Publicaciones en Ciencias y Tecnología Vol. 10 Nº 1, Ene-Jun (2016) 12–24

Artículo de investigación original

Optimización del diseño de parámetros: Método Forest-Genetic univariante

Optimizing parameter design: The univariate Forest-Genetic method

Adriana Villa-Murillo[a], Andrés Carrión García[b], Antonio Sozzi Rodriguez[a]

[a] *Universidad Centroccidental Lisandro Alvarado. Decanato Agronomía. Departamento Ingeniería Agrícola. Cabudare, Venezuela*
[b] *Universidad Politécnica de Valencia. Departamento de Estadística e Investigación Operativa Aplicadas y Calidad. Valencia. España.*

Recibido: 05-10-2015 Aceptado: 06-02-2017

Resumen

El Dr Genichi Taguchi desarrolló en los años 80 una metodología para la mejora del diseño de parámetros de productos y procesos, conocida como metodología Taguchi. Diversas propuestas han surgido en las que se mezclan técnicas de inteligencia artificial. Proponemos la creación de un híbrido entre Random Forest (RF) y los Algoritmos Genéticos (GA) en tres fases; normalización, modelización y optimización. La primera fase corresponde a la preparación previa del conjunto de datos mediante funciones de normalización. En la modelización se determina la función objetivo utilizando estrategias basadas en RF para predecir el valor de la respuesta en un conjunto de parámetros dado. Finalmente, en la fase de optimización se obtiene la combinación óptima de los niveles de los parámetros mediante la integración de propiedades dadas por nuestro esquema de modelización en el establecimiento del correspondiente GA. Se comparan los resultados de forma numérica con aportes recientemente encontrados en la literatura. Nuestra propuesta metodológica se concentra en las variables de mayor importancia producto del proceso de modelización con RF, lo que permite desarrollar y dirigir de manera más eficiente las nuevas generaciones en la fase de optimización y en consecuencia, alcanzar significativas mejoras en cuanto al objetivo de calidad considerado.

Palabras clave: Taguchi, árboles de regresión y clasificación, random Forest, algoritmos genéticos, redes neuronales artificiales.
Código UNESCO: 1209.05

Abstract

In the 80's, Dr Genichi Taguchi developed a methodology for processes and product parameters design improvement known as the Taguchi methodology. Different proposals have emerged involving artificial intelligence techniques. Our proposal consists of a hybrid methodology that combines Random Forest (RF) and Genetic Algorithms (GA) in three phases: normalization, modeling and optimization. The first phase corresponds to the previous preparation of the data set by using normalization functions. In the modeling, the objective function is determined using strategies based on RF to predict the value of the response in a given set of parameters. Finally, in the optimization phase, the optimal combination of the parameter levels is obtained by integrating properties given by our modeling scheme into the corresponding GA. The results are compared numerically with the contributions recently found in the literature. Our methodological proposal focuses on the most important variables resulting from the RF modeling process, which allows to develop and direct more efficiently the new generations in the optimization phase, and consequently, achieve significant improvements in the quality objective considered.

Keywords: Taguchi, classification and regression trees, random forest, genetic algorithm, artificial Neural Networks.
UNESCO Code: 1209.05

Impresa: ISSN 1856-8890. Dep. Legal pp200702LA2730. Digital: ISSN 24779660 Dep. Legal ppi201402LA4590.

Email addresses: avilla@ucla.edu.ve (Adriana Villa-Murillo), acarrion@eio.upv.es (Andrés Carrión García), asozzi@ucla.edu.ve (Antonio Sozzi Rodriguez)

1. Introducción

La ingeniería de calidad surge como una disciplina para detectar y prevenir problemas de calidad desde las etapas tempranas del desarrollo y diseño del producto, hasta los problemas asociados con sus funciones y costes derivados de la fabricación y puesta en el mercado. Taguchi propuso, enfatizar el esfuerzo en la calidad del producto desde el diseño del mismo, desarrollando una aproximación al diseño de experimentos. Ésto se conoce como Diseño Robusto de Parámetros, donde el objetivo es reducir la variación de los productos y procesos para seleccionar el conjunto de los factores de control que proporcione el mejor rendimiento y menor sensibilidad a los factores de ruido.

Las ideas de Taguchi constituyen grandes contribuciones a la ingeniería de calidad, sin embargo, desde sus inicios han dado lugar a grandes discusiones para su aplicación. Box y Meyer [1] muestran que en algunos casos es posible identificar factores que afecten tanto la varianza como la media empleando diseños factoriales fraccionados en lugar de los arreglos ortogonales (ortogonal arrays) sugeridos por Taguchi, poco después Ryan [2] publica en su obra una discusión detallada de las limitaciones de los procedimientos empleados y recomienda el uso de algún tipo de diseño discriminante para eliminar factores no significantes e incorporar procedimientos de programación no lineal para la fase de optimización.

Más recientemente, Maghsoodlo *et al* [3] en su obra muestra que las técnicas Taguchi no son fáciles de aplicar en la vida real, considerando que las relaciones señal-ruido (S/N, por sus siglas en inglés) carecen de rigurosidad estadística para identificar el mejor nivel del factor que minimice las pérdidas de calidad. Tsui [4] y Montgomery [5] afirman que dividir en dos arreglos ortogonales trae como consecuencia el aumento en el número de repeticiones de los experimentos, lo que se traduce en costes innecesarios. Otros trabajos como los de Miller y Wu[6], Su y Chang [7], Zang *et al* [8], entre otros, expresan que en la práctica, el enfoque de Taguchi se limita a elegir el mejor nivel del factor entre un grupo especificado previamente, así como su limitación a parámetros de tipo discreto y el estudio de experimentos para disminuir el rango de los niveles de los factores de control, lo que trae un excesivo número de ensayos y falta de rigor en las conclusiones.

Diversas alternativas han surgido con el fin de mejorar el diseño de parámetros de Taguchi. En casos de una sola característica de calidad Chiu *et al* [9], Tay y Butler [10] combinan las ideas de Taguchi con el empleo de ANN pero dicha combinación resulta ineficiente en el momento de obtener la combinación óptima de parámetros [7]. Su y Chang [7] proponen un enfoque que combina las ANN con SA y Chang [11] nuevamente emplea ANN pero ahora en combinación con GA para el proceso de optimización.

Nuestra propuesta plantea una alternativa a dichas metodologías en tres fases. La primera fase consiste en la normalización del conjunto de datos a fin de minimizar su variabilidad original. En la segunda fase llevamos a cabo la modelización: diseñamos un esquema basado en RF que nos permitirá establecer la función objetivo para predecir el valor de la respuesta a un conjunto de parámetros dados; todo ello con el fin de obtener una metodología más económica y transparente que la basada en el uso de ANN. En la tercera fase, bajo un esquema de GA, proponemos integrar la medida de importancia de las variables dada por el RF en el cruce de los cromosomas y definimos la función de evaluación de las nuevas generaciones mediante interpolaciones entre los nodos del RF. El resto del artículo se estructura como sigue. La sección 2 describe brevemente los fundamentos teóricos de RF y GA que dan origen al presente estudio. En la sección 3 se establece la metodología propuesta. Un caso de ilustración se presenta en la sección 4 a fin de validar nuestra propuesta y establecer las respectivas comparaciones numéricas. Finalmente se resalta la eficiencia de nuestra propuesta como conclusiones en la sección 5.

2. Fundamentos teóricos

2.1. Random Forest (RF)

Esta técnica se basa en la construcción de árboles de predicción mediante el empleo de Bootstrap y Bagging, lo que garantiza la estabilidad del proceso. Cada árbol es construido usando muestras bootstrap con reposición a fín de corregir el error de predicción que se genera a consecuencia de la selección específica de una muestra y para disponer, por cada árbol, de una muestra independiente *out-of-bag* para la estimación del error de clasificación; puesto que aproximadamente un tercio de la muestra original queda excluida de cada muestra generada por bootstrap. Para cada división de un

nodo, no se selecciona la mejor variable de entre todas como en CART, sino que se selecciona al azar un conjunto de variables de un tamaño previamente establecido y se restringe la selección de la variable de división a dicho conjunto. De esta forma se incluye una mayor variabilidad de árboles y se reduce la dependencia del resultado con las divisiones precedentes.

El proceso *out-of-bag* (OOB) consiste en usar el conjunto train **T** para crear otras k muestras bootstrap $\mathbf{T}_k$, se construyen los árboles $h(\mathrm{x}, \mathrm{T}_k)$ y el promedio de ellos será el predictor bagged. Posteriormente para cada $(y, \mathbf{x})$ de **T** se construyen los árboles en cada $\mathbf{T}_k$ que no contienen a $(y, \mathbf{x})$, es decir, las muestras que quedaron fuera de las muestras bootstrap, siendo éstos los clasificadores OOB que permitirán estimar el error de clasificación sobre el conjunto **T**. Las muestras OOB también son usadas en RF para calcular la fuerza de predicción de cada una de las variables usada, conociéndose esto como la *importancia de las variables*, que está condicionada a su interacción con el resto de las variables.

Segal [12], Trevor *et al* [13] y Siroky [14] atribuyen a RF la virtud de reducción de la dependencia entre árboles en la determinación de los nodos mediante la elección aleatoria de conjuntos de predictores en cada árbol. Breiman [15] sugiere determinar previamente el número de variables a elegir en cada nodo (m_{try}), para el caso de árboles de regresión recomienda $m_{try} = \frac{p}{3}$, donde p es el número de predictores en la base de datos. Vale la pena acotar que, como veremos en nuestra propuesta de modelización, dicho parámetro puede ser optimizado mediante validación cruzada u OOB. Finalmente, a efectos del presente trabajo se resume el algoritmo RF mediante los siguientes pasos.

- Se toman B muestras bootstrap de tamaño N del conjunto train.
- Se crean T_b, $(b = 1, ..., B)$ árboles con las muestras hasta que se obtiene el tamaño mínimo en el nodo terminal. Esto se logra de forma recursiva mediante los siguientes pasos:
  1. Seleccionar aleatoriamente m_{try} variables del conjunto total de P variables.
  2. Seleccionar la óptima variable de división entre las p variables.
  3. Dividir el nodo en dos nodos hijos.
- El conjunto de salida es el ensamble (promedio) de los $\{T_b\}_1^B$ árboles, es decir:

$$\hat{f}_{RF}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \tag{2.1}$$

- La estimación de la tasa de error o error de clasificación se obtiene mediante el conjunto OOB.

2.2. *Algoritmos Genéticos (GA)*

Los algoritmos genéticos (GA) son métodos adaptativos inspirados en la *teoría biológica de la evolución* formulada por Darwin a mediados del siglo XIX. En la naturaleza, los individuos de una población permanecen en constante competencia por recursos como agua, refugio y comida. Los individuos con mayor éxito en tal lucha tienen más probabilidades de sobrevivir y de tener una descendencia mayor. Al contrario, individuos peores adaptados tienen un número menor de descendientes, o incluso ninguno. Todo ello implica que los genes de los individuos mejor adaptados se propagarán a través de las generaciones. La combinación de características buenas a través de los ancestros puede dar lugar a descendencias mejores adaptadas que los padres. De ésta manera, las especies evolucionan adaptándose cada vez más a su entorno a través de las generaciones.

Tales ideas se transfieren a los problemas de optimización de forma bastante natural. Las soluciones factibles de un problema específico corresponden a los miembros de una especie particular, donde la aptitud de cada miembro se mide por el valor de la función objetivo. La *población* actual en cada iteración (generación) consiste en un conjunto de soluciones de prueba y se entiende como los miembros vivos de la especie. Algunos de los miembros más jóvenes de la población (en especial los miembros mas aptos) sobreviven a la adultez y se convierten en *padres* (aparejados de forma aleatoria) que tendrán *hijos* (nuevas soluciones de prueba) que tienen algunas de las características (genes) de los padres. Como los miembros más aptos de la población tienen una mayor probabilidad de convertirse en padres que los otros, el GA tiende a crear *poblaciones mejoradas* a medida que avanzan las generaciones. De vez en cuando ocurren *mutaciones*, de modo que las nuevas generaciones pueden adquirir características que no poseen los padres. Éste fenómeno ayuda a los GA a explorar una parte de la región factible, quizás mejor, que la considerada con anterioridad. Finalmente, la

supervivencia del más apto tiende a conducir al GA hacia una solución de prueba (la mejor de todas las consideradas) que al menos es mas cercana al óptimo.

Aunque la analogía con el proceso de la evolución biológica define la esencia de cualquier GA, no es necesario adherirse rígidamente sino que debe ser considerada como punto de partida para definir los detalles del GA que se ajuste mejor al problema bajo consideración [17]. Apoyándonos en tal consideración presentamos, a continuación, un híbrido que combina eficientemente las cualidades de RF y GA para el diseño de parámetros, el cual hemos definido como el método Forest-Genetic.

3. Metodología propuesta

Para el diseño de la metodología Forest-Genetic, suponemos que la respuesta y_{ijkl} es determinada por:

$$y_{ijkl} = f_i\left(\mathbf{X}_k, M_j, Z_l\right) + \varepsilon_{ijkl} \tag{3.1}$$

donde $f_i\left(\mathbf{X}_k, M_j, Z_l\right)$ representa la función de la $ijkl$-ésima respuesta con el correspondiente k-ésimo combinación del vector de factores control, el j-ésimo nivel del factor señal y el l-ésimo nivel del factor ruido; mientras que ε_{ijkl} representa el error aleatorio.

El método Forest-Genetic consta de tres fases: normalización, modelización y optimización. Como fase inicial proponemos la preparación previa del conjunto de datos mediante funciones de normalización, lo que obedece a dos razones. En primera instancia, la normalización permite reducir la variabilidad propia del conjunto de datos, por lo que no sólo nos concentramos en realizar la respectiva comparación, sino que seleccionaremos la mejor (en términos de variabilidad) de entre tres propuestas distintas. Además nuestra metodología pretende ser una alternativa en el uso de las ANN como técnica para el establecimiento de la relación input/outputs, por lo que a fines de comparaciones numéricas es vital la igualdad de escalas en el conjunto de datos. El cuadro 1 muestra las funciones de normalización a comparar, propuestas por Villa-M *et al* [16].

Cuadro 1. Funciones de normalización

$f_1(x) = \begin{cases} 0 & x = 0 \\ 1 - (0{,}85)^a & \text{en otros casos} \end{cases}$	donde $a = (\frac{x}{min})^{\beta}$, $\beta = -(\frac{2{,}4573}{ln(\frac{min}{max})})$
$f_2(x) = \begin{cases} 0 & x = 0 \\ \frac{1}{1+\exp(-x')} & \text{en otros casos} \end{cases}$	donde $x' = \frac{x-min}{max-min}$
$f_3(x) = \begin{cases} 0 & x = 0 \\ \frac{1}{1+\exp(-x')} & \text{en otros casos} \end{cases}$	donde $x' = \frac{x-\mu_x}{\sigma_x}$

La fase de modelización, corresponde a un esquema de trabajo cuyo producto final será la predicción de la respuesta para cada conjunto de valores de los factores de control mediante el algoritmo RF. Tal algoritmo posee como base de entrenamiento árboles de regresión, por tal razón es necesario el ajuste de sus parámetros propios: número mínimo de observaciones en cada nodo, número mínimo de observaciones en los nodos terminales y el parámetro de complejidad; denotados como *minsplit*, *minbucket* y cp respectivamente. Tales ajustes garantizan la creación de árboles complejos al menor coste de la tasa de error. Además, se deben determinar los parámetros propios de RF, como lo son el número de árboles a ensamblar (n_{tree}) y el número de variables de la muestra aleatoria que serán candidatos en cada división (*mtry*);

éste último parámetro debe ser optimizado mediante un estudio preliminar de la tasa de error OOB.

Para finalizar la metodología propuesta, se desarrolla la fase de optimización como un híbrido entre las ventajas que ofrece RF en el reconocimiento de patrones y la eficiencia de GA como metaheurística. El objetivo es la determinación de los niveles de los parámetros que proporcionen el valor óptimo de la variable respuesta acorde a la característica de calidad predeterminada. El proceso se inicia formando aleatoriamente n cromosomas, compuestos por $p = k + j + l$ genes, que corresponden a los k elementos del vector de factores de control, j niveles del factor señal y l niveles del factor ruido. Denotaremos a las poblaciones como a_z, tal que la población inicial $z = b$ es denotada como $a_z = a_b$. Donde, a_{bi} representa el i-ésimo cromosoma de la población b.

La nueva generación estará formada por la unión de los cromosomas padres a_{bi} y los hijos $a^{'}_{bi}$ con sus correspondientes valores respuesta $\hat{y}_i$. Esto es $a_{Bi} = a_{bi} \bigcup a^{'}_{bi}$. Como condición de parada se comparan las respuestas estimadas de la generación actual a_{Bi} con la inicial a_b en consideración a la característica de calidad bajo estudio y la convergencia de los cromosomas en un mismo rango de respuestas.

RF es un algoritmo bajo el esquema CART, por lo cual, toda nueva observación (cromosoma) será ajustada en los límites del nodo correspondiente, lo que limita al GA en su campo de exploración. Por tal razón definimos nuestra función fitness como una función de predicción de los nuevos cromosomas, mediante la interpolación entre nodos como se expresa en la ecuación (3.2)

$$\hat{y}_{h_i} = \left| \frac{\hat{y}_{m_i}\left(g_{p_i} - g_{h_i}\right) - \hat{y}_{p_i}\left(g_{m_i} - g_{h_i}\right)}{g_{p_i} - g_{m_i}} \right| \tag{3.2}$$

donde $\hat{y}_{h_i}$ representa la estimación de la respuesta del i-ésimo cromosoma hijo, $\hat{y}_{m_i}$ y $\hat{y}_{p_i}$ la predicción de la i-ésima madre y el i-ésimo padre provenientes de RF. Finalmente g_{p_i}, g_{m_i} y g_{h_i} corresponden a los valores del gen del i-ésimo padre, i-ésima madre e i-ésimo hijo que serán determinados por las medidas de importancia de las variables proveniente de RF. Tales medidas de importancia, las usaremos para dar pesos a los genes y de esta forma ayudar al GA en su búsqueda de forma más dirigida. Así, aquel gen que obtenga el mayor peso será el usado en la ecuación (3.2). Dichos pesos, serán determinados mediante la ecuación (3.3).

$$PC = \frac{I_{x_k}}{\sum_{k=1}^{p} I_{x_k}} \tag{3.3}$$

donde I_{x_k} representa la importancia del k-ésimo gen en RF, $(k = 1, ..., p)$.

Para la formación de las siguientes generaciones, se adopta el cruce simple de 1 punto pero con una variante: la asignación de pesos a los genes del vector de factores control $\mathbf{X}$ mediante la medida de importancia de las variables calculadas en (3.3). De esta forma, aumentamos la probabilidad de cruce de los cromosomas en torno a los genes de mayor importancia en el diseño.

A continuación se presenta en detalle el procedimiento de cada fase y se ilustra con la figura 1.

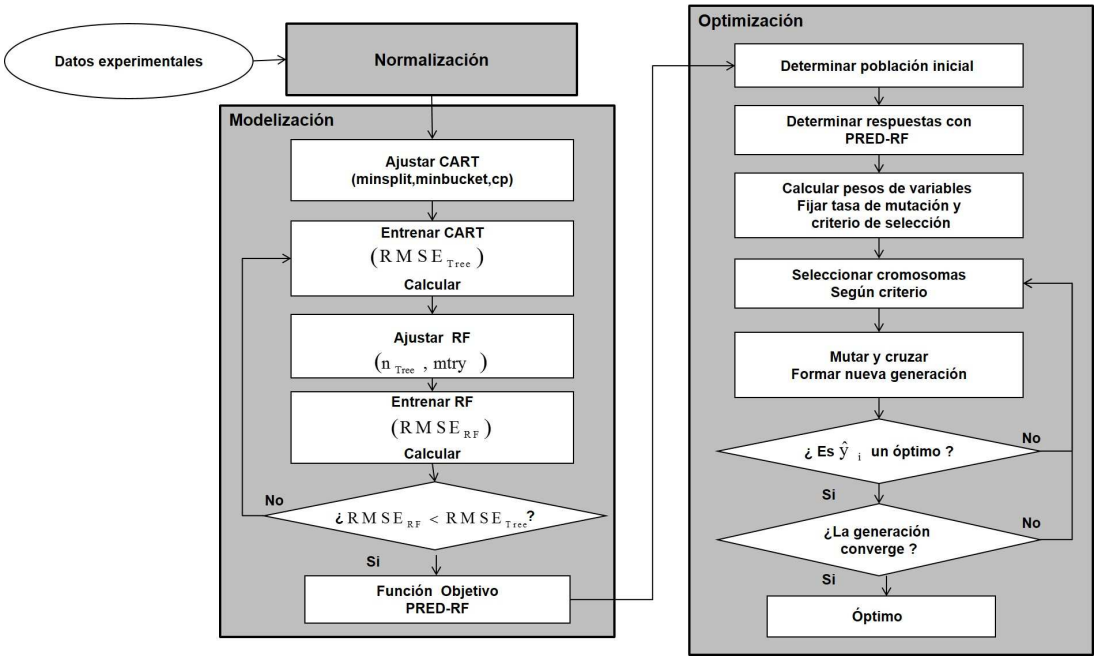

Figura 1. Algoritmo Forest-Genetic univariante

Fase Normalización

Initialization: Hacer $i = p, \quad p = \{1, 2, 3\}$

Step 1: Aplicar normalización $f_i(x)$ al conjunto de datos originales.

Fase Modelización

Step 2: Ajuste de CART mediante la muestra completa normalizada para determinar $minsplit, minbucket$ y cp, así como su $RMSE$, el cual denotaremos como $RMSE^i_{tree}$. Determinar los parámetros de RF n_{tree} y $mtry$; éste último parámetro debe ser optimizado mediante OOB.

Step 3: Dividir aleatoriamente la muestra original en conjunto train y conjunto test.

Step 4: Entrenar RF con el conjunto train. Obtener las predicciones de las respuestas para el conjunto test mediante dicho modelo.

Step 5: Calcular el $RMSE_{RF}$ obtenido con las respuestas predichas para conjunto test. Si $RMSE_{RF} < RMSE^i_{tree}$ ir a step 6. En otro caso, volver a step 2 para obtener nuevos valores de los parámetros $minsplit, minbucket, cp$ y $RMSE^i_{tree}$.

Step 6: Si $i = 3$ ir a paso 7, en caso contrario hacer $i = p + 1$

Step 7: Determinar $f_i^* = arg \min_{i=1,2,3}(RMSE^i_{RF})$, es decir, la normalización que proporciona el menor RMSE con el algoritmo RF.

Fase Optimización

Step 8: Hacer z=b

- Emplear PRED-RF para determinar las respuestas de la población inicial a_b .
- Aplicar la ecuación (3.3) para la asignación del peso de importancia a cada gen en el vector $\mathbf{X}$.
- Llevar el conjunto de cromosomas con sus correspondientes predicciones a escala inicial. Así queda conformada la población inicial a_b.

- Definir la tasa de mutación t y el criterio de selección de los individuos más aptos.

Step 9: Evaluación. Determinar el valor de respuesta y_i más cercana a la característica de calidad e ir a step 10.

Step 10: Selección. Seleccionar los individuos más aptos de acuerdo al criterio definido en el step 8. A ésta selección la definimos como a'_{bi}.

- **Cruce**: Determinar aleatoriamente el gen del vector **X** como punto de cruce y realizar cruce en a'_{bi}.
- **Mutación:** Llevar a cabo la mutación de los genes mediante la tasa t definida en step 8.
- **Predicción para la nueva generación:** $\forall a'_{bi}$ determinar el valor de respuesta $\hat{y}_i$ mediante la ecuación (3.2).

Step 11: Reemplazo. Hacer $a_{Bi} = a_{bi} \bigcup a'_{bi}$.

Step 12: Convergencia :

12.1 Comparar el valor de respuesta y_i del paso 1 con las respuestas $\hat{y}_i$, ¿es $\hat{y}_i$ un óptimo, $\forall \hat{y}_i \epsilon a_{Bi}$?. En caso afirmativo ir a 12.2. En caso contrario ir a paso 10.

12.2 ¿$a_{z=B}$ converge? en caso afirmativo ir a step 13, caso contrario hacer $z = b + 1$ e ir a step 10.

Step 13: Obtener la combinación óptima de los niveles de los parámetros en **X** de la respuesta $\hat{y}_i$.

4. Caso de ilustración

El cuadro 2 muestra un estudio adoptado por [7] y corresponde a un proceso de moldeo por inyección asistida por gas. Los datos se presentan en un L_{18} con 8 factores de control y 5 repeticiones (muestras), donde la respuesta y_i, $(i = 1, 2, ..,5)$ representa la longitud en el canal de gas. Los factores de control son: temperatura del molde, temperatura de fusión, velocidad de inyección, tiempo de inyección, presión, distancia, tiempo de retardo y tiempo de presión constante, denotados por A,B,C,D,E,F,G and H, respectivamente.

Cuadro 2. Factores control y valores respuestas del experimento

No.	Factores control								Respuestas				
	A	B	C	D	E	F	G	H	y_1	y_2	y_3	y_4	y_5
1	50	230	50	1	90	64	0	0	42	40	57	68	74
2	50	230	60	1.5	110	65	0.5	3	71	76	74	74	75
3	50	230	70	2	130	66	1	6	84	80	83	80	82
4	50	240	50	1	110	65	1	6	37	29	34	38	41
5	50	240	60	1.5	130	66	0	0	117	115	121	123	116
6	50	240	70	2	90	64	0.5	3	37	36	36	39	36
7	50	250	50	1.5	90	66	0.5	6	85	87	88	93	90
8	50	250	60	2	110	64	1	0	28	26	24	25	29
9	50	250	70	1	130	65	0	3	84	79	84	79	73
10	60	230	50	2	130	65	0.5	0	74	84	64	69	65
11	60	230	60	1	90	66	1	3	84	87	95	88	94
12	60	230	70	1.5	110	64	0	6	71	68	68	70	65
13	60	240	50	1.5	130	64	1	3	25	24	25	28	24
14	60	240	60	2	90	65	0	6	88	88	89	90	79
15	60	240	70	1	110	66	0.5	0	114	124	125	117	118
16	60	250	50	2	110	66	0	3	106	106	104	99	107
17	60	250	60	1	130	64	0.5	6	31	41	43	36	40
18	60	250	70	1.5	90	65	1	0	60	53	58	51	60

En la fase de modelización se ha desarrollado el algoritmo basado en ANN propuesto por [7] en términos de RMSE a efectos comparativos. El conjunto de datos es dividido aleatoriamente en train y test. El conjunto train consta de 72

observaciones y el conjunto test consta de 18 observaciones. Cabe resaltar que toda nuestra propuesta ha sido programada bajo lenguaje R, empleando además los package rpart y randomForest en las versiones 3.1-46 y 4.5-33 respectivamente.

4.1. ***Control de parámetros***

Las figuras 2 y 3 muestran el estudio del parámetro cp en contraste con el error relativo. El objetivo es elegir el árbol más complejo posible (mayor número de divisiones) al menor coste en la tasa de error. En dicha figura se puede observar que la tasa de error se estabiliza a partir del árbol de tamaño 8 entonces, podemos elegir $size = 15$ con un $cp = 0{,}00016$.

Breiman [15] define $m_{try} = \frac{p}{3}$, entonces $m_{try} = \frac{8}{3} = 2,67 \approx 3$. Ahora bien, el algoritmo RF permite monitorizar este parámetro mediante la tasa de error OOB. A tal fin se presenta la figura 3 donde se muestra el comportamiento de dicha tasa para distintos valores de m_{try}, fijándose de esta manera el parámetro $m_{try} = 4$.

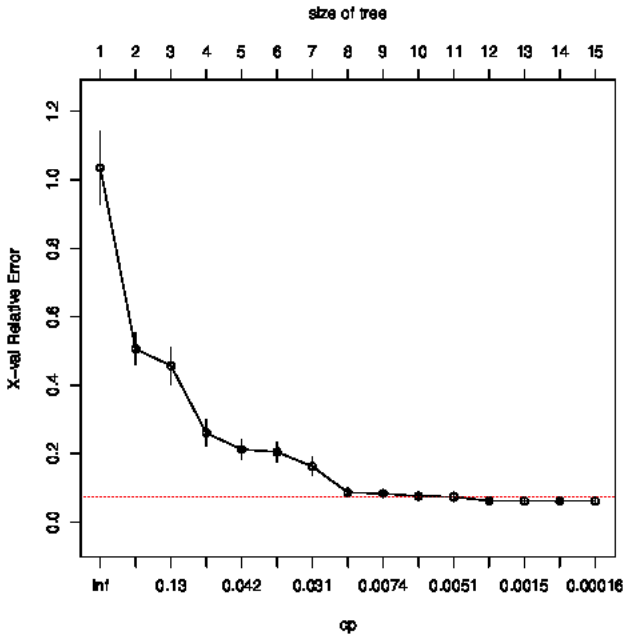

Figura 2. Ajuste del parámetro cp

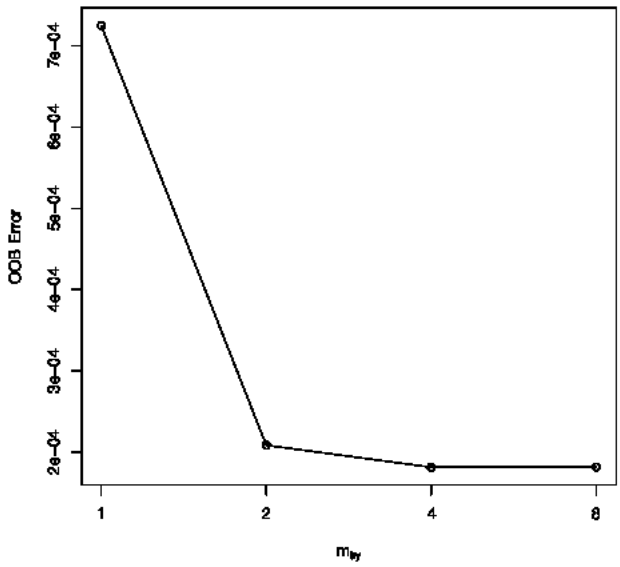

Figura 3. Ajuste del parámetro m_{try}

A continuación se presentan en resumen los ajustes necesarios para la fase de modelización.

- **CART**: $minsplit = 10, minbucket = 2, cp = 0{,}00016$.
- **Algorithm RF**: $n_{Tree} = \{1000, 5000, 10000, 15000\}$,$m_{try} = 4$.
- **Algorithm ANN**: Software Qnet 2000, ANN en Backpropagation, Normalización mediante la función sigmoid. Arquitecturas: $8-3-1, 8-4-1, 8-5-1, 8-6-1, 8-7-1, 8-8-1$ en las iteracciones {1000, 5000, 10000 y 15000 respectivamente}.

4.2. *Fase normalización*

La figura 4 muestra la variabilidad del conjunto de datos según las tres funciones de normalización (f_1, f_2, f_3) del cuadro 1. Es f_2 quien posee el menor valor en cuanto a desviación estándar y por ende concentra mejor el conjunto de datos. Lo que puede conducir a RMSE más bajos y por lo tanto un mejor ajuste de la función objetivo deseada. Sin embargo, a efectos de mayor rigurosidad en la elección de la función de normalización se realiza un ANOVA bajo un diseño completamente aleatorizado, el cual arroja diferencias altamente significativas ($p - value = 0{,}000137$) entre las funciones de normalización. Las medias y desviaciones estándar por función de normalización se muestran en la figura 4, donde si bien es cierto f_2 posee el mayor valor de media, (f_1 y f_3 poseen menor media por la presencia de valores extremos) también es cierto que su desviación estándar es muy inferior con respecto al resto de funciones; lo cual a efecto de los objetivos del presente estudio proporciona mayor estabilidad estadística en los resultados de los RMSE.

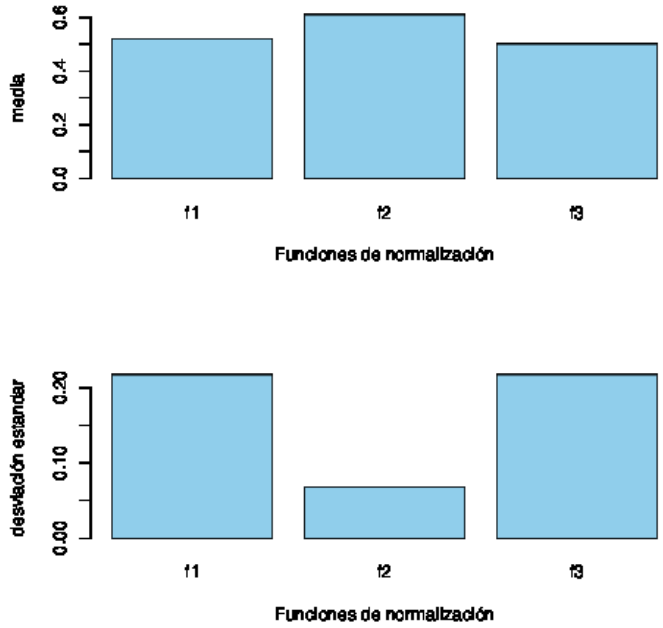

Figura 4. Media y Desviación estándar por función de normalización

4.3. *Fase modelización*

Su y Chang [7] en su estudio, recomiendan como mejor arquitectura la $8-5-1$ por lo que, a nuestros fines comparativos, tomaremos dicha arquitectura para la comparación de los resultados obtenidos y que denotamos como ANN851. Se presentan los resultados en términos de RMSE de ANN851 y RF mediante el cuadro 3, donde para la iteración 1000 ANN851 se muestra totalmente ineficiente dado al gran valor de RMSE alcanzado. Si consideramos ahora la iteración 15000 es precisamente ANN851 quien presenta los valores mas bajos de RMSE pero con valores muy cercanos a los alcanzados por RF, por lo que se puede decir en términos del número de iteraciones, que es RF quien permanece como el mejor método en cuanto al consumo de recursos computacionales.

Para cuantificar lo anteriormente dicho, se define la medida **P** % como el porcentaje de *progreso* que implica el uso del algoritmo RF frente al algoritmo ANN851 en términos de RMSE. Los resultados se muestran de nuevo en el cuadro 3, donde se puede observar que a tan solo 1000 iteraciones RF presenta un progreso del 73.86 %. También se puede verificar la estabilidad de los valores RMSE en RF para todas las iteraciones, lo cual no sucede para ANN851 quien mejora sus valores al aumentar el número de iteraciones al punto de alcanzar un progreso del 36.14 % frente a PRED-RF cuando se ejecutan 15000 iteraciones.

Cuadro 3. Mejora (en %) de los valores de RMSE de PRED-RF con respecto a PRED-ANN851

n	RMSE		
	PRED-RF	PRED-ANN851	**P** %
1000	0,0135	0,0517	**73,86**
5000	0,0134	0,0158	15,45
10000	0,0134	0,0153	12,88
15000	0,0135	0,0099	**−36,14**

La función objetivo producto de esta fase de modelización es llevada a escala inicial a fin de su respectiva optimización en la siguiente fase. Por tanto, vale la pena comparar nuevamente los RMSE de RF y ANN851 en escala inicial

considerando tanto el conjunto train como el test. Lo anterior se presenta en el cuadro 4, así como la medida **P** %. Se puede ver como con 1000 iteraciones se alcanza una mejora del 75 % y 77 % de RF sobre ANN851 en los conjuntos train y test respectivamente. Análogamente, vemos como ANN851 en su valor más alto de iteraciones (15000) alcanza un rendimiento del 28 % y 4 % en los conjuntos train y test respectivamente. Todo lo anterior se traduce en eficiencia estadística de nuestro proceso de modelización en contraste con las ANN quienes en la iteración 1500 evidencia un claro sobreajuste.

Cuadro 4. Mejora (en %) de los valores de RMSE de RF con respecto a ANN851, en conjuntos de train y test

n	Train			Test		
	RF	ANN851	**P** %	RF	ANN851	**P** %
1000	6,17	24,75	**75,07**	6,13	26,94	**77,24**
5000	6,16	5,69	−8,13	6,00	7,35	18,40
10000	6,15	5,21	−18,06	6,03	6,42	5,93
15000	6,14	4,78	**−28,44**	6,14	5,92	**−3,73**

4.4. *Fase optimización*

Se estiman las respuestas de 400 cromosomas generados aleatoriamente con la estructura del vector $X = \{A, B, C, D, E, F, G, H\}$ mediante nuestro algoritmo RF y se calculan los pesos de cada gen mediante la función (3.3) como lo muestra el cuadro 5.

Cuadro 5. Medidas de importancia y pesos de los genes según RF

Gen	A	B	C	D	E	F	G	H
I_{x_k}	10.134	28.601	18.318	16.1313	17.810	**64.761**	40.125	20.995
Pesos	0.047	0.132	0.084	0.074	0.082	**0.299**	0.185	0.097

El cuadro anterior también nos permite determinar nuestra función fitness. Como se puede observar, F es el gen con mayor peso, por lo que la ecuación (3.2) queda denotada en (4.1)

$$\hat{y}_{h_i} = \left| \frac{\hat{y}_{m_i}\left(f_{p_i} - f_{h_i}\right) - \hat{y}_{p_i}\left(f_{m_i} - f_{h_i}\right)}{f_{p_i} - f_{m_i}} \right| \tag{4.1}$$

donde f_{p_i}, f_{m_i} y f_{h_i} corresponden a los valores del gen F del i-ésimo padre, i-ésima madre e i-ésimo hijo correspondientemente. El cuadro 6, compara nuestro resultado con los obtenidos por Su y Chang[7] y Chiu [9], así como con el resultado obtenido por la metodología Taguchi, (reflejado en [7]).

Cuadro 6. Comparación de los resultados en diferentes propuestas de optimización

Método	Factores control								$\hat{y}$
	A	B	C	D	E	F	G	H	
Taguchi	50	240	50	2	130	64	1	3	19.8
Chiu *et al*	50	240	50	2	130	63.5	1	6	13.5
Su y Chang	48.2	235	46	0.85	85.1	64	1	6	7.4
Forest-Genetic	50.16	235.02	50.96	0.24	91.44	64.16	0.25	0	**2.45**

Se puede observar como los resultados presentados por el método Taguchi y la propuesta por Chiu [9], basada en ANN, son ligeramente diferentes en la combinación de los parámetros pero si en la respuesta estimada. Diferencias

mucho más notorias se muestran en la propuesta de Su y Chang [7], basada en ANN y SA, con una estimación casi a la mitad del obtenido por Chiu [9]. Finalmente, se presentan nuestros resultados, donde no solo minimizamos la respuesta estimada en aproximadamente 5 puntos sino que una de las diferencias más notorias en la combinación de los parámetros es presentada por el valor de H, quien corresponde a uno de los factores con mayor peso de importancia en la cuadro 5, corroborando nuevamente la eficiencia de nuestro algoritmo de modelización.

5. Conclusiones

El Diseño de parámetros, propuesto por Taguchi, tiene como objetivo reducir la variación de los productos y procesos para seleccionar el conjunto de niveles de los factores de control que proporcionen el mejor rendimiento y menor sensibilidad a los factores de ruido. Ese mejor rendimiento ha sido identificado para la selección de niveles discretos definidos en los factores, pero en el caso en que esos factores sean en realidad continuos, se podría avanzar más en la optimización. El método Forest-Genetic constituye una alternativa para la mejora en el diseño de parámetros que se desarrolla en tres fases: normalización, modelización y optimización. Combinamos eficientemente las ventajas que ofrece el algoritmo RF en el reconocimiento de patrones e integramos sus medidas de importancia en los operadores genéticos de GA. Dos casos de ilustración nos han permitido validar nuestra propuesta y contrastar nuestros resultados con los más recientes aportes hallados en la literatura. Resumimos los méritos de nuestra propuesta en los siguientes puntos:

- Forest-Genetic no presupone independencia entre los factores ni linealidad de los factores con las respuestas, por tanto este método es perfectamente aplicable a casos de correlaciones entre factores y relaciones no lineales.
- Forest-Genetic fué diseñado considerando factores de ruido y señal con características de calidad de tipo dinámico o estático, por tanto puede ser aplicado en casos de ausencias o presencia de dichos factores y en cualquier tipo de característica de calidad.
- Forest-Genetic permite optimizar la función de normalización, en virtud a la dispersion de los datos y el rendimiento general de los algoritmos.
- La fase de modelización de Forest-Genetic se basa en CART y RF, por tanto puede ser usado en presencia de parámetros de tipo continuo y/o discreto, ampliando así su aplicación real.
- La fase de modelización de Forest-Genetic esta basada en CART y RF lo que minimiza el riesgo de sobreajuste de los modelos.
- La función fitness y el esquema de cruce diseñado en la fase de optimización permite a Forest-Genetic la dirección más eficiente en los hiperplanos de búsqueda.

Como se puede notar, todo lo anterior obedece a características teóricas que dotan a Forest-Genetic de buenas propiedades. Sin embargo vale la pena mencionar que el caso ilustrativo nos permitió comparar numéricamente con las últimas propuestas metodológicas halladas en la literatura (cuadro 6). Verificamos que nuestra estrategia de modelización se presenta más robusta y estable en presencia de diferentes números de iteraciones en comparación con las ANN. Así mismo, observamos que Forest-Genetic requieren un número inferior de iteraciones tanto en la fase de modelización como en la de optimización para alcanzar buenos resultados en comparación con las metodologías que comprenden el uso de ANN y SA.

6. Referencias

[1] G. E. P. Box y R. D. Meyer. Dispersion effects from fractional designs. *Technometrics*, 28(1):19-27, 1986.
[2] T. P. Ryan. *Statistical methods from quality improvement*. Jhon Wiley Sons, 1989.
[3] S. Maghsoodloo, G. Ozdemir, V. Jordan y C.-H. Huang. Strengths and limitations of Taguchi's contributions to quality, manufacturing and process engineering. *Journal of Manufacturing Systems*, 2(23):73-126, 2004.
[4] K. Tsui. A critical look at Taguchi's modeling approach for robust design. *Journal of Applied Statistics*, 1(26):81-98, 1996.
[5] G. E. P. Box y R. D. Meyer. *Introduction to statistical quality control*. Wiley Sons, 2001.
[6] A. Miller y C.F. Wu. Manufacturing Quality Control By Means Of A Fuzzy ART Network Trained On Natural Process Data, *Statistical Science*, 11(2):122-136, 1996.
[7] C.-T. Su y H.-H. Chang. Optimization of parameter design: an intelligent approach using neural network and simulated annealing. *International Journal of Systems Science*, 31(12):1543-1549, 2000.

[8] C. Zang, M.I. Friswell y J.E. Mottershead. A review of robust optimal design and its application in dynamics. *Computer and Structures*, 8(3):315-326, 2005.
[9] C.-C. Chiu, C.-T. Su, G-H Yang, J.-S. Huang , S.-C. Chen y N.-T. Cheng. Selection of optimal parameter in gas-assisted injection moulding using a neural network model and the Taguchi method. *International Journal of Quality Science*, 2:106-120, 1997.
[10] K. M. Tay y C. Butler. Modeling and optimizing of a mig welding process-A case study using experimental designs and neural networks. *Quality and Reliability Engineering International*, 13:61-70, 1997
[11] H.-H. Chang. Applications of neural networks and genetic algorithms to Taguchi's. *International journal of electronic business management*, 3(2):90-96, 2005.
[12] M. R Segal. *Machine Learning Benchmarks and Random Forest Regression*. Center for Bioinformatics and Molecular Biostatistics. University of California, San Francisco, 2003.
[13] H. Trevor, R. Tibshirani y J. Friedman. *The elements of statistical learning. Data mining, inference and prediction*. Springer series in statistic, 2009.
[14] D. S. Siroky. Navigation random Forest and related advances in algorithmic modeling. *Statistic Surveys*, 3:147-163, 2005.
[15] L. Breiman. Random Forest. *Machine Learning*, 45:5-32, 2001.
[16] A. Villa-Murillo, A. Carrión y S. San Matías. Modeling response variables in Taguchi design parameter using CART and Random Forest based system. *Communications in dependability and quality management. An international journal*, 15(4):5-15, 2012.
[17] F. Hillier y G. Lieberman. *Introducción a la investigación de operaciones*. McGraw Hill, 2006.