB la cual asume que los pesos sinápticos de la RNA son variables aleatorias con distribuciones normales, no siendo necesario la división de los datos en los subconjuntos de entrenamiento y validación. Así, cuando el número de datos es reducido, esta función de entrenamiento presenta mejor generalización que la “parada temprana”.

El modelo desarrollado para predecir la demanda de agua diaria se aplicó al Sector VII de la Zona Regable Bembézar M. D., Sur de España. Los resultados indicaron el modelo desarrollado en este trabajo tuvo mejores estadísticos y medidas de error que los modelos desarrollados por otros autores.

Este modelo ha demostrado ser una herramienta útil que, con necesidades no muy grandes de datos y de tiempo, puede ofrecer importantes ventajas para el desarrollo de políticas y la gestión del agua de riego. Además, dicha información puede ser de utilidad para la programación del funcionamiento de la estación de bombeo y la contratación de la

tarifa eléctrica que reduzca al mínimo los términos de potencia.

## 5.- Agradecimientos

Este trabajo es parte del proyecto TEMAER (AGL2014-59747-C2-2-R), financiado por el Ministerio de Economía y Competitividad de España.

Este trabajo fue apoyado por el contrato predoctoral FPU (Formación del Profesorado Universitario) del Ministerio de Educación, Cultura y Deporte de España cuyo beneficiario es Rafael González Perea.

## 6.- Bibliografía

- Baños R, Gil C, Reca J, Montoya F.G., 2010. A memetic algorithm applied to the design of water distribution networks. *Appl Soft Comput* 10:261–266
- Deb K, Pratap A, Agarwal S, Meyarivan T, 2002. A fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002; Vol. 6, Nº 2
- Farmani R, Savic D.A., and Walters G.A., 2005. Evolutionary multi-objective optimization in water distribution network design. *Eng. Optimiz* 37 (2), 167-183
- González Perea R, Camacho Poyato E, Montesinos P, Rodríguez Díaz J.A., 2015. Irrigation Demand Forecasting Using Artificial Neuro-Genetic Networks. *Water Resources Management* 29, 5551-5567
- Mackay D, 1992. Bayesian Interpolation. *Neural Computation* 4(3), 415–447
- Pulido-Calvo I, Gutiérrez-Estrada J.C., 2009. Improved irrigation water demand forecasting using a soft-computing hybrid model. *Biosystems Engineering* 102(1), 202-218
- Rumelhart D.E., Hinton G.E., Williams R.J., 1986. Learning representations by backpropagation errors. *Nature* 323(9), 533–536
- Savic D, 2002. Single-objective vs. multiobjective optimization for integrated decision support. Proc., First Biennial Meeting of the Int. Environmental Modeling and Software Society (IEMSs), Lugano, Switzerland, 7-12
- See L, Openshaw S, 2000. A hybrid multi-model approach to river level forecasting. *Hydrological Sciences Journal* 45(4), 523–536
- Ventura S, Silva M, Pérez-Bendito D, Hervás C, 1995. Artificial neural networks for estimation of kinetic analytical parameters. *Analytical Chemistry* 67(9), 1521–1525