TITULO: Una Solución a los Problemas de Experimentación, Modelación y Optimización de Sistemas Vinculados a la Administración, (Parte V). Implementación del método de optimización Complex.

TITLE: A Solution to the Experimentation, Modelation and Optimization Problems of Systems Associated to Management, (Part V). Implementation of the Complex Optimization Method.

AUTORES:

Lic. Horacio Salvador Hernández.
MSc. Emilio Quevedo Ríos.

PAÍS: Cuba

RESUMEN:

Se expone la implementación del método de optimización Complex de Boxl, preparado para trabajar con variables discretas y continuas a la vez, como parte de las herramientas de optimización del paquete de software para el diseño, modelación y optimización, ESTADSOL.

PALABRAS CLAVES:

OPTIMIZACIÓN, MODELACIÓN, DISEÑO DE EXPERIMENTOS

ABSTRACT:

This paper reports the implementation of the Box´s Complex optimization method, adapted to work at the same time with discreet and continuous variables, like part of the optimization tools of the software package for the design, modelling and optimization, ESTADSOL.

KEY WORDS:

OPTIMIZATION, MODELLING, EXPERIMENTAL DESIGN

INTRODUCCIÓN:

El Método Complex de Box (MCB) es un procedimiento secuencial con el cual no se necesita conocer la dirección del vector gradiente para alcanzar un óptimo local de la función objetivo en una región convexa [1]. Se desarrolló con el objetivo de mejorar los problemas que surgen en la construcción de la figura inicial en el Método Simplex [2]. Al igual que el Método Simplex, el MCB se basa en la generación y desplazamiento de una figura de k vértices en un espacio $\mathbf{R}^n$ (complex), donde **n** es el número de variables explicativas del modelo. El método exige que el número de vértices sea mayor que el número de variables ($k \geq n+1$). La formación del complex inicial se ejecuta del siguiente modo:

1. Se fija el primer vértice de la figura en un punto P fiable, o sea que satisface todas las restricciones del modelo g(x) ≤ 0 [3].

2. Se generan los restantes k - 1 vértices en base a las relaciones:

$$x_{ij} = x_i^{min} + r_{ij}(x_i^{max} - x_i^{min}) \quad \text{para } i = \overline{1;\ n} \ \text{ y } \ j = \overline{2;\ k} \qquad (1)$$

Donde r_{ij} son números aleatorios entre 0 y 1. Los puntos así obtenidos estarán dentro de la región rectangular de búsqueda.

3. Si algunos de los vértices no cumple alguna de las restricciones implícitas se mueve su posición a la mitad de la distancia hasta el centroide de los restantes vértices fiables. Las nuevas coordenadas del punto serán:

$$x_{iN'} = \tfrac{1}{2} \cdot (x_{iN} + x_{iC}) \qquad i = \overline{1; n} \qquad (2)$$

El uso de esta figura de tamaño variable evita la dificultad de construir un simplex regular que satisfaga todas las restricciones y que sea de tamaño razonable.

El movimiento del complex hacia el óptimo, segunda etapa del algoritmo, sigue los siguientes pasos:

1. Se evalúa la función objetivo en cada uno de los vértices del complex, seleccionando la mejor y peor respuesta y_B y y_W, en los vértices P_B y P_W respectivamente. Se comprueba el cumplimiento del criterio de convergencia:

$$|y_B - y_W| \leq D \qquad (3)$$

Si se satisface la condición de terminación, se concluye la optimización, reportándose a y_B como el valor óptimo de la respuesta obtenido en el punto $P_{B(xB)}$.

2. Se rechaza el punto P_W, sustituyéndolo por un nuevo punto $P_{N(xN)}$, obtenido por reflexión de P_W con respecto al centroide $P_{C(xC)}$ de los restantes puntos del complex. P_N se halla a una distancia α (α ≥ 1) veces más lejos del centroide P_C que el punto rechazado, pero en la dirección definida por el vector con origen en el punto rechazado y que pasa por el centroide. Las coordenadas del nuevo vértice están dadas por la siguiente expresión:

$$x_{iN} = \alpha(x_{iC} - x_{iW}) + x_{iC} = (1+\alpha)x_{iC} - \alpha x_{iW} \qquad (4)$$

Las coordenadas del centroide están dadas por:

$$x_{iC} = \frac{1}{k-1}\left(\sum_{j=1}^{k} x_{ij} - x_{iW}\right) \quad (5)$$

3. Si el nuevo punto posee el peor valor del conjunto de vértices en la nueva figura, su localización se mueve a la mitad de la distancia en dirección al centroide (expresión 2).

4. Si en este acercamiento no se encuentra un punto adecuado, se aleja el punto nuevo del centroide por fuera de su posición inicial en un intento de encontrar una zona más favorable.

5. Si el punto generado no satisface una de las restricciones explícitas para xi, esa variable se lleva a una pequeña distancia d dentro de los límites establecidos para obtener un punto factible.

$$x_i = x_i^{\min} + \delta \quad \text{o} \quad x_i = x_i^{\max} - \delta \quad (6)$$

Se toma una u otra expresión en dependencia de si la frontera que se cruza es la correspondiente al límite inferior o superior.

6. Adicionalmente, si alguna restricción **g(x) ≤** 0 es violada, el punto es movido a la mitad de la distancia en dirección al centroide según la expresión 2. Si aún así la restricción sigue siendo violada, esta retracción se continúa hasta tanto no se halle un punto factible.

La búsqueda del óptimo continúa con repetidos rechazos y generaciones de vértices hasta que el complex se reduce esencialmente al centroide y se cumple la condición de convergencia (expresión 3), o el número de iteraciones supera un limite preestablecido.

Este proceso es básicamente el que se aplica cuando las n variables son continuas [4]. Si en el problema hay algunas variables discretas o todas son de es tipo, cada una definida sobre un conjunto $\mathbf{U_i}$, entonces, la función $\mathbf{y = f(x)}$ queda definida sobre los nodos determinados por el producto cartesiano $\mathbf{U = U_1 \times U_2 \times U_3 \times \ldots \times U_c}$ [4]. Para estos casos, al generarse un nuevo punto se tiene en cuenta que las coordenadas, correspondientes a variables discretas, tomen los valores adecuados para los cuales están definidas dichas variables.

El objetivo de este trabajo es hacer un reporte acerca del procedimiento que se siguió para programar el MCB como parte de las herramientas de optimización del software para el diseño de experimentos, modelación y optimización, ESTADSOL [5].

RESULTADOS DEL TRABAJO:

El Método Complex de Box se programó en el módulo Complex.bas. El método trabaja sobre un objeto de la clase **TModelo** [6] previamente compilado,

contándose con variables configuradas, función objetivo, derivadas parciales y restricciones.

Variables del módulo Complex.bas

El método hace uso de las variables globales de la optimización para el número máximo de iteraciones *OptNumItera*, exactitud *OptExactitud*, tipo de extremo buscado *OptExtremo* y punto inicial de la búsqueda *OptPtoIni*. Además se utilizan las variables específicas siguientes:

Public CplxLargo As Single: se usa para acomodar las coordenadas correspondientes a variables ilimitadas, en el complex inicial.

Public CplxCtetExp As Single: constante de expansión.

Public CplxConverg As Byte: indica si la convergencia se establece por el número de repeticiones del mejor punto o por la condición 3.

Public CplxNumRepMejor As Byte: número de repeticiones del mejor punto que provoca un cambio en la ejecución del algoritmo.

Private Complex() As TPunto: arreglo con los puntos que componen el complex.

P*rivate Centroide As TPunto:* contiene el punto centroide.

Private PtoMejor As Integer: índice del punto con la mejor respuesta en el arreglo de puntos del complex.

Private PtoPeor As Integer: índice del punto con la peor respuesta en el arreglo de puntos del complex.

Private ContMejoresIguales As Integer: contador del número de veces que se repite el mejor punto.

Private ContItera As Integer: contador de iteraciones.

Private NCplx As Integer: número de puntos en el complex.

Private NCoord As Integer: número dimensiones del espacio de búsqueda (número de variables independientes del modelo).

Procedimientos y funciones del módulo Complex.bas

Al programar el Método Complex de Box se consideró primero la generación del complex inicial y después el movimiento de éste hacia el óptimo, para ello se confeccionaron básicamente los siguientes subprogramas o funciones:

Private Sub CplxInicializa(): procedimiento de inicialización. En él se comprueba que el modelo esté debidamente compilado y no contenga errores.

Se verifica que el punto de partida de la optimización satisfaga las restricciones establecidas. Se inicializan las variables *NCplx*, *ContItera* y *ContMejoresIguales*. Las variables globales relacionadas con el Método Complex, se inicializan con anterioridad.

Private Sub CalcCentroide(Np As Integer, R As Integer): calcula el centroide de los puntos del complex según la expresión 5. En el parámetro Np se pasa el número de vértices a considerar. La inclusión de este parámetro permite, durante la formación del primer complex, calcular el centroide de los Np puntos fiables ya establecidos. Si en Np se pasa un valor nulo o superior al número máximo de puntos en el complex, en el cálculo se consideran todos los vértices. El parámetro R lleva el índice del punto que se rechaza.

Private Sub MueveAMitad(I As Integer): mueve el vértice del complex con índice I, a la mitad de la distancia que lo separa del centroide, utilizando la expresión 2.

Private Function CplxMejor(): devuelve el índice del vértice del complex que posee la mejor respuesta.

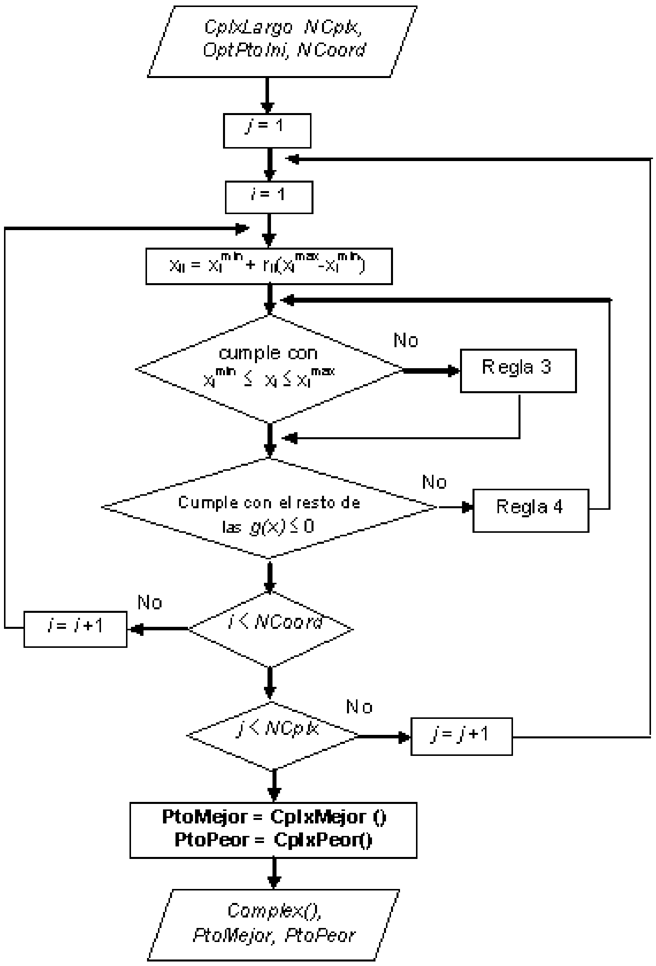

Figura 1. Diagrama en bloques del procedimiento *GeneraCplxIni*.

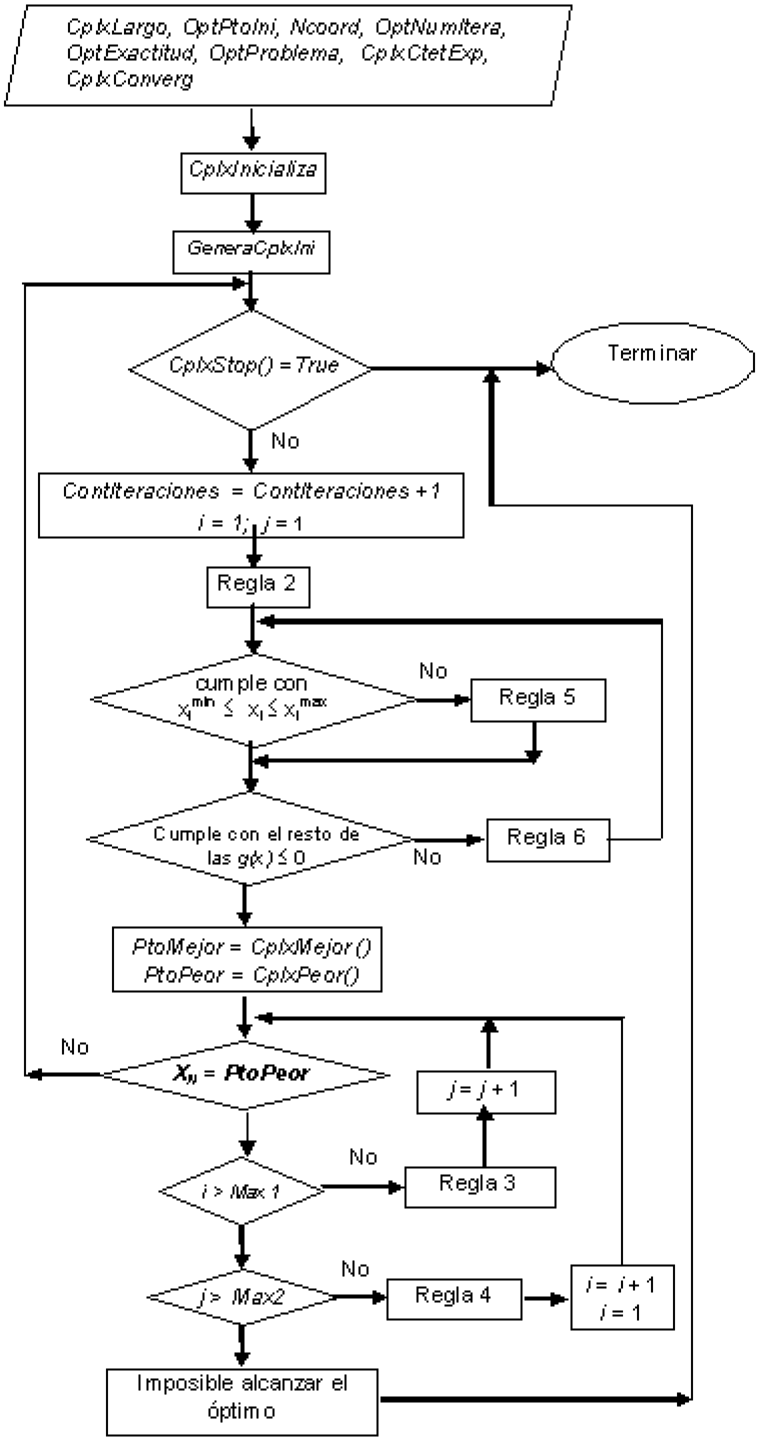

Figura 2. Diagrama en bloques del procedimiento *ComplexMetodo*

Private Function CplxPeor(): devuelve el índice del peor punto del complex.

Private Function CplxStop() As Boolean: devuelve True si se satisface el criterio de parada, que puede ser uno de los tres siguientes: si se alcanza el número de iteraciones máximo prefijado, si se cumple la condición de convergencia (expresión 3), o si el mejor punto (PtoMejor) no cambia en un número determinado de iteraciones.

Private Sub GeneraCplxIni: genera los vértices de la figura inicial según la fórmula 1. Este es uno de los procedimientos fundamentales del algoritmo y su diagrama en bloques se muestra en la figura 1.

Public Sub ComplexMetodo(): procedimiento principal del Método Complex. Controla la ejecución del método, haciendo llamadas a los procedimientos y funciones subordinados. La generación de un nuevo punto (regla2) se hace mediante la expresión 4. La corrección a la violación de restricciones explícitas (regla 5) se realiza según la expresión 6. Para el acercamiento al centroide cuando se incumplen las restricciones implícitas (regla 6) se utiliza la expresión 2. El diagrama en bloques del procedimiento se muestra en la figura 2.

CONCLUSIONES:

Este método se distingue por su efectividad en la localización de los extremos locales de funciones de varias variables, permitiendo combinaciones de variables discretas y continuas, definidas en espacios restringidos. El software presentado incluye, junto a otros métodos, el Complex, facilitando la labor del experimentador en la obtención de modelos de los sistemas en estudio y su posterior optimización.

El método progresa más rápidamente cuando los puntos $\mathbf{x}_j$ están lejos del óptimo, pero la velocidad de convergencia no parece ser particularmente sensible a los cambios de α y k. Box recomienda que a debe estar alrededor de 1,3 y k aproximadamente igual a 2n (aunque se usan menos si n es mayor que, digamos, 5) [2]. Si se usa un número insuficiente de vértices, tal como k = n +1, el complex tiende a colapsar y aplanarse a lo largo de la primera restricción encontrada.

BIBLIOGRAFÍA:

1. Box, George E. P. Empirical model building and response surfaces / George E. P. Box, Norman R. Draper. _ _ 2. ed. _ _U. S. A. : John Wiley, 1992. _ _669 p.

2. Box, M. G. A new method of constrained optimization and a Comparison with other methods. Computer Journal (US) 8: 42-52, 1965.

3. Salvador Hernández, Horacio; Emilio Quevedo Ríos. Una solución a los problemas de experimentación, modelación y optimización de sistemas vinculados a la administración, (Parte III). Implementación de la optimización por el Método del Gradiente. Ciencias Holguín (Cuba), 2002.

4. Box , George E. P. Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building / George E. P. Box, W. G. Hunter, and J. S. Hunter. _ _ U.S.A.: Wiley, 1978. _ _524 p.

5. Salvador Hernández, Horacio; Emilio Quevedo Ríos. Una solución a los problemas de experimentación, modelación y optimización de sistemas vinculados a la administración, (Parte I). Ciencias Holguín (Cuba), 2002.

6. Salvador Hernández, Horacio; Emilio Quevedo Ríos. Una solución a los problemas de experimentación, modelación y optimización de sistemas vinculados a la administración, (Parte II). Implementación de un compilador y evaluador de expresiones matemáticas. Ciencias Holguín (Cuba), 2002.

DATOS DE LOS AUTORES:

Nombre:

Lic. Horacio Salvador Hernández. Profesor Asistente
MSc. Emilio Quevedo Ríos. Profesor Auxiliar

Centro de trabajo:

Departamento de Física – Química de la Universidad de Holguín. Ave. XX Aniversario s/n. Piedra Blanca. Holguín 80400. Gaveta Postal 57.