

UN ALGORITMO DE OPTIMIZACION GLOBAL BASADO EN EL MODELO PROBABILISTICO GEOMETRICO. APLICACION A UN PROBLEMA DE CLASIFICACION

M. Sánchez García (*) y C. González Martín(&)

(*)Universidad Complutense de Madrid

(&)Universidad de La Laguna

Abstract

In this paper we introduce a stochastic method for global optimization and, for that, we use ideas which are based in the geometric probabilistic model. The algorithm is applied to a matricial classification problem.

Keywords: Global Optimization, Local Search, Classification Problems, Geometric Probabilistic Model.

A.M.S. Subject Classification: 90C15, 90C99

1.-INTRODUCCION

Supongamos que $f: \mathbb{R}^m \rightarrow \mathbb{R}$ es una función que se desea optimizar sobre $S \subset \mathbb{R}^m$, $S \neq \emptyset$. Una expresión de dicha optimalidad se podría concretar en determinar óptimos locales de f en S . De ello, precisamente, se ocupan gran cantidad de procedimientos que aparecen en la literatura sobre el tema (ver, por ejemplo, Bazaraa y Shetty (1979), McCormick (1983), Luemberger (1984),...). Sin embargo, en muchos casos, se precisa contar con algoritmos que, además, distingan entre los óptimos locales y tengan la capacidad de localizar el "mejor" de ellos. El problema de diseñar tales algoritmos es conocido como Problema de Optimización Global (POG).

Los problemas de optimización global tienen importancia no solo porque pretenden mantener la coherencia de búsqueda del óptimo absoluto que impone "a priori" todo problema de optimización. La determinación del óptimo global es imprescindible en multitud de problemas económicos, de diseño ingenieril, de estimación estadística, etc. Sin embargo, hay que señalar que existen dificultades, marcadas primordialmente por las características y restricciones del problema, que dictan la aceptación de resultados más modestos; por ejemplo, al nivel de detección de algún óptimo local. Es precisamente en esta etapa donde se ha podido dar solución a muchos casos particulares de problemas de optimización global.

Las técnicas de resolución de POGs se pueden agrupar en dos clases: Métodos Determinísticos y Métodos Estocásticos. En ambos casos, las posibilidades de éxito precisan generalmente de un reforzamiento de

hipótesis adicionales para las funciones que se manejan. La gran mayoría de técnicas se han desarrollado para el caso sin restricciones, es decir $S=\mathbb{R}^m$.

Rinooy Kan y Timmer (1989), clasifican las técnicas determinísticas y estocásticas usadas en la resolución de POGs en cinco bloques:

- 1) Métodos de Partición y Búsqueda.
- 2) Métodos de Aproximación y Búsqueda.
- 3) Métodos de Decrecimiento Global.
- 4) Métodos de mejora de un óptimo local.
- 5) Métodos de Enumeración de Óptimos Locales.

Se observa que los cuatro primeros conjuntos de métodos presentan como fundamento la necesidad de encontrar puntos que mejoren los disponibles en la etapa de cálculo actual. Si fuera fácil encontrar dichos puntos cualesquiera de los métodos referidos resolvería con eficacia cualquier problema de optimización global. Sin embargo, en general, este no es el caso.

Por ello los métodos que más se han desarrollado son los del tipo (5). Estos procedimientos trabajan en dos fases: local y global. En la fase local se determina un óptimo local o un ϵ -óptimo local, mientras que en la fase global se selecciona el mejor de los óptimos locales ya calculados. (ver Boenders y Rinooy Kan (1987), Rinooy Kan y Timmer (1987,1989)).

En el presente trabajo construimos un algoritmo del tipo (5) cuyo proceso secuencial se basa en un modelo estocástico geométrico. En el apartado 2 se introducen la notación y conceptos básicos que se relacionan con las operaciones del método auxiliar de búsqueda local. En el apartado 3 se construye la variable multinomial asociada con las regiones de atracción y se describen técnicas para estimar probabilidades asociadas con esta variable. En el apartado 4 se introduce un algoritmo de optimización global el cuál, en el apartado 5, se aplica a un problema de clasificación

2.- CONCEPTUALIZACION Y METODOLOGIA

Sean $S \subseteq \mathbb{R}^m$, $f: S \rightarrow \mathbb{R}$ una función. S y f se mantendrán fijos, hechos que determinarán el que no se hagan explícitas las dependencias de otros conceptos de f y S . Supondremos que S es compacto y que se pretende encontrar un punto de S que minimice la función f .

Un algoritmo A es, en general, un proceso de cálculo asociado a una transformación $T: S \rightarrow S$. Por conveniencia, identificamos el algoritmo con la

transformación; es decir, diremos que $A(x)=x'$, si y sólo si $T(x)=x'$.

Definición 2.1

Un algoritmo A se denomina **de búsqueda local** (para minimizar f sobre S), si y sólo si se satisfacen las tres condiciones siguientes:

- (i) $\forall x \in S, f(x) \geq f(A(x))$
- (ii) x es un óptimo local si y sólo si $A(x)=x$
- (iii) $\forall x \in S, \exists \lim_{n \rightarrow \infty} A^n(x)$ y $\lim_{n \rightarrow \infty} A^n(x)$ es un óptimo local.

Si denotamos por Z al conjunto de óptimos locales que f tiene en S, llamaremos **función óptimo local** a cualquier función

$$BL: S \rightarrow Z$$

Una forma cómoda de construir una función óptimo local es, cuando se conoce un algoritmo de búsqueda local A, a través de la fórmula:

$$BL_A(x) = \lim_{n \rightarrow \infty} A^n(x)$$

Para cada $z \in Z$ y cada función óptimo local BL, sea

$$RA_{BL}(z) = \{y \in S / BL(y) = z\}$$

Definición 2.2

El conjunto $RA_{BL}(z)$ se llama **región de atracción de z** para la función óptimo local BL.

Se suprimirá el subíndice BL cuando no haya lugar a confusión. Si $D \subseteq \mathbb{R}^m$, por $v(D)$ denotaremos el hipervolumen de D medido en \mathbb{R}^m en términos de la medida de Lebesgue. Por las hipótesis introducidas más arriba, si $D \subseteq S$, entonces $v(D) < \infty$.

3.- PROCESOS SECUENCIALES DE CALCULO DE OPTIMOS LOCALES. REGLAS DE PARADA

De lo dicho en la introducción, se infiere que los métodos del tipo (5) calculan secuencialmente óptimos locales; esto es, elementos del conjunto Z. Una forma de realizar este proceso sería mediante un algoritmo A de búsqueda local y su función óptimo local asociada BL.

El problema de óptimo local se simplificaría si conociéramos el conjunto Z. En este caso, bastaría hallar un $z' \in Z$ tal que $f(z') \leq f(z), \forall z \in Z$. No supone, sin embargo, mucha pérdida de generalidad el suponer que Z es finito. Esta hipótesis, mantenida a lo largo del presente trabajo, se

concretará en suponer que $Z=\{z_1, \dots, z_k\}$, aunque tanto k como los elementos de Z sean desconocidos.

Definición 3.1

$\forall z \in S$, definimos $\xi(z)=e_j$ (vector de \mathbb{R}^k de ceros, salvo un uno en el lugar j -ésimo) si y sólo si $BL(z)=z_j$.

Definición 3.2

$$\text{Llamamos } p_i = \frac{v(RA(z_i))}{v(S)}, \quad i=1, \dots, k$$

p_i es la probabilidad de que al seleccionar aleatoria y uniformemente un elemento $z \in S$, se tenga que $z \in RA(z_i)$.

Es obvio que si seleccionamos aleatoria, independiente y uniformemente N puntos de S , $M=\{x_1, \dots, x_N\}$, supuesto conocido k , la función:

$$\eta(x_1, \dots, x_N) = \sum_{i=1}^N \xi(x_i) \quad \text{es una variable aleatoria multinomial de parámetros } (N, p_1, \dots, p_k).$$

En este trabajo sólo usaremos la función η de forma indirecta.

El proceso que proponemos consiste en seleccionar aleatoria, independiente y uniformemente puntos x_1, \dots, x_N de S y en calcular BL en dichos puntos. En la etapa i -ésima, esto es, después de haber simulado los puntos x_1, \dots, x_i , dispondremos de un conjunto $Z_i=\{z_1, \dots, z_{s(i)}\} \subseteq Z$ de puntos óptimos locales. Una cuestión fundamental en este esquema de trabajo es la de poder determinar si $Z_i=Z$ ó, más bien, decidir si $Z_i=Z$ ó $Z_i \neq Z$ sin cometer excesivos errores.

Para facilitar el proceso en curso, llamamos $p_i^1 = \sum_{z_j \in Z_i} p_j$. Aunque p_i^1 es desconocido, existen formas fáciles de estimarle a través, por ejemplo, del modelo geométrico que se adopta en este trabajo.

Es obvio que si $Z_i=Z$, entonces $p_i^1=1$. Además, como Z es supuesto finito, $p_i^1=1$ equivale a $Z_i=Z$.

Teorema 3.1

Supongamos que $BL(x_i) \notin Z_{i-1}$, $BL(x_j) \in Z_i$, para $j \in (i, i+1, \dots, i+r-1)$, y $BL(x_{i+r}) \notin Z_i$. En estos supuestos, el estimador de máxima verosimilitud para p_1^{i+r-1} es, según el modelo geométrico:

$$\hat{p}_1^{i+r-1} = \frac{r}{r+1}$$

Demostración

Según el modelo probabilístico geométrico, la función de verosimilitud es $FV(p) = p^r(1-p)$. Derivando, obtenemos: $FV'(p) = p^{r-1}(r-(r+1)p)$, la cual se anula únicamente para

$$p = \frac{r}{r+1}$$

Como para $p=0$ y $p=1$, $FV(p)=0$, se tiene que $\hat{p}^{i+r-1} = \frac{r}{r+1}$. ■

El estimador geométrico calculado previamente sólo tiene en cuenta la sucedido desde la etapa i a la $i+r-1$, pero no considera la información obtenida hasta la etapa i -ésima.

Suponemos que hasta la etapa i -ésima se han obtenido los mínimos locales z_1, \dots, z_{s-1} y, que en la etapa $i+r$ -ésima, se ha obtenido el óptimo local z_s . Sea n_j el número de veces que se ha repetido el óptimo local z_j . Entonces, es claro que $1 + \sum_{j=1}^{s-1} n_j = i+r$.

En el proceso que seguiremos, en la etapa i habremos calculado para $q_1^i = \sum_{j=1}^{s-1} p_j$ un estimador \hat{q}_1^i . Es claro que el correspondiente estimador de q_1^{i+r-1} , que llamaremos h_1^{i+r-1} , debe ser función de q_1^i y de \hat{p}_1^{i+r-1} . Convendremos en que dicha función es una combinación convexa de ambos estimadores, es decir:

$$h_1^{i+r-1} = \frac{i}{r+i} \hat{q}_1^i + \frac{r}{r+i} \hat{p}_1^{i+r-1}$$

Para obtener el estimador de q_1^{i+r} , hay que sumar a h_1^{i+r-1} un estimador de la probabilidad de p_s . Teniendo en cuenta que un estimador de p_j es $h_1^{i+r-1} = \frac{n_j}{i+r-1}$ y un estimador de p_s es $h_1^{i+r-1} = \frac{1}{i+r-1}$, se obtiene que

$$\hat{q}_1^{i+r} = h_1^{1+r-1} \frac{i+r}{i+r-1} = \frac{i}{r+i-1} \hat{q}_1^i + \frac{r}{r+i-1} \hat{p}_1^{1+r-1}$$

Por tanto, cuando nos encontremos en una secuencia de cálculo de óptimos locales en la que se producen repeticiones de óptimos obtenidos previamente, el valor del estimador de la suma de las probabilidades de los óptimos ya obtenidos aumenta. En el procedimiento que propondremos a continuación, este comportamiento se extiende al caso en que se obtiene un nuevo óptimo local.

4.- ALGORITMO: NOTACION Y ESQUEMA

Antes de presentar el esquema algorítmico, introduzcamos alguna notación básica referida a los elementos que intervienen en él.

4.1.- Notación

- \hat{q}_1^i representa la estimación de la suma de probabilidades de los óptimos locales hallados por el algoritmo hasta la iteración i -ésima.
- LIS es un vector que guarda los índices los óptimos locales encontrados.
- NCU es un vector donde se almacenan las frecuencias absolutas de los óptimos locales.
- NRE es una variable que mide el número de iteraciones que han transcurrido desde que se añadió a LIS el último óptimo local encontrado.
- k es el número de óptimos locales encontrados.
- S_1 es un paralelepípedo que contiene a S .
- BL es una función óptimo local.
- i es la variable que indica el número de iteraciones.
- r^i y m^i son variables necesarias para realizar los cálculos.

4.2.- Algoritmo

Paso 0

Colocar $\hat{q}_1^1 = \frac{1}{m}$, $NRE=0$, $k=0$, $\varepsilon_1 > 0$, $\varepsilon_2 > 0$ suficientemente pequeños, $M_2 = m^1 = M$, $IN=1$, $i=1$ e ir al paso 1

Paso 1

Simular, aleatoria y uniformemente puntos $u \in S_1$, hasta hallar un punto $w \in S$. Hacer $x_1 = w$, evaluar $z_1 = BL(x_1)$ e ir al paso 2.

Paso 2

Si $i=1$, ir al paso 3. En otro caso, ir al paso 4.

Paso 3

Colocar $LIS(1)=z_1$, $NCU(1)=1$, $k=1$, $NRE=1$, $i=i+1$ e ir al paso 1.

Paso 4

Si existe $j \in \{1, \dots, k\}$ tal que $z = LIS(j)$, colocar $IN = IN + 1$, $NCU(j) = NCU(j) + 1$, $NRE = NRE + 1$, $r^{i+1} = r^i + IN$, $m^{i+1} = m^i + IN$, $i = i + 1$ e ir al paso 7. En otro caso, hacer $k = k + 1$, $LIS(k) = z$, $NCU(k) = 1$, $r^{i+1} = r^i$, $m^{i+1} = m^i + IN$, $i = i + 1$ e ir al paso 5.

Paso 5

Calcular $\lambda_1 = \frac{i - NRE}{i}$, $h_1^1 = \frac{NRE}{NRE + 1} \lambda_1 + (1 - \lambda_1) q_1^{i-1}$. Ir al paso 6.

Paso 6

Sea $q_1^1 = \frac{M_1}{M_2}$, fracción irreducible de números naturales. Calcular $IN = [M_1/i]$, $NRE = 1$, $r^1 = M_1 + IN$, $m^1 = M_2$, hacer $q_1^1 = \frac{r^1}{m^1}$ e ir al paso 1.

Paso 7

Calcular $q_1^1 = \frac{r^1}{m^1}$ e ir al paso 8.

Paso 8

Si $q_1^1 > 1 - \epsilon_1$ ó $\frac{1}{NRE} < \epsilon_2$, ir al paso 9. En otro caso, ir al paso 1.

Paso 9

$\forall j \in \{1, \dots, k\}$, estimar p_j , probabilidad de atracción de z_j , por $NCU(j)/i$.

5.- APLICACION: OPTIMIZACION GLOBAL EN CLASIFICACION

Suponemos que disponemos de un conjunto Ω de n objetos O_1, \dots, O_n . Sobre cada objeto O_i medimos m variables X_1, \dots, X_m ; obteniendo la matriz $A = (a_{ij})$, donde $a_{ij} = X_j(O_i)$. Sea $a_i = (a_{i1}, \dots, a_{im})$, para $i = 1, \dots, n$, la fila i -ésima de A . Se supone que los vectores fila a_i son todos distintos.

El problema que planteamos es el de hallar una clasificación óptima de Ω en k clases, con $k < n$.

Dado $S = \prod_{j=1}^m [c_j, C_j]$, siendo $c_j = \min_{1 \leq i \leq n} a_{ij}$, $C_j = \max_{1 \leq i \leq n} a_{ij}$, usando una distancia d definida sobre \mathbb{R}^m , definimos la función $f: S \subseteq \mathbb{R}^{km} \rightarrow \mathbb{R}$, de la siguiente manera:

Para $y_1, \dots, y_k \in S$, sean $D_j = \{i / d(a_i, y_j) < d(a_i, y_h) \text{ para } h < j \text{ y } d(a_i, y_j) \leq d(a_i, y_h) \text{ para } h \geq j\}$, $\forall j \in \{1, \dots, k\}$. Definiremos entonces:

$$f(y_1, \dots, y_k) = \sum_{j=1}^k \sum_{i \in D_j} d(a_i, y_j)$$

La optimalidad de la clasificación buscada se reduce a establecer el óptimo global de f .

Para usar el algoritmo desarrollado previamente, necesitamos un procedimiento de búsqueda local que esquemizamos de la siguiente manera:

Algoritmo de búsqueda local

Paso 0

Elegir una distancia d sobre \mathbb{R}^m . Hallar, por simulación, k puntos $y_1^1, \dots, y_k^1 \in \prod_{j=1}^m [c_j, C_j]$, hacer $s=1$ e ir al paso 1.

Paso 1

Calcular $D_j^s = \{i / d(a_i, y_j^s) < d(a_i, y_h^s) \text{ para } h < j \text{ y } d(a_i, y_j^s) \leq d(a_i, y_h^s) \text{ para } h \geq j\}$, $\forall j \in \{1, \dots, k\}$. Si $s=1$ ó $D_j^s \neq D_j^{s-1}$ para algún $j \in \{1, \dots, k\}$, ir al paso 2. En otro caso, si $D_j^s = D_j^{s-1}, \forall j \in \{1, \dots, k\}$, parar con la clasificación óptima local $\{D_j^s\}_{j=1, \dots, k}$.

Paso 2

$\forall j \in \{1, \dots, k\}$ tal que $D_j^s = \emptyset$, hallar $a_r \in D_t^s$ tal que $d(a_r, y_j^s) \geq d(a_i, y_h^s)$, $\forall i \in D_h^s, \forall h \in \{1, \dots, k\}$. Hacer $y_j^{s+1} = a_r$ y redefinir $D_j^s = \{r\}$. Ir al paso 3.

Paso 3

$\forall j \in \{1, \dots, k\}$, sea y_j^{s+1} el punto que minimiza la función:

$$g_j(y) = \sum_{i \in D_j} d(a_i, y)$$

Hacer $s=s+1$ e ir al paso 1.

6.- EXPERIENCIAS COMPUTACIONALES

Hemos programado los anteriores procedimientos en lenguaje FORTRAN para ejecutarlos sobre el VAX PDP-11 del Centro de Cálculo de la Universidad de La Laguna.

Los ejemplos de problemas de clasificación que aquí presentamos constan de 50 y 100 objetos, sobre cada uno de los que se han medido 5, 10 y 15 variables, respectivamente. Los resultados obtenidos son los siguientes:

M=50, M=5, K=2, OPTIMOS LOCALES= 3

OPT. GLOBAL: FUN(1)= 6883.48681641, PROB.= 0.8260869383812

CLASE(1): 8 11 15 19 21 23 25 26 28 29 30 31 32 34 35 36 37 40 43 44 50

CLASE(2): 1 2 3 4 5 6 7 9 10 12 13 14 16 17 18 20 22 24 27 33 38 39 41 42

45 46 47 48 49

N=50, M=5, K=3, OPTIMOS LOCALES = 5

OPT. GLOBAL: FUN(4)= 11103.04003906, PROB.= 0.0500000007451

CLASE(1): 8 11 15 21 26 29 34 36

CLASE(2): 1 2 3 4 5 6 7 9 10 12 13 14 16 17 18 20 22 24 33 38 39 41 45 46
47 48 49

CLASE(3): 19 23 25 27 28 30 31 32 35 37 40 42 43 44 50

N=100, M=5, K=2, OPTIMOS LOCALES= 3

OPT. GLOBAL: FUN(3)= 13901.25292969, PROB.= 0.3750000000000

CLASE(1): 8 11 15 19 21 23 25 26 27 28 29 30 31 32 34 35 36 37 38 40 42 43
44 50 54 55 58 60 61 62 64 68 70 71 75 76 77 79 81 82 83 85 86 87 90 91 92
95 98 99 100

CLASE(2): 1 2 3 4 5 6 7 9 10 12 13 14 16 17 18 20 22 24 33 39 41 45 46 47
48 49 51 52 53 56 57 59 63 65 66 67 69 72 73 74 78 80 84 88 89 93 94 96 97

N=100, M=5, K=3, OPTIMOS LOCALES = 4

OPT. GLOBAL: FUN(2)= 22959.04882813, PROB.= 0.3333333432674

CLASE(1): 11 15 21 26 34 54 62 72 77 81 99 100

CLASE(2): 8 19 23 25 27 28 29 30 31 32 35 36 37 38 40 42 43 44 50 55 58 60
61 64 68 70 71 75 76 79 82 83 85 86 87 90 91 92 95 98

CLASE(3): 1 2 3 4 5 6 7 9 10 12 13 14 16 17 18 20 22 24 33 39 41 45 46 47
48 49 51 52 53 56 57 59 63 65 66 67 69 73 74 78 80 84 88 89 93 94 96 97

N=50, M=10, K=2, OPTIMOS LOCALES= 3

OPT. GLOBAL: FUN(3)= 9167.04394531, PROB.= 0.4615384638309

CLASE(1): 1 2 3 4 5 6 7 9 10 12 13 14 16 17 18 20 22 24 33 39 41 45 46 47
48 49

CLASE(2): 8 11 15 19 21 23 25 26 27 28 29 30 31 32 34 35 36 37 38 40 42 43
44 50

N=50, M=10, K=3, OPTIMOS LOCALES= 23

OPT. GLOBAL: FUN(7)= 14190.12500000, PROB.= 0.0114942528307

CLASE(1): 2 4 7 9 10 12 39 41 46 47 49

CLASE(2): 8 11 15 19 21 23 25 26 27 28 29 30 31 32 34 35 36 37 38 40 42 43

44 50

CLASE(3): 1 3 5 6 13 14 16 17 18 20 22 24 33 45 48

N=100, M=10, K=2, OPTIMOS LOCALES =4

OPT. GLOBAL: FUN(3)= 19883.20117188, PROB.= 0.0769230797887

CLASE(1): 8 11 15 21 48 54 72 77 81 87 99 100

CLASE(2): 1 2 3 4 5 6 7 9 10 12 13 14 16 17 18 19 20 22 23 24 25 26 27 28
29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 49 50 51 52 53 55
56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 73 74 75 76 78 79 80 82 83
84 85 86 88 89 90 91 92 93 94 95 96 97 98

N=100, M=10, K=3, OPTIMOS LOCALES =8

OPT. GLOBAL: FUN(8)= 30298.49414063, PROB.= 0.0454545468092

CLASE(1): 8 11 15 21 54 77 81 87 99 100

CLASE(2): 1 2 3 4 5 6 7 9 10 12 13 14 16 17 18 20 22 24 33 39 41 45 46 47
48 49 51 52 53 56 57 59 63 65 66 67 69 72 73 74 78 80 84 88 89 93 94 96 97
CLASE(3): 19 23 25 26 27 28 29 30 31 32 34 35 36 37 38 40 42 43 44 50 55 58
60 61 62 64 68 70 71 75 76 79 82 83 85 86 90 91 92 95 98

N=50, M=15, K=2, OPTIMOS LOCALES= 3

OPT. GLOBAL: FUN(3)= 12466.16796875, PROB. = 0.0714285746217

CLASE(1): 1 47 48

CLASE(2): 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 49 50

N=50, M=15, K=3, OPTIMOS LOCALES = 26

OPT. GLOBAL: FUN(6)= 18308.58203125, PROB. = 0.0098039219156

CLASE(1): 1 30

CLASE(2): 2 4 6 9 14 17 20 22 24 26 27 31 32 33 34 35 36 37 38 39 41 43 44
45 46 48 49 50

CLASE(3): 3 5 7 8 10 11 12 13 15 16 18 19 21 23 25 28 29 40 42 47

N= 100, M= 15, K=2, OPTIMOS LOCALES =1

OPT. GLOBAL: FUN(1)= 24544.46093750, PROB.= 1

CLASE(1): 1 5 38 48 53 77 99 100

CLASE(2): 2 3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

27 28 29 30 31 32 33 34 35 36 37 39 40 41 42 43 44 45 46 47 49 50 51 52 54
55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 78 79 80
81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98

N=100, M=15, K=3, OPTIMOS LOCALES =6

OPT. GLOBAL: FUN(5)= 35680.28515625, PROB.= 0.1304347813129

CLASE(1): 1 5 33 38 48 53 77 99 100

CLASE(2): 4 7 8 9 10 12 16 19 21 23 25 26 29 30 31 32 34 35 36 37 40 42 43
44 45 46 47 49 50 52 54 55 58 60 61 62 66 68 69 70 71 72 73 74 75 76 78 79
81 83 85 87 88 90 91 92 94 96 98

CLASE(3): 2 3 6 11 13 14 15 17 18 20 22 24 27 28 39 41 51 56 57 59 63 64 65
67 80 82 84 86 89 93 95 97

Nota: PROB representa la probabilidad estimada de la correspondiente región de atracción. FUN es el valor correspondiente de la función objetivo. CLASE contiene los índices correspondientes de las filas de la matriz de entrada.

7.- REFERENCIAS

- BAZARAA, M. S. y SHETTY, C.M. (1979) "Nonlinear Programming". John Wiley
- BERBEE, H.C.P., BOENDER, G.C.E., RINNOOY KAN, A.H.G., SCHEFFER, C.L., SMITH, R.L. and TELGEN, J. (1987) "Hit-and-Run Algorithms for the Identification of Nonredundant Linear Inequalities". *Mathematical Programming* 37, 184-207
- BOENDER, G.C.E., RINNOOY KAN, A.H.G., STOUGIE, L. and TIMMER, G.T. (1982) "A Stochastic Method for Global Optimization". *Mathematical Programming* 22, 125-140
- BOENDER, G.C.E. (1984) "The Generalized Multinomial Distribution: A Bayesian Analysis and Applications". Ph. D. Thesis, Erasmus University (Rotterdam)
- BOENDER, G.C.E., RINNOOY KAN, A.H.G. (1987) "Stopping Rules for Global Optimization". *Mathematical Programming* 37, 59-80.
- DIXON, L.C.W. and SZEGO, G.P. (eds.) (1975) "Towards Global Optimization". North Holland.
- DIXON, L.C.W. and SZEGO, G.P. (eds.) (1978) "Towards Global Optimization 2". North Holland.
- ERDOS, P. and SPENCER, J. (1979) "Probabilistic Methods in Combinatorics". Academic Press.
- LUEMBERGER, D. G. (1984) "Linear and Nonlinear Programming". Addison Wesley

- McCORMICK, G.P. (1983) "Nonlinear Programming. Theory, algorithms and applications". John Wiley
- RINNOOY KAN,A.H.G. and TIMMER.G.T.(1987) "Stochastic Global Optimization I: Clustering Methods". Mathematical Programming 39,27-56
- RINNOOY KAN,A.H.G. and TIMMER.G.T.(1987) "Stochastic Global Optimization II: Multi Level Methods". Mathematical Programming 39,57-78
- RINNOOY KAN,A.H.G. and TIMMER.G.T.(1989) "Global Optimization". En Optimization, Nemhauser, Rinooy Kan y Todd (eds). North Holland.
- RUBINSTEIN,R.Y.(1981)"Simulation and the Monte Carlo Method". John Wiley.

Recibido: 9 de Enero de 1991