

Pronóstico puntos críticos de la serie temporal “consumo de energía eléctrica del sector industrial en la ciudad de Medellín,” usando algoritmos genéticos

Héctor Tabares^{a}, Jesús Hernández^b*

^aDepartamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Antioquia. Medellín, Colombia.

^bUniversidad Nacional de Colombia. Medellín, Colombia.

(Recibido el 24 de abril de 2006. Aceptado el 29 de octubre de 2006)

Resumen

Los algoritmos genéticos (AG) están inspirados en el principio darwiniano de la evolución de las especies y en la genética. Son algoritmos probabilísticos que ofrecen un mecanismo de búsqueda paralela y adaptativa, basado en el principio de supervivencia de los más aptos y en la reproducción.

Este artículo presenta una introducción a los fundamentos de los AG. También enseña el simulador *software* AG_UdeA desarrollado con un propósito didáctico para la enseñanza de los AG. El principal aporte consiste en la aplicación de los AG para pronosticar los puntos críticos de consumo de energía eléctrica del sector industrial de la ciudad de Medellín para un período de 24 horas.

----- *Palabras clave:* algoritmos genéticos, redes neuronales artificiales.

Forecasting of critical points of the temporary series “electrical power consumption of the industrial sector in Medellín city”, using genetic algorithms

Abstract

The Genetic Algorithms (GA) are inspired by the Darwin’s principle of the evolution of the species and the genetic. They are probabilistic algorithms that offer an adaptative and parallel search mechanism, based on the principle of survival of the most capable and in the reproduction.

1 Autor de correspondencia: Teléfono: +57+4+250 57 57, fax: +57+4+263 82 82, correo electrónico: htabares@udea.edu.co. (H. Tabares).

This article presents an introduction to the foundations of the GA. Also we show the software simulator AG_UdeA developed with a didactic purpose for the teaching of the GA. The main contribution consists on the application of the GA to predict the critical points for electrical power consumption in the industrial sector in Medellín city for a period of 24 hours.

----- *Key words:* genetic algorithms, artificial neural networks.

Introducción

Al enfrentar la tarea de hallar la solución a un determinado problema, uno de los pasos fundamentales consiste en la elección de la herramienta adecuada para abordar dicha tarea; esta elección está obviamente condicionada por el tipo de problema a solucionar y por su complejidad. [1, 2]

En múltiples ocasiones se puede emplear un procedimiento explícito para hallar la solución exacta del problema, mediante una expresión matemática directa, la interpretación de alguna gráfica, la ejecución de un algoritmo iterativo o cualquier otra estrategia.

No obstante, existen problemas en los que no es viable obtener una solución exacta, porque no se conoce algún método que provea esa solución, o porque conociéndolo este resulta ser muy complejo de interpretar o muy lento en su ejecución.

La heurística es un conjunto de estrategias de búsqueda de soluciones particularmente útiles en estos casos. La idea fundamental de una heurística es buscar “buenas” soluciones al problema, aunque no se asegure que se encuentra una solución exacta. Los AG son un caso particular de estas técnicas heurísticas.

La razón por la cual los algoritmos genéticos tienen ese nombre, es porque se trata de estrategias que fueron concebidas como una imitación simplificada de la mecánica que ha gobernado los procesos genéticos de la selección natural que se dan en la naturaleza.

Por tanto, ha tomado algunos términos propios de las ciencias biológicas para describir los componentes del algoritmo (genes, cromosoma, reproducción, mutación, etc.). No hay que perder de vista, sin embargo, que los AG no pretenden modelar el proceso biológico, tan sólo se trata de una estrategia heurística de búsqueda de soluciones, inspirados en dicho proceso.

El objetivo de un AG es la optimización de una *función de evaluación* (también denominada *fitness*), sujeta a ciertas restricciones. Para ello

las variables de la función se codifican en *genes*; una colección de genes, que corresponde a un punto en el espacio de entrada de la función de evaluación, es un *individuo*. En un AG se produce una colección de individuos (una *población*) que a través de un cierto número de iteraciones va mejorando. La población de un determinado ciclo es la *generación* correspondiente a ese ciclo.

Con este artículo se presenta la aplicación de los AG para pronosticar los puntos críticos de la serie temporal “consumo de energía eléctrica del sector industrial de la ciudad de Medellín en un período de 24 horas”.

Conceptos previos

Gen

Un gen es la cantidad de información necesaria para codificar una característica específica de un organismo o estructura genética (cromosoma). [1, 2]

Cromosoma

Los AG utilizan “cromosomas”. En este caso son representaciones basadas en las características de los individuos o sucesos, a través de un conjunto de estructuras de conocimiento llamadas “genes” o “patrones”. Cada patrón o gen debe representar la presencia o la ausencia de una propiedad o característica de determinado individuo.

Los cromosomas también son llamados “cadenas de genes” o “genotipos”. La longitud de los cromosomas por lo general es fija. Cada gen puede tomar solamente un pequeño rango de valores o alelos. Frecuentemente, el adjetivo “pequeño” sólo significa “1” ó “0”, e indica la presencia o ausencia de una determinada característica del individuo o proceso que se modela.

Genoma

El genoma es todo el material genético de los cromosomas de un organismo particular. En el caso del genoma humano, tiene 3.000 millones de nucleótidos.

Medida de salud (*fitness*).

El objetivo de los AG es encontrar una estructura que ejecute bien, en términos de una medida de funcionamiento:

$$V : A \longrightarrow R +$$

Donde:

A: un conjunto dado de estructuras o cromosomas, *R*+: números reales positivos.

Entonces, la función *V* provee una medida de salud (*fitness*), para un cromosoma dado.

Cada gen de la cadena tiene un peso asociado, llamado “idoneidad del gen o patrón”, que recoge su importancia, representatividad o rendimiento en experiencias pasadas.

Entonces, dado un genotipo particular, la función *V* y los pesos de los genes proporcionan una forma para determinar la probabilidad de que una cadena sea seleccionada para contribuir a la siguiente generación elaborada a partir de la acción del AG.

Algoritmo genético (AG)

Las operaciones básicas de los AG [1, 2] son las siguientes: generación, cruce, mutación, reproducción y eliminación de patrones ineficaces (véase tabla 1).

La estrategia básica de los AG se resume en el algoritmo (*P(t)* es la población en la generación *t*).

Tabla 1 Algoritmo genético básico

Inicio principal
t = 0;
Crear P(t)
Mientras que (no se cumpla la condición de parada)
Evaluar P(t)
Seleccionar P(t + 1) a partir de P(t)
Recombinar P(t + 1)
t = t + 1
Fin ciclo
Fin principal

Crear P(t)

La generación implica crear, mediante técnicas aleatorias, la población inicial, P(0), de los individuos, utilizando un alfabeto específico por el usuario, donde cada gen adopta los valores “0” ó “1” y cada posición que ocupa en el cromosoma se asocia con un peso o medida de su importancia.

Evaluar

El desempeño de un AG suele analizarse siguiendo la función de evaluación del mejor de los individuos de cada generación, sin embargo, también puede seguirse la del peor individuo, el promedio de las funciones de evaluación de todos los individuos o su desviación estándar.

Operadores de probabilidad

El operador de probabilidad asigna una probabilidad de supervivencia a los N individuos de una cierta generación. Un individuo con una buena función de evaluación $f(i)$ deberá tener una mayor probabilidad de supervivencia que otro con una peor función de evaluación. Estos son algunos de los operadores más usuales:

- Proporcional: la probabilidad de supervivencia depende de la función de evaluación (mejor función significa mayor probabilidad).

Para maximizar

$$p(i) = \frac{f(i)}{\sum f(i)}$$

Para minimizar

$$p(i) = 1 - \frac{f(i)}{\sum f(i)}$$

- Homogénea: a todos los individuos se les asigna la misma probabilidad de supervivencia.

$$p(i) = \frac{1}{N}$$

- Ranking lineal: se ordena la población de mejor individuo a peor. La probabilidad de supervivencia depende de su posición i en el arreglo.

Seleccionar

El operador de selección establece cuántas copias se crearán de cada individuo en la siguiente generación. Se suele disponer que el número de individuos en todas las generaciones se mantenga constante. Estos son algunos de los operadores más usuales:

- Estocástica con reemplazo. Se crea una ruleta con tantas casillas como individuos tenga la población; el ángulo de cada casilla es proporcional a la probabilidad de supervivencia. Se juega a la ruleta tantas veces como individuos tenga la población y se crea una copia de cada individuo ganador.
- Estocástica universal. Se crea una ruleta, como en el caso anterior, pero sólo se juega una vez; las casillas ganadoras están igualmente espaciadas, y hay tantas como individuos tenga la población.

Recombinar

Operadores de asignación de parejas

El operador de asignación de parejas decide qué individuos se cruzan entre sí para generar nuevos individuos. Estos son algunos de los operadores más usuales:

- Aleatoria. Se buscan aleatoriamente las parejas.
- Siguiendo. La pareja de cada individuo es la siguiente en el arreglo de individuos que forma la población.
- Extremos. Al primer individuo le corresponde el último, al segundo el penúltimo, etc.

Cruce

El cruce es una operación típicamente genética donde los patrones de mayor peso, u otros aleatoriamente seleccionados, producen un intercambio de genes a partir de un punto de cruce (gen), determinado aleatoriamente.

El cruzamiento introduce constantemente nuevos cromosomas a probar, por lo que su efecto incide en otros elementos de la población.

Los operadores de cruce más usuales son:

- Cruce simple: dados dos individuos (padres) con genomas $P_1 = (C_1^1, \dots, C_L^1)$ y $P_2 = (C_1^2, \dots, C_L^2)$ respectivamente, se generan dos nuevos individuos (hijos) con genomas:

$$H_1 = C_1^1, C_2^1, C_3^1, \dots, C_i^1, C_{i+1}^2, C_{i+2}^2, \dots, C_j^2$$

$$H_2 = C_1^2, C_2^2, C_3^2, \dots, C_i^2, C_{i+1}^1, C_{i+2}^1, \dots, C_j^1$$

Donde i (denominado punto de cruce) es un número entero aleatorio del intervalo $[1, L-1]$

- Cruce doble: similar al anterior, pero con dos puntos de cruce aleatorios i, j . Los genomas de los hijos serán:

$$H_1 = C_1^1, C_2^1, \dots, C_i^1, C_{i+1}^2, \dots, C_j^2, \dots, C_{l-1}^1, C_l^1$$

$$H_2 = C_1^2, C_2^2, \dots, C_i^2, C_{i+1}^1, \dots, C_j^1, \dots, C_{l-1}^2, C_l^2$$

Mutación

El operador de mutación efectúa alteraciones en la información genética. Con esta función se busca explorar nuevas zonas del espacio de búsqueda, y así intentar escapar de posibles óptimos locales en los que pueda quedar atrapado el algoritmo. El operador de mutación más usual es el uniforme. En este caso, se toma al individuo de peores resultados (en caso de que haya más de uno, pueden aplicarse diferentes métodos de elección; el más simple es tomar el primero en la lista) y se altera el valor de un gen elegido aleatoriamente (si es un 1 cambia a 0, y viceversa).

Operadores de reproducción

El operador de reproducción establece una estrategia general de reproducción entre parejas que se cruzan. Estos son algunos de los operadores más usuales:

- Dos padres, dos hijos: por cada pareja se crean dos hijos que reemplazan a sus padres.
- Mejor padre – mejor hijo: por cada pareja se crean dos hijos. Pero sólo el mejor hijo reemplaza al peor padre.

- Mejores entre padres e hijos: por cada pareja se crean dos hijos; de los cuatro, se mantienen los dos mejores.

Implementación software algoritmo genético

Con el objeto de validar el funcionamiento de los AG, realizar experimentos numéricos y pruebas de ensayo, se desarrolló el programa de computadora AG_UdeA.exe. La implementación software se encuentra en la siguiente dirección electrónica:

<http://ingenieria.udea.edu.co/portal>

Ruta de acceso:

- Páginas web académicas.
- Temas de apoyo a cursos.

Héctor Tabares O.

- Algoritmos genéticos (AG_UdeA.exe).

Serie temporal consumo de energía eléctrica

La función de consumo de energía eléctrica del sector industrial de la ciudad de Medellín en un periodo de 24 horas es como se ilustra en la tabla 2.

Tabla 2 Serie histórica. Demanda de energía del sector industrial en un periodo de 24 horas

<i>Demanda de energía del sector industrial en un periodo de 24 horas</i>													
Hora	0	2	4	6	8	10	12	14	16	18	20	22	24
kW	5000	4800	4800	4800	9000	10000	9500	9000	7500	7800	8000	6500	5000

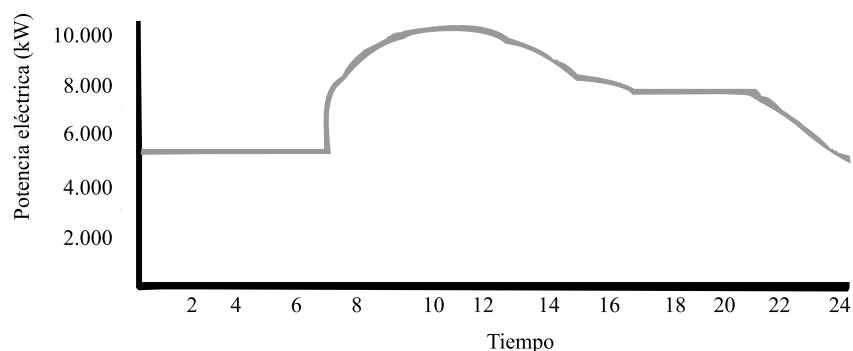
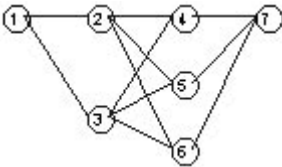


Figura 1 Serie histórica


Una aproximación a la serie histórica de datos se obtiene utilizando algoritmos de interpolación de Newton o una Red Neuronal Artificial (RNA) del tipo PERCEPTRON MULTICAPA (PMC) [3, 4, 5]. Se eligió la segunda opción pues ésta tiene mayor

habilidad para generalizar casos a partir de entradas desconocidas. La técnica de entrenamiento fue Regla Delta Generalizada y el error final obtenido fue de 0,01. La topología, sin neuronas adaptativas, y pesos finales del PMC se muestran en la tabla 3.

Tabla 3 Topología y pesos de la RNA

Topología			Pesos																																												
			<table border="1"> <thead> <tr> <th></th> <th>Wij</th> <th>Peso Inicial</th> <th></th> <th>Wij</th> <th>Peso Inicial</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>w1 2</td> <td>-1.6644</td> <td>7</td> <td>w2 6</td> <td>10.1487</td> </tr> <tr> <td>2</td> <td>w1 3</td> <td>4.3715</td> <td>8</td> <td>w3 6</td> <td>-0.5788</td> </tr> <tr> <td>3</td> <td>w2 4</td> <td>6.4382</td> <td>9</td> <td>w4 7</td> <td>-9.4774</td> </tr> <tr> <td>4</td> <td>w3 4</td> <td>-21.0937</td> <td>10</td> <td>w5 7</td> <td>-7.7576</td> </tr> <tr> <td>5</td> <td>w2 5</td> <td>-1.8495</td> <td>11</td> <td>w6 7</td> <td>8.6870</td> </tr> <tr> <td>6</td> <td>w3 5</td> <td>5.8113</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>				Wij	Peso Inicial		Wij	Peso Inicial	1	w1 2	-1.6644	7	w2 6	10.1487	2	w1 3	4.3715	8	w3 6	-0.5788	3	w2 4	6.4382	9	w4 7	-9.4774	4	w3 4	-21.0937	10	w5 7	-7.7576	5	w2 5	-1.8495	11	w6 7	8.6870	6	w3 5	5.8113			
	Wij	Peso Inicial		Wij	Peso Inicial																																										
1	w1 2	-1.6644	7	w2 6	10.1487																																										
2	w1 3	4.3715	8	w3 6	-0.5788																																										
3	w2 4	6.4382	9	w4 7	-9.4774																																										
4	w3 4	-21.0937	10	w5 7	-7.7576																																										
5	w2 5	-1.8495	11	w6 7	8.6870																																										
6	w3 5	5.8113																																													

El pronóstico de los puntos críticos de la serie temporal usando el programa de computadora AG_UdeA es como se ilustra a continuación:

Primero. Presione el botón “Lectura Tuplas”  de la interfaz del sistema. Llene la forma como se ilustra en la figura 2.

Los individuos seleccionados fueron los mismos que aparecen en el histograma de consumo de

energía del sector industrial. Estos son los puntos críticos de la curva.


Segundo. Presione el botón “Validar RNA”  de la interfaz del sistema. Presione el botón “Calcular” del conjunto “Validar Grupo de Valores”. El sistema muestra la curva de aproximación obtenida con un PMC a la serie histórica de datos (véase figura 3).



Figura 2 Lectura datos de entrenamiento

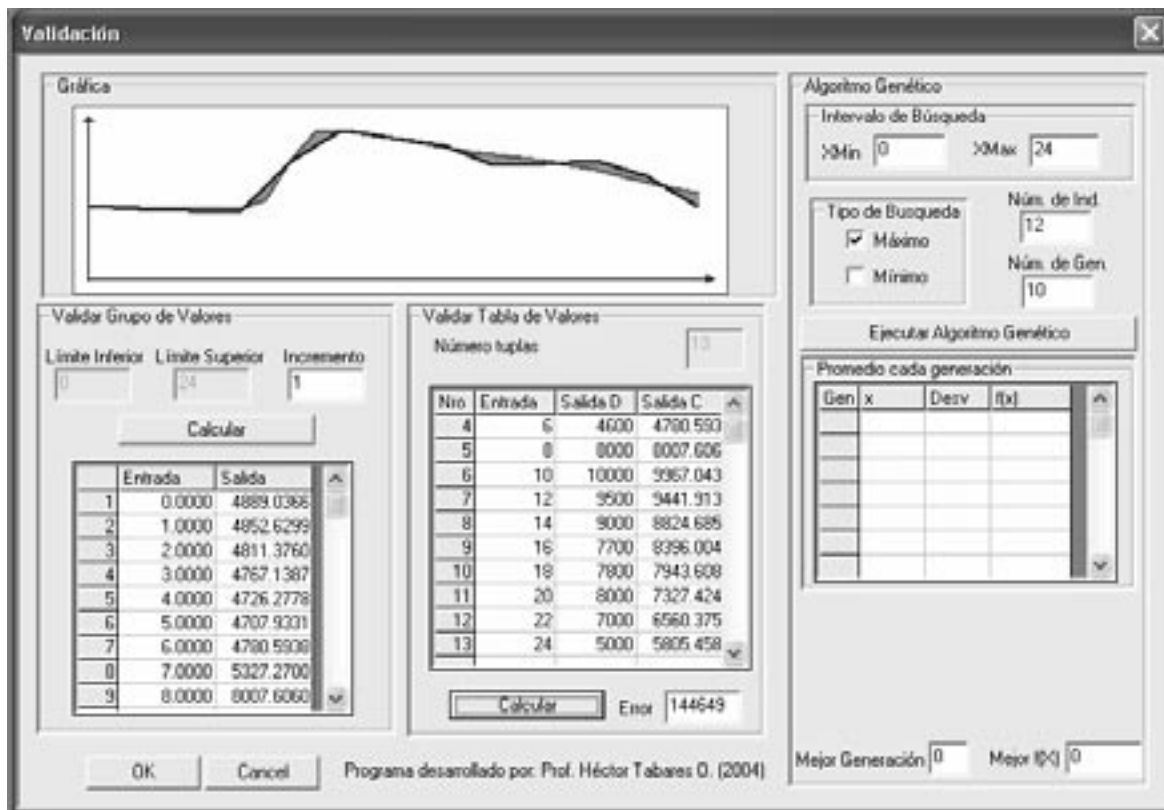


Figura 3 Validación RNA

Tercero. Usando AG se pronosticará la hora de mayor consumo de energía eléctrica. Para ello se emplearán 13 individuos generados aleatoriamente por medio de la función *random* incorporada en el lenguaje de programación VC++, como población inicial entre un rango de [0, 24]. Los operadores utilizados son:

- Probabilidad: proporcional.
- Selección: estocástica con reemplazo.
- Asignación de pareja: siguiente.
- Mutación: uniforme.
- Reproducción: cruce simple.

Siga las siguientes instrucciones sobre la interfaz del sistema (véase figura 4).

- Presione el botón “*Máximo*” del conjunto “*Tipo de Búsqueda*”
- Presione el botón “*Ejecutar Algoritmo Genético*”.

El sistema muestra gráfica y tubularmente en el conjunto “*Promedio cada generación*” la solución del problema con AG. En la décima (10) generación el sistema informático encontró que a las 9.92 horas (9:55 a. m.) hubo el mayor consumo de energía con 9.973,27 kW. Estos valores son aproximaciones a los registros reales: a las 10:00 a. m. se presentó la máxima demanda de energía con un valor de 10.000 kW. El error fue de 8,3 y 0,26% respectivamente. Los resultados obtenidos muestran en general un buen desem-

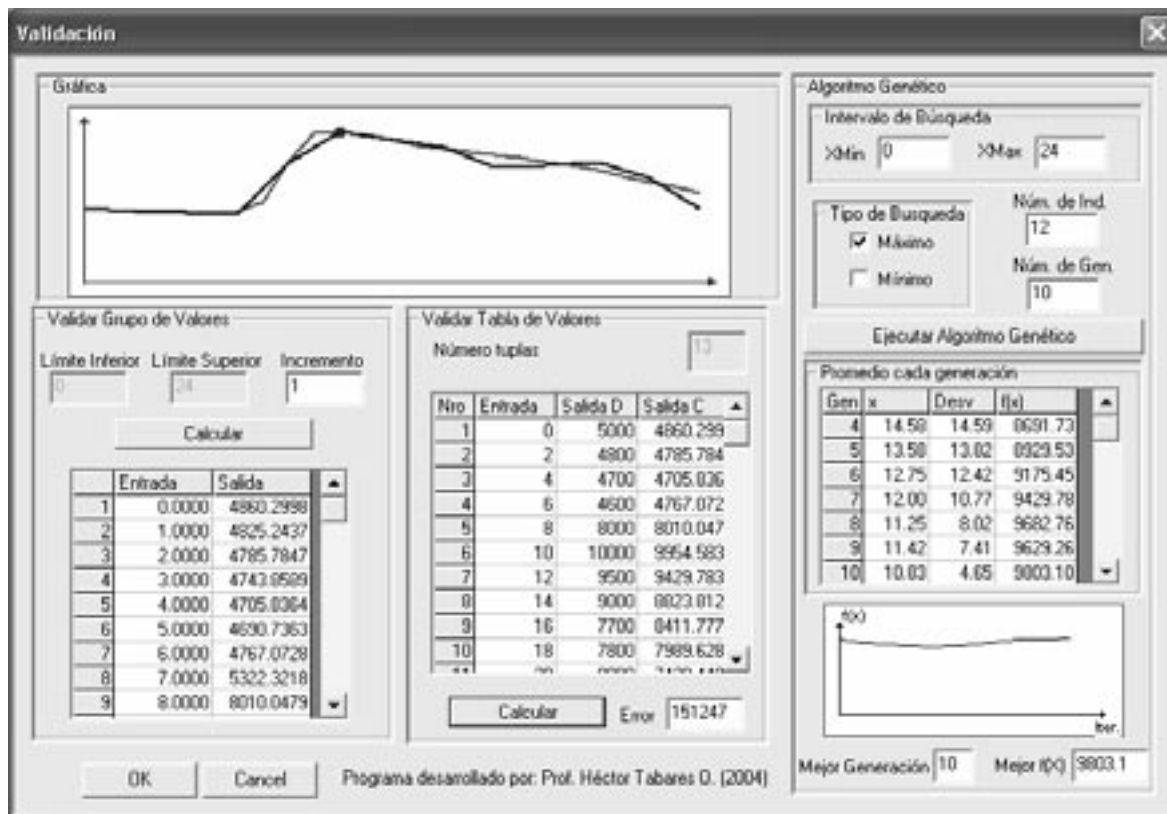


Figura 4 Ejecución AG. Punto de máxima

peño del AG, proporcionando resultados acordes con el dominio del problema.

Es importante comentar que la detección del punto crítico en la primera generación obedece a que la función *random* eligió un superindividuo que dominó la convergencia del algoritmo en el resto de las iteraciones.

Cuarto. Pronóstico hora de menor consumo de energía eléctrica. Se utilizará el mismo número de individuos, de generaciones y de operadores del caso anterior (véase figura 5).

- Presione el botón “*Tipo de Búsqueda*” mínimo.
- Seguidamente presione el botón “*Ejecutar Algoritmo Genético*”.

Estudiando el histograma “*Promedio cada generación*”, se observa que la generación número seis (6) encontró el menor consumo de 4.708,8 kW a las 5.08 horas (5:04 a. m.). Los registros reales muestran que el menor consumo fue de 4.600 kW ocurrido a las a las 6:00 a. m. Los errores en la aproximación fueron 2,3 y 16% respectivamente.

Ventajas y limitaciones de los AG

Algunas ventajas que se obtienen al utilizar este tipo de algoritmos son:

- Se requiere de muy poca información para resolver un problema.
- Puede optimizar funciones con una gran cantidad de variables, multimodales, discontinuas o con ruido.

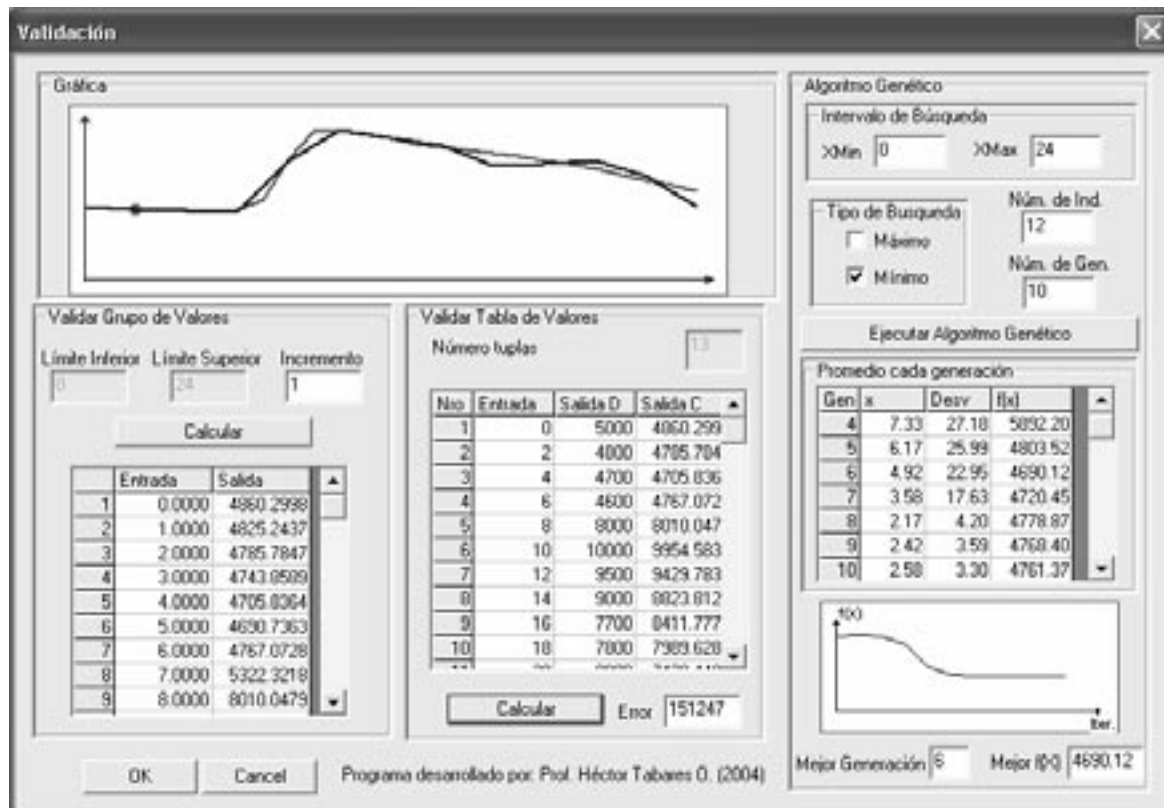


Figura 5 Ejecutar AG. Punto de mínima

Aunque puede parecer una magnífica solución para muchos problemas, tiene grandes desventajas:

- Un problema que enfrenta los AG es el de la permutación que hace la evolución menos eficiente, y también dificulta a los operadores de recombinación la producción de hijos con alta aptitud. La mutación es el método empleado para superar la permutación. Este operador induce a explorar la totalidad del espacio de búsqueda y permite mantener la diversidad genética en la población, para que el AG sea capaz de encontrar soluciones entre todas las permutaciones posibles de la red.
- Su espacio de búsqueda (i. e., sus posibles soluciones) debe estar delimitado dentro de un cierto rango. Este punto es muy importante, y lo más recomendable es intentar resolver problemas que tengan espacios de búsqueda discretos —aunque éstos sean muy grandes—. Sin embargo, también podrá intentarse usar la técnica con espacios de búsqueda continuos, pero preferentemente cuando exista un rango de soluciones relativamente pequeño.
- Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en cierta medida de los parámetros que se utilicen —tamaño de la población, número de generaciones, etc.
- Pueden converger prematuramente debido a una serie de problemas de diversa índole.
- Debe poderse definir una función de aptitud que indique qué tan buena o mala es una cierta respuesta.

- En la práctica los resultados finales que arrojan los AG son difíciles de seguir y probar, encontrándose con que, en ocasiones los AG resuelven satisfactoriamente un problema de optimización dado y otras se quedan muy alejadas del óptimo. Los estudiosos del tema han tratado de caracterizar lo que han denominado problemas AG-fáciles (aquellos en los que los AG proporcionan buenos resultados) y AG-difíciles con el objetivo de saber de antemano, al estudiar un nuevo problema, si los AG son una buena elección para su resolución.

Conclusiones

Como lo indican los experimentos numéricos presentados en este artículo, se encontró que los AG pronostican de manera aproximada los puntos críticos de la curva “Consumo de energía eléctrica del sector industrial en la ciudad de Medellín”. El punto de máxima, tiempo y potencia consumida, guardan una error de 1,96% con respecto a los valores reales.

Los puntos de mínima se obtuvieron con un error de 1,95% respectivamente.

En este caso, la operación exitosa del AG se debe a que se identificó claramente el espacio de búsqueda, (0, 24) y el número de individuos que conformarán cada generación.

Referencias

1. T. Masters. *Practical Neural Networks recipes in C++*. San Diego, CA. EE. UU. Ed. Academic Press, Inc. 1993. pp. 21-494.
2. M. Melanie. *An Introduction to Genetic Algorithms*. Cambridge, Massachusetts, MIT Press. 1996. p. 10.
3. J. Hilera, *Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones*. Madrid, España. Alfaomega, 2000. pp. 22-408.
4. B. Martín del Brio. *Redes neuronales y sistemas difusos*. México D. F., México. AlfaOmega, 2002. pp. 20-312.
5. X. Yao. “*Evolving Artificial Neural Networks. School of Computer Science*”. Proceedings IEEE, septiembre de 1999. B15 2TT.