

Metodología de optimización de la calidad de productos

RECIBIDO: 16/02/15 ACEPTADO: 12/11/15

JUAN M. CEVALLOS*
 LUIS R. RAEZ**

RESUMEN

En este estudio se ha desarrollado un método para optimizar los parámetros de calidad de productos que consta de cinco pasos: 1) Determinar las características de calidad del producto y las variables del proceso 2) Desarrollar un diseño experimental con Métodos Taguchi 3) Desarrollar los experimentos con Metodología de Superficie de Respuesta. 4) Determinar una red neuronal que represente las relaciones entre las variables y características de calidad. Usar variables difusas si se tiene información no determinística. 5) Optimizar con el uso de algoritmos genéticos. En esta propuesta, las redes neuronales artificiales RNA permiten estimar funciones de respuesta, en el caso de contar con variables cualitativas se las procesa con lógica difusa LD y en el paso de optimización se usan algoritmos genéticos AG. Se presenta un ejemplo de optimización con respuestas múltiples para comprobar el método.

Palabras clave: algoritmos genéticos, diseño de experimentos, lógica difusa, optimización de la calidad, redes neuronales artificiales

OPTIMIZATION METHODOLOGY OF THE QUALITY OF PRODUCTS

ABSTRACT

In this study we have developed a method for optimizing the parameters of quality of products that consists of five steps: 1) Determine the characteristics of product quality and process variables 2) Develop an experimental design with Taguchi Methods 3) Develop experiments with Response Surface Methodology. 4) Determine a neural network that represents the relationships between variables and quality characteristics. Using fuzzy variables if there is information not deterministic. 5) Optimize with the use of genetic algorithms. In this proposal, artificial neural networks ANN allow to estimate response functions; in the case of having the qualitative variables these are processed with fuzzy logic LD and in the optimization step genetic algorithms GA are used. An example of optimization with multiple responses is presented to verify the method.

Keywords: artificial neural networks, design of experiments, fuzzy logic, genetic algorithms, quality optimization

1. INTRODUCCIÓN

Un enfoque clásico para optimizar problemas de respuestas múltiples es aplicar el Diseño de Experimentos DDE, a continuación, modelos de regresión múltiple para estimar las relaciones entre las respuestas y factores controlables; luego se combinan las diferentes respuestas con una función de deseabilidad y finalmente se optimizan los factores controlables. Genichi Taguchi desarrolla la Ingeniería de la Calidad, que es una simplificación de los métodos estadísticos tradicionales para aplicarlos a la mejora de la calidad en las empresas.

El Diseño de Experimentos DDE, es un método de mejora de la calidad recomendado por muchos expertos en mejora del desempeño de procesos y productos. El DDE ayuda a investigar los efectos de los factores controlables y no controlables sobre una o más respuestas de interés. El objetivo del DDE es determinar el conjunto óptimo de valores de los factores controlables de tal forma que se alcancen las características de calidad deseadas con el mínimo de variación. Por lo general la evaluación de la calidad de productos o procesos involucra el estudio simultáneo de varias características de calidad con cada una teniendo su propia importancia relativa para los consumidores. Este problema es referido como un problema de optimización de respuestas múltiples. En este tipo de problema la meta es determinar el conjunto de valores de los factores controlables que permitan la mejor combinación de respuestas. (Montgomery, 2006). Cuando los procesos son complejos el DDE tiene limitaciones y se utilizan métodos evolutivos heurísticos. De esto es justamente lo que trata el presente trabajo. Se desarrolla una propuesta alternativa para situaciones complejas de optimización de la calidad mediante el uso de Redes Neuronales Artificiales, Algoritmos Genéticos y Lógica Difusa.

Según Ortiz *et al.* (2004) y Montgomery (2006), hay diversos métodos para abordar los problemas de optimización de respuesta múltiple.

Un primer grupo consiste en la superposición de curvas de contorno de cada respuesta y con base a ellos se encuentra la región de interés que por las diferentes respuestas es satisfecha. El problema principal con este enfoque es que no se puede identificar la solución más importante. Myers y

* Doctor en Ingeniería. Profesor de la Facultad de Ingeniería Industrial. UNMSM. E-mail: juanmanuelcevallos@gmail.com

** Magister en Ingeniería Industrial. Profesor de la Facultad de Ingeniería Industrial. UNMSM. E-mail: luisraez2010@hotmail.com

Montgomery (2009), manifiestan que este enfoque es eficaz sólo cuando se tiene pocos factores controlables. Un segundo grupo consiste en formular el problema en la forma de un problema de optimización restringida. Kim y Lin (2006) se refieren a este método, como enfoque basado en prioridades. Este enfoque utiliza la respuesta más importante como la función objetivo y el resto de las respuestas son consideradas como restricciones. Ejemplos de este enfoque se puede encontrar en trabajos sobre metodología de superficie de respuesta, como los de Del Castillo y Montgomery, (1993) y Kim y Lin (2006). La principal desventaja de estos enfoques es que no consideran de forma simultánea todas las respuestas. Un tercer método y más general consiste en enfoques que construyen modelos para las respuestas de interés, los cuáles son combinados en un solo valor escalar; y luego se optimiza el modelo de una sola variable por medio de un método de optimización. Para ello se puede utilizar el método de mínimos cuadrados (MC), luego aplicar funciones de deseabilidad y finalmente optimización con técnicas de búsqueda, como la simplex Nelder-Mead, para funciones de superficie de respuesta no diferenciables y gradiente reducido generalizado (GRG) para las funciones objetivo diferenciables, Del castillo et al., 1996. Asimismo, para dicha optimización se pueden aplicar procedimientos heurísticos, tales como los algoritmos genéticos (AG), recocido simulado, o la búsqueda tabú, Ortiz *et al.* (2004). Por otro lado, las Redes Neuronales Artificiales son métodos que permiten establecer relaciones entre variables de gran complejidad con menor error que los métodos estadísticos tradicionales como la regresión lineal múltiple ó la metodología de superficie de respuesta, Haykin (1994); Noorossana (2008). En este trabajo se desarrolla un enfoque basado en los Métodos Taguchi para el diseño experimental, en redes neuronales artificiales (RNA) para obtener los modelos, en lógica difusa para abordar el problema con respuestas cualitativas y se optimiza con el uso de Algoritmos Genéticos AG.

2. MÉTODO

Enfoque Propuesto. La metodología consta de cinco pasos, y por el tipo de técnicas que utiliza permite representar relaciones muy complejas que no se podrían representar por métodos tradicionales.

Los pasos son: 1) Determinar las características de calidad del producto y las variables del proceso 2) Desarrollar un diseño experimental con Métodos Taguchi 3) Desarrollar los experimentos con Metodología de Superficie de Respuesta. 4)

Determinar una red neuronal que represente las relaciones entre las variables y características de calidad. Usar variables difusas si se tiene información no determinística. 5) Optimizar con el uso de algoritmos genéticos. Se utilizan los software Minitab y Matlab.

En el primer paso se determinan las características de calidad con base a lo que requiere el mercado, luego se selecciona el arreglo ortogonal requerido para la información que se tiene. Como tercer paso se desarrolla los experimentos con base en Metodología de Superficie de Respuesta. El cuarto paso consiste en utilizar las redes neuronales RNA para estimar la relación entre los factores controlables como las entradas y las respuestas como las salidas; y se usa lógica difusa para tratar a las variables cualitativas, y, como quinto paso se usan Algoritmos Genéticos AG para la optimización. Este procedimiento nos permite tomar ventaja de las capacidades de las redes neuronales para la aproximación de las funciones; del potencial de la lógica difusa para representar variables cualitativas, y también la aptitud de los AG en la optimización no lineal y compleja de las funciones. Las respuestas cualitativas, se pre procesan y luego se transforma en vectores, utilizando la lógica difusa, de modo que puedan ser analizadas por las redes neuronales. Con el objeto de trabajar con metas para las funciones objetivo a optimizar se usa una función híbrida de algoritmos genéticos con fgoalattain; asimismo, se requiere tener límites para variable de entrada y salida; se trabaja con las variables de entrada y salida y se obtiene un Frente de Pareto (Optimización de primera vuelta – ida), el mismo que sirve de base para una nueva Optimización de primera vuelta –retroceso (donde las variables de entrada y salida son introducidas al revés) y se obtiene un nuevo Frente de Pareto (Optimización de primera vuelta – retroceso). La misma operación se realiza para la segunda vuelta. De esta manera se puede conseguir obtener resultados que están dentro de los límites de entrada y salida.

Las características de calidad que se buscan deben ser obtenidas del mercado; sin embargo, puede ayudar mucho el uso de Normas Técnicas Internacionales o Nacionales, como ISO, ASTM, ASME, CODEX, ANSI, DIN, JIS, AFNOR, NTP, y otras que establecen requisitos de productos terminados.

El diseño experimental nos ayuda a recoger los datos necesarios para la formación de las redes neuronales. Se usan Diseños de Metodología de Superficie de Respuesta MSR como el Diseño compuesto central (CCD), debido a su

capacidad para proporcionar la información requerida, mediante la cobertura del espacio experimental. Tales diseños ayudan a las redes neuronales a aproximar la función del proceso. Para simplificar el diseño experimental se pueden utilizar los Métodos Taguchi, aunque también podrían utilizarse diseños de experimentos tradicionales completos si se tiene pocas variables. El uso de la metodología de superficie de respuesta permite tener mayor información a efectos de facilitar la búsqueda de los valores óptimos.

Se utilizan las RNA para aproximar las relaciones entre las variables. Dependiendo de qué tipo de RNA se utiliza, existen diferentes parámetros que ajustar, pero el concepto común compartido es que todas necesitan ser entrenadas. Se utilizan ejemplos, datos patrón, para entrenar la red neuronal. Cada ejemplo se compone de un par de datos, entrada - salida: una señal de entrada y su correspondiente respuesta deseada para la red neuronal. Así, un conjunto de ejemplos representa el conocimiento sobre el tema de interés, Haykin (1994). Dados los datos para el diseño de una red neuronal; en primer lugar, se selecciona una adecuada arquitectura para la red neuronal. En segundo lugar, un subconjunto de los ejemplos se utiliza para entrenar la red por medio de un algoritmo de aprendizaje adecuado; y en tercer lugar, el desempeño de la red entrenada se prueba con datos que no se ha utilizado inicialmente (generalización).

Con relación a las respuestas cualitativas, en forma de expresiones metalingüísticas como "Muy Bueno", "bueno", "medio", etc, se definen conjuntos difusos, fuzificación. La fuzificación de una respuesta cualitativa significa definir diferentes funciones de membresía para diferentes valores cualitativos de que la respuesta pertenecen un dominio predefinido, Rajasekaran (2003). Las RNA no son capaces de procesar los conjuntos difusos en la forma de funciones de pertenencia o membresía. Por ello, se alimentan a las redes neuronales con la información en la forma de vectores. Para ello, el dominio de los conjuntos difusos se divide en n intervalos iguales y en $n + 1$ puntos de resultados, los grados de pertenencia están representados por un vector con $n+1$ elementos. El vector de $n + 1$ elementos es una aproximación de la función de pertenencia. Al aumentar el valor de n resultará en una mejor aproximación. La determinación de un valor apropiado para n depende de la naturaleza de la función de pertenencia.

Con relación a la aplicación de una red neuronal para la estimación de la respuesta, se busca su relación con los factores controlables. La salida de

la red neuronal entrenada debe tener un número de neuronas igual al número de respuestas. Las entradas para estas redes son factores controlables y las salidas son respuestas. La capa de salida de la red tiene una neurona por respuesta cuantitativa y una por respuestas cualitativas; en cada respuesta cualitativa considerar el número de alternativas de las variables cualitativas, n . Para el entrenamiento se dividen los datos en tres grupos: entrenamiento, validación y prueba. Antes del entrenamiento de la red, los factores significativos para cada respuesta tienen que ser identificados en forma de un subconjunto de todos de los factores controlables que producen el mínimo cuadrado medio del error mse. Las RNA perceptrón multicapa con algoritmo de aprendizaje backpropagation se utilizaron con este objetivo. El modelo que produce el más bajo mse para los datos de prueba contiene los factores significativos. Si la diferencia entre el modelo con todos los factores y un modelo de subconjunto de ellos no es significativa, el modelo completo se prefiere.

Además hay que construir la más adecuada RNA para estimar las respuestas, se puede considerar las redes perceptrón multicapa MLP o de función de base radial RBF, utilizando el criterio del mse para obtener la mejor.

Con relación a la optimización con Algoritmos Genéticos AG, éstos se eligen porque los métodos basados en el gradiente son complejos de utilizar debido a lo difícil de su cálculo con base a las superficies de respuestas. Sin embargo, cuando las redes neuronales se utilizan, no es indispensable una superficie de respuesta. Además los AG son un método de búsqueda heurístico potente para la optimización de funciones no lineales y complejos. Un AG tiene distintos parámetros cuyos valores deben ser determinados antes de iniciar el paso de optimización. Diferente autores, entre ellos Ortiz *et al.* (2004=, han propuesto el uso de diseños de experimentos robustos para determinar el mejor conjunto de los parámetros del AG. Para los AG se deben precisar parámetros como el tamaño de la población, el criterio de selección, el tipo de cruzamientos para la reproducción, las mutaciones y el criterio de parada de búsqueda; Deb, 2009. Los algoritmos genéticos pueden dar resultados más precisos cuando trabajan de manera híbrida con la función fgoalattain que permite establecer metas y pesos, MATLAB Optimization Toolbox (2014).

Los pasos cuarto y quinto se trabajan de manera conjunta debido a que se tiene límites de las entradas (x) y salidas(y); sin embargo los software trabajan con límites de entrada más no de salida; este limitación se supera aplicando la optimización

de ida y retorno; es decir, de ida significa que las entradas son las x y las salidas son las y; en retorno se aplica al revés; es decir, los resultados de la ida se aplican en el retorno como entradas, y viceversa. Con el objetivo de obtener mayor precisión se realiza esta operación 2 veces, o vueltas.

3. RESULTADOS

Ejemplo aplicativo. Los datos de este ejemplo han sido obtenidos del trabajo de Pignatiello, 1993.

Paso 1. Se tienen tres variables controlables, de entrada y dos variables de respuesta, con cuatro réplicas. Paso 2. El experimento puede ser llamado como diseño 2^3 (Arreglo ortogonal 2^3 de Métodos Taguchi) para tres variables controlables con dos niveles cada variable, cuatro réplicas y respuesta bivariada a ser observada. En este ejemplo, el ruido experimental se debe sólo a variaciones al azar por lo que no hay factores ruido explícitos. Los datos experimentales se presentan en la Tabla 1.

Tabla 1. Datos experimentales

X1	X2	X3	Y1	Y2
-1	-1	-1	109.895	67.6974
1	-1	-1	100.192	67.0264
-1	1	-1	106.078	72.9353
1	1	-1	104.120	72.9878
-1	-1	1	113.515	68.2934
1	-1	1	98.732	67.0955
-1	1	1	103.145	71.6818
1	1	1	104.454	76.9003
-1	-1	-1	109.759	67.2374
1	-1	-1	99.634	66.1779
-1	1	-1	105.642	72.8508
1	1	-1	104.802	74.2487
-1	-1	1	111.121	68.4693
1	-1	1	99.357	63.6112
-1	1	1	106.959	76.2657
1	1	1	105.029	77.0322
-1	-1	-1	110.704	67.962
1	-1	-1	100.269	66.5758
-1	1	-1	105.670	72.5768
1	1	-1	104.203	73.9371
-1	-1	1	112.854	68.9576
1	-1	1	102.842	68.6470
-1	1	1	107.620	77.4958
1	1	1	99.786	67.9890
-1	-1	-1	109.773	66.9268
1	-1	-1	100.600	67.9431
-1	1	-1	105.393	72.3754
1	1	-1	104.335	73.2824
-1	-1	1	106.666	64.7051
1	-1	1	94.235	62.4188
-1	1	1	103.440	76.3739
1	1	1	104.923	75.7691

Los valores deseados de salida son $y_1=103$, e $y_2=73$; y se considera un rango de ± 3 .

Paso 3. Se calculan el Análisis de Varianza y las Ecuaciones de Regresión. Con los 32 datos; 3 columnas de x y 2 columnas de y; con el software

Minitab; y con los comandos: Stat/ Anova/ General Linear Model/ Cuadro de dialogo: Responses Y1 Y2; Modelo $x_1|x_2|x_3$ (es decir, los factor individuales y todas sus interacciones); se obtiene los resultados que se presentan en las Tablas 2 y 3.

Tabla 2. Análisis de Varianza y Coeficientes de Regresión de y_1

Análisis de Varianza						
Analysis of Variance for Y1, using Adjusted SS for Tests						
Source	DF	Seq SS	Adj SS	Adj MS	F	P
X1	1	317.022	317.022	317.022	73.90	0.000**
X2	1	0.647	0.647	0.647	0.15	0.701
X3	1	1.276	1.276	1.276	0.30	0.590
X1*X2	1	181.123	181.123	181.123	42.22	0.000**
X1*X3	1	3.922	3.922	3.922	0.91	0.349
X2*X3	1	0.358	0.358	0.358	0.08	0.775
X1*X2*X3	1	1.953	1.953	1.953	0.46	0.506
Error	24	102.956	102.956	4.290		
Total	31	609.256				
**Los únicos significativos						
S = 2.07119 R-Sq = 83.10% R-Sq(adj) = 78.17%						
Coeficientes de Regresión						
Estimated Effects and Coefficients for Y1 (coded units)						
Term	Effect	Coef	SE Coef	T	P	
Constant		104.867	0.3661	286.41	0.000	
X1	-6.295	-3.148	0.3661	-8.60	0.000**	
X2	-0.284	-0.142	0.3661	-0.39	0.701	
X3	-0.399	-0.200	0.3661	-0.55	0.590	
X1*X2	4.758	2.379	0.3661	6.50	0.000**	
X1*X3	-0.700	-0.350	0.3661	-0.96	0.349	
X2*X3	-0.211	-0.106	0.3661	-0.29	0.775	
X1*X2*X3	0.494	0.247	0.3661	0.67	0.506	
S = 2.07119 PRESS = 183.032						
R-Sq = 83.10% R-Sq(pred) = 69.96% R-Sq(adj) = 78.17%						

Por tanto la ecuación de Y_1 es:

$$Y_1 = 104.87 - 3.15x_1 + 2.38 x_1 x_2 \tag{1}$$

Tabla 3. Análisis de Varianza y Coeficientes de Regresión de y2

Análisis de Varianza						
Analysis of Variance for Y2, using Adjusted SS for Tests						
Source	DF	Seq SS	Adj SS	Adj MS	F	P
X1	1	3.894	3.894	3.894	0.80	0.380
X2	1	412.975	412.975	412.975	85.01	0.000**
X3	1	2.511	2.511	2.511	0.52	0.479
X1*X2	1	3.344	3.344	3.344	0.69	0.415
X1*X3	1	6.477	6.477	6.477	1.33	0.260
X2*X3	1	12.082	12.082	12.082	2.49	0.128
X1*X2*X3	1	0.052	0.052	0.052	0.01	0.918
Error	24	116.590	116.590	4.858		
Total	31	557.925				

****El único significativo**

S = 2.20407 R-Sq = 79.10% R-Sq(adj) = 73.01%

Coeficientes de Regresión					
Estimated Effects and Coefficients for Y2 (coded units)					
Term	Effect	Coef	SE Coef	T	P
Constant		70.4515	0.3896	180.82	0.000
X1	-0.6976	-0.3488	0.3896	-0.90	0.380
X2	7.1848	3.5924	0.3896	9.22	0.000**
X3	0.5603	0.2801	0.3896	0.72	0.479
X1*X2	0.6465	0.3233	0.3896	0.83	0.415
X1*X3	-0.8998	-0.4499	0.3896	-1.15	0.260
X2*X3	1.2289	0.6144	0.3896	1.58	0.128
X1*X2*X3	-0.0807	-0.0404	0.3896	-0.10	0.918

S = 2.20407 PRESS = 207.272

R-Sq = 79.10% R-Sq(pred) = 62.85% R-Sq(adj) = 73.01%

Por tanto la ecuación de Y2 es:

$$Y2 = 70.45 + 3.59x2 \quad (2)$$

Luego se tiene:

$$y = \frac{[y1] - [(104.87 - 3.15x1 + 2.38x1x2)]}{y2} \quad (3)$$

Estas ecuaciones permiten obtener la superficie de respuesta, pero como se aprecia de los R² obtenidos de 83% y 79 %, respectivamente, se obtendría determinados valores de optimización con un error mayor aún. Sin embargo la información

que proporcionan las dos ecuaciones es de mucha utilidad para el trabajo de mejora de la calidad. Pasos 4 y 5. Con Redes Neuronales Artificiales RNA y Algoritmos Genéticos AG; utilizando MATLAB.

Primera vuelta; ida:

Los datos de entrada están en la matriz xp (x1, x2, x3) y los de salida en la matriz yp (y1, y2). Con los comandos que se indican se obtiene una RNA.

Se introducen los datos al workspace y los comandos son; >> [trainInd,vallnd,testInd]=

```
dividetrain(32); net=fitnet; [net,tr]=train(net,xp',yp');
outputs= net(xp'); errors= gsubtract (yp', outputs);
performance=perform(net,yp',outputs)
```

Se obtiene: performance = 5.8138

```
Características de la RNA: >> net.layers{1}.size;
ans = 10; >> net.layers{2}.size; ans = 2;
```

```
>> net.layers{1}.transferFcn; ans = tansig; >> net.
layers{2}.transferFcn; ans = purelin
```

Pesos(W) y descentramiento(b), para las capas 1 y 2, se presentan en la Tabla 4.

Seguidamente se optimiza con base a la RNA obtenida; los comandos son:

```
>> optimtool; >> objcFcn=@(xp) sim(net,xp');
Solver: gamultiobj; Fitness Function : @(xp) sim
(net,xp'); N° variables: 3; Bounds lower [-1,-1,-1]
upper [1,1,1]; Hybrid function: fgoalattain (default).
Nota los datos de entrada están en la matriz xp y
los de salida están en la matriz yp.
```

Luego de 132 iteraciones; se obtiene el Frente Pareto de 16 resultados.

Considerando los valores deseados de $y_1=103$ e $y_2=73$; los valores del resultado que mejor

cumple lo buscado corresponden al quinto; siendo $x_1=0.9999$; $x_2=-0.5075$; $x_3=-0.9994$; $y_1= 95.5766$; e $y_2 = 71.3125$

Las x están dentro del rango -1 y 1; pero y_1 está lejos del valor deseado (103+-3) e y_2 está dentro del rango (73+-3). Por tanto se realiza el retroceso de la primera vuelta.

Primera vuelta; retorno:

A continuación se usan los valores del Frente Pareto anteriores para aplicarlos a la RNA y AG; pero al revés. Es decir las entradas serán la matriz xpp que vienen de la matriz yp (anteriores salidas, 2 variables); y las salidas serán la matriz ypp que vienen de la matriz xp (anteriores entradas, 3 variables). Se introducen los datos al workspace y los comandos son: >> [trainInd,vallnd, testInd] = dividetrain(16); >> net=fitnet; >> [net,tr]=train(net,xpp',ypp'); >> outputs= net(xpp'); >> errors= gsubtract (ypp',outputs); >> performance= perform (net,ypp', outputs)

Se obtiene: performance = 4.6995e-04

Con los comandos vistos en la ida primera vuelta se obtiene las características de la RNA obtenida; que por espacio no se presentan nuevamente.

Tabla 4. Pesos (W) y descentramiento (b) de la RNA

Pesos (w) y descentramiento(b) de la capa 1									
>> net.IW{1}:									
2.1026	-2.3565	0.8028;	1.8417	1.7221	-1.8421;	-1.6432	1.9272	1.6552	
1.9830	-0.2397	2.2807;	0.7353	2.7294	1.1263;	-2.0706	-1.7650	1.2748	
-2.0040	-0.8337	2.1289;	0.3658	2.8844	-0.7540;	2.5719	1.6287	0.7684	
1.9112	1.8098	-1.4290							
>> net.b{1}:									
-2.7486;	-2.1468;	1.6224;	-0.8642;	-0.1029;	-0.3491;	-0.9030;	1.6946;		
2.2096;	3.0480								
>> net.LW{1}: []									
Pesos (w) y descentramiento(b) de la capa 2									
>> net.IW{2}: []									
>> net.b{2}:									
0.0164;	-0.5505;								
>> net.LW{2}:									
Columns 1 through 9									
-0.2424	-0.2198	-0.2081	-0.1297	0.5840	-0.3574	0.2172	-0.6196	-.3958	
-0.3671	0.1169	0.1711	0.0851	-0.1358	-0.1753	0.3300	0.0830	0.1307	
Column 10									
0.2392									
0.6309									

Los comandos para aplicar los AG, son:

```
optimtool; Solver: gamultiobj; Fitness Function : @(xpp) sim(net,xpp'); N° variables: 2; Bounds lower [100,70] upper [106,76]; Hybrid function: fgoalattain (default)
```

Luego de 132 iteraciones; se obtiene el Frente Pareto de 11 resultados:

Considerando como criterio los valores deseados de $y_1=103+-3$ e $y_2=73+-3$; y los x entre -1 y 1 los valores mejores corresponden al sexto resultado, siendo: $x_1=1.0822$; $x_2=-0.2909$; $x_3=0.4928$; $y_1=104.5412$; $y_2 = 75.9999$. Las y están dentro del rango $100+-3$ y $73+-3$; x_1 está fuera el rango -1 y $+1$; y x_2 y x_3 están dentro de los rangos. Por tanto se realiza la segunda vuelta.

Segunda vuelta; ida

Se crean los archivos de entrada y salida con los resultados del Frente Pareto anterior (11); pero en este caso, las entradas estarán en la matriz $xppp$ que vienen de la matriz ypp (anteriores salidas, 3 variables); y las salidas estarán en la matriz $yppp$ que vienen de la matriz xpp (anteriores entradas, 2 variables). Se introducen los datos al workspace y los comandos son: `>> [trainInd,vallnd, testInd]=dividetrain(15); >> net=fitnet; >> [net,tr]=train(net,xppp',yppp'); >> outputs=net(xppp'); >> errors=gsubtract(yppp',outputs); >> performance=perform(net,yppp',outputs);`

%Se obtiene: performance = 0.0020

Con los comandos vistos en la ida primera vuelta se obtiene las demás características de la RNA obtenida; que por espacio no se presentan nuevamente.

Los comandos para aplicar los AG, son:

```
>> optimtool; >> objcFcn=@(xppp) sim(net,xppp'); Fitness Function: @(xppp) sim (net, xppp'); Number of Variables: 3; Bounds Lower:[0,0,0] Upper: [1,1,1]; >> hybrid function: fgoalattain (default).
```

Luego de 149 iteraciones, se obtiene un Frente Pareto de 20 resultados:

Considerando como criterio los valores deseados de $y_1=103+-3$ e $y_2=73+-3$; y los x entre -1 y 1 los valores mejores corresponden al sexto resultado (todos los resultados de y y x salieron dentro de los rangos) siendo estos: $x_1= 0.8255$; $x_2= 0.0051$; $x_3= 0.9100$; $y_1= 103.761931$; $y_2= 72.86839$

Segunda vuelta; Retorno:

Se crean los archivos de entrada y salida con los resultados del Frente Pareto anterior (20); en este caso las entradas estarán en el archivo $xpppp$

que viene de la matriz $yppp$ (anteriores entradas, 2 variables) e $ypppp$ que viene de la matriz $xppp$ (anteriores, entradas, 3 variables). Se introducen los datos en el workspace y se aplican los siguientes comandos: `>> [trainInd,vallnd, testInd]=dividetrain(32); net=fitnet; [net,tr]=train(net,xp',yp'); outputs= net(xp'); errors= gsubtract (yp', outputs); performance=perform(net,yp',outputs)`

Se obtiene: performance = 0.000003

Con los comandos vistos en la ida primera vuelta se obtiene las demás características de la RNA obtenida; que por espacio no se presentan nuevamente.

Los comandos para aplicar los AG, son:

```
>> optimtool; >> objcFcn=@(xppp) sim(net,xppp'); Solver: gamultiobjective; Fitness Function: @(xpppp) sim (net, xpppp'); Number of Variables: 2; Bounds Lower:[102,72] Upper: [104,74]; >> hybrid function: fgoalattain (default).
```

Luego de 102 iteraciones, se obtiene un Frente Pareto de 30 resultados, de los cuales los valores dentro de los límites son: 2,3,4,6,9,11,13,14,16,19,20,22,23,24,28,29; en total 16 de un total de 30. Siendo el mejor el mejor el onceavo; siendo éstos: $x_1= 0.4021$; $x_2= 0.0187$; $x_3= 0.9545$; $y_1= 103.1701$; $y_2= 73.4071$.

Es decir, los valores óptimos encontrados para optimizar la calidad de este producto son los valores de las variables de entrada $x_1=0.4021$; $x_2= 0.0187$; $x_3= 0.9545$; estos valores permiten obtener las características de producto terminado $y_1 = 103.1701$ e $y_2= 73.4071$.

En los casos que se aplicó algoritmos genéticos multiobjetivos se probaron varias alternativas y el mejor resultado se obtuvo con Fracción de cruzamiento de 0.8, Fracción de Pareto de 0.35, con tope de 600 generaciones, un tamaño de población de 75. Los demás parámetros de Matlab R 2012b para algoritmos genéticos multiobjetivo se mantuvieron en los valores por default.

4. DISCUSIÓN

En este trabajo se propone un enfoque para la optimización de los problemas de respuesta múltiple. La novedad del enfoque es que para establecer las relaciones entre las variables se utilizan las redes neuronales RNA. Así se estima la relación de las salidas con los factores controlables y se optimiza mediante un algoritmo genético (GA), el mismo que se desarrolla con base en los resultados de las RNA. Debe tenerse presente que

el resultado obtenido de las redes neuronales es un conjunto de relaciones de pesos y funciones de transferencia, que constituyen las redes neuronales; dicho resultado se transforma a vectores y se introduce como datos para el algoritmo genético, lo cual lo realiza de manera automática el software respectivo.

Otro aspecto novedoso es que se han considerado límites tanto para las variables de entrada como para las de salida. Hecho que es muy difícil de resolver con los métodos tradicionales, para ello se realizan dos vueltas y cada vuelta con ida y retorno. En la primera vuelta de ida se utilizan como datos de entrada a las RNA las x y de salida las y ; además, en el retorno de la primera vuelta se utilizan como datos de entrada las y ; y como salida las x . Lo mismo se hace en la segunda vuelta, de ida y retorno. De esta manera se utilizan como límites tanto los que tienen las x como los que tienen las y . En el ejemplo aplicativo que se presenta se puede apreciar que los errores que dan las RNA son muy pequeños, con cuadrados medios del error menores a la diez milésima; y se obtiene un grupo de valores, los del Frente de Pareto, que cumplen con los requisitos de entrada y salida.

5. CONCLUSIÓN

La optimización de los parámetros de calidad de productos es posible utilizando redes neuronales y algoritmos genéticos; el método propuesto que contiene cinco pasos permite trabajar con muchas variables y relaciones complejas.

6. REFERENCIAS BIBLIOGRÁFICAS

- [1] Deb, K. (2009). Multi-Objective Optimization using Evolutionary Algorithms. Ed. Wiley. EE. UU.
- [2] Del Castillo, E. y Montgomery, D.C. (1993). A nonlinear programming solution to the dual response problema. Journal of Quality Technology. Vol 25, N.º 3, pp. 199-204.
- [3] Del Castillo *et al.* (1996). Modified desirability functions for multiple response optimization. Journal of Quality Technology. Vol. 28, N.º 3, pp. 337-345.
- [4] Haykin, S. (1994). Neural Networks. Ed Macmillan College Publishing Company. New York. EE. UU.
- [5] Kim, K. J. y Lin, D.K.J. Dual response surface optimization: a fuzzy modeling approach. Journal of Quality Technology. Vol 30, N.º 1, pp. 1-10.
- [6] MATLAB Optimization Toolbox. (2014). Matlab MathWorks. EE. UU.
- [7] Myers, R., Montgomery, D. (2009). Response surface methodology: process and product optimization using designed experiments. Ed. Wiley. 2a edición. EE. UU.
- [8] Montgomery, D. (2006). Diseño y Análisis de Experimentos. Ed. Wiley. 4.ª ed. México.
- [9] Nelder, J Mead, R. (1965). A simplex method for function minimization. Computation Journal. Vol. 7, pp 308-313.
- [10] Noorosonna, R. *et al.* (2008). An artificial neural network approach to multiple-response optimization. International Journal of Advance Manufacturing Technology. Vol. 40, pp. 1227-1238
- [11] Ortiz F. *et al.* (2004). A Genetic Algorithm Approach to Multiple Response Optimization. Journal of Quality Technology. Vol. 36, N.º 4, pp 432-450.
- [12] Pignatiello, Jr. J. J. (1993). Strategies fo robust multiresponse quality engineering. IIE Transactions. Vol 25, N.º 1, pp. 5-15.
- [13] Rajasekaran, S. (2003). Neural Networks, Fuzzy Logic, and Genetic Algorithms. Ed. PHI Learning. India.