OPTIMIZADOR HÍBRIDO BASADO EN ALGORITMOS GENÉTICOS Y GRADIENTE

Carlos Duque Bregón, Enrique Fernández del Río

Departamento de Electrónica y Teoría de Circuitos
Universidad Europea de Madrid

cdjunior@cas.es, jenrique.fernandez@tel.uem.es

ABSTRACT

In this communication a commercial software to solve optimization problems is presented. The program implements a hybrid optimizer based on genetic algorithms and the gradient method. The application has been developed to RYMSA Co., where they are using it for the optimization of a number of electromagnetic problems. It is important to point out that this software has a broader range of application, in the sense that it can be used to solve any kind of parametric optimization problem.

1. INTRODUCCIÓN

Esta aplicación ha sido concebida debido a la necesidad de optimizar distintos procesos y productos que se desarrollan en los departamentos de I+D de ingeniería. Frecuentemente el ingeniero de diseño dispone de sus propios programas de análisis (ficheros ejecutables) pero, también con mucha frecuencia, cuando se pasa a la fase de optimización, ésta se realiza de una forma manual o semi-automática. En concreto, es habitual lanzar el/los ejecutable/s por lotes (modo *batch*), esperar los resultados de ejecución, analizarlos y volver a reconducir la nueva ejecución. La aplicación que se describe en este trabajo pretende automatizar el proceso de optimización. La idea consiste en que sea la aplicación (el optimizador) quien se encargue de reconducir la optimización de forma automática e inteligente.

La técnica elegida para la optimización principal se basa en los algoritmos genéticos (AG) [1-2]. La razón de esta elección se fundamenta en que los espacios de búsqueda en problemas de ingeniería son, en general, muy grandes, pudiendo, además, aparecer múltiples máximos o mínimos (extremos) locales. En este contexto los métodos deterministas son menos eficientes. La eficiencia de los AG reside en su modo de operación pues trabajan con criterios estocásticos (derivados de la propia selección natural de las especies) combinados con toma de decisiones deterministas, de forma que resulta más fácil evitar caer en extremos locales. El otro método implementado actúa seguidamente al AG y está basado en el gradiente [3]. Este método lo que hace es afinar aún más el extremo obtenido por los AG pues estos últimos trabajan con las variables codificadas en lugar de con las variables en sí, como ocurre en el gradiente.

En el apartado 2 se detalla el optimizador desarrollado, mientras que en el apartado 3 se muestran algunos resultados obtenidos con esta aplicación en el campo de los arrays de antenas. Finalmente se resumen las principales conclusiones derivadas de este trabajo.

2. DESCRIPCIÓN DE LA APLICACIÓN

La aplicación es de muy fácil uso y es totalmente configurable. Los datos que se van obteniendo durante el proceso de optimización se van mostrando en pantalla en tiempo real y, además, se van almacenando en formato excel para su posterior tratamiento.

La aplicación consiste en seis carpetas: ficheros, operadores genéticos (ver fig. 1), recentrado, variables, gradiente y resultados.

Fig. 1 Carpeta para la configuración de los operadores genéticos

El intercambio de datos entre la función a optimizar y el optimizador se realiza a través de ficheros. Éstos se configuran en la sección de ficheros. También está implementada la opción de comenzar a optimizar desde una optimización anterior. Debido a lo tedioso, aunque sencillo, que puede llegar a ser la configuración de los intervalos de búsqueda de las variables, y demás campos, se ha programado también la opción consistente en que se pueda cargar una configuración previamente existente en formato excel.

Dentro de la carpeta operadores, fig. 1, configuramos los distintos operadores genéticos que se han programado, selección, cruce y mutación, y sus variantes.
Debido a la posibilidad que existe de que el máximo absoluto se encuentre fuera de los intervalos de búsqueda que se hayan configurado para cada variable, en vez de tener que esperar a que termine el programa para reconfigurar los nuevos intervalos de forma manual, el optimizador del presente trabajo se reconfigura automáticamente (recentrado) para que el espacio de búsqueda cambie en dirección al máximo. Además, en esta carpeta se configura la reinicialización automática del optimizador en el caso en que éste se haya congelado, es decir que todos los individuos se parezcan al mejor de ellos en un porcentaje a configurar.
En la carpeta gradiente se especifica la condición de cambio de método de optimización (de AG se pasa a optimizar con el gradiente) y se configura el gradiente. La función del gradiente es afinar el máximo obtenido por los AG pues puede que dicho máximo se encuentre entre dos escalones de cuantificación.
En la carpeta variables se configura el espacio de búsqueda de cada una de las variables de la función así como el número de generaciones y de individuos del AG. Las variables pueden recibir un nombre indicativo de lo que son y se configuran de forma sencilla mediante la ventana que aparece en la fig. 2.

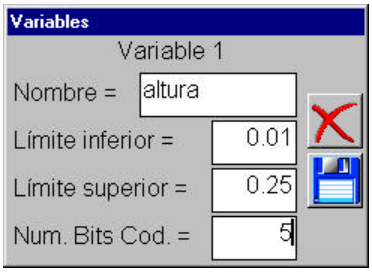

Fig. 2 Ventana para la configuración de una variable

Debido al gran tiempo de cálculo que requiere la evaluación de algunas funciones, y a la posibilidad de que durante una misma optimización aparezca varias veces la función evaluada en un mismo punto, la aplicación se ha programado de tal forma que no se evalúa dos veces una misma función en un punto que haya sido evaluada en las diez últimas generaciones. El número diez ha sido elegido en compromiso entre la memoria utilizada por el programa y el número de evaluaciones que se evitan.

3. EJEMPLO DE APLICACIÓN

En este apartado se ilustra la utilidad de la aplicación por medio de un ejemplo perteneciente al ámbito de la antenas. Se trata de decidir qué elementos de un array lineal, de $2N$=200 elementos equiespaciados $\lambda/2$, se alimentan y cuáles no, con el objetivo de conseguir la mayor relación lóbulo principal a lóbulos laterales.
La función de evaluación empleada viene dada en al ec. (1), donde u=cos(f), a_n = 1 ó 0, y $2N_{on}$ es el numero de elementos alimentados en total. El optimizador realizó 3000 evaluaciones de la función (valor realmente pequeño en comparación con el espacio de búsqueda: $2^{100} \cong 10^{30}$) y se consiguió obtener que el máximo de los lóbulos secundarios estuviera a 23.92 dB por debajo del lóbulo principal, como se aprecia en la figura 3, y un llenado del 73 % (N_{on} =73).

$$FF(u) = \left| 2 * \sum_{n=1}^{N} a_n \frac{\cos\left(2p\left(nd - \frac{d}{2}\right)u\right)}{N_{on}} * sen(f) \right| \qquad (1)$$

La configuración del array sería:

1111111111111111111111111111111111111111011111111101
0110101111110110101010100101001001001000110100100 1

Dónde el primer 1 de la primera fila es el primer elemento situado a la derecha del centro del array, y el último 1 de la segunda fila es el último elemento de la parte derecha del array.

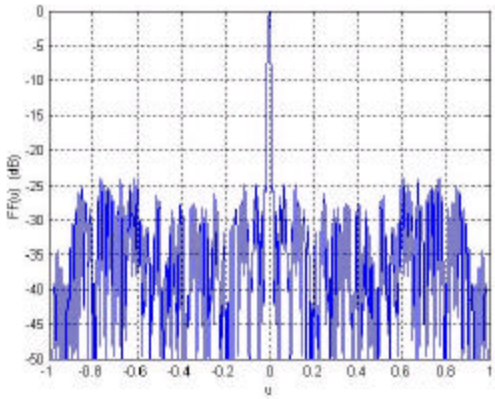

Fig. 3 Campo lejano de un *thinned array* de 200 fuentes con un diagrama de cada elemento del tipo |senϕ|

4. CONCLUSIONES

En este trabajo se describe un optimizador basado en una técnica combinada algoritmos genéticos-gradiente que se ha desarrollado como una aplicación en entorno Windows. El usuario de la aplicación ha de disponer de un ejecutable (o conjunto de ejecutables) que simule cualquier problema de análisis paramétrico y, obviamente, ha de definir, a partir de los resultados del análisis, su función objetivo (función a optimizar). La aplicación, una vez configurada, llamará al ejecutable de análisis, éste devolverá a la aplicación el valor de la función objetivo, la aplicación decidirá en qué nuevo punto evaluar la función objetivo, y el proceso se reiterará hasta encontrar los valores de los parámetros que optimizan el problema en cuestión.

5. REFERENCIAS

[1] Goldberg, D. E., *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wesley, 1989.
[2] Rahmat-Samii Y. y Michielssen E., *Electromagnetic Optimization by Genetic Algorithms* Wiley Series in Microwave and Optical Engineering, 1999.
[3] Fletcher R. *Practical Methods of optimization.* John Wiley & Sons Ltd., 1987.