

# **Análisis de algunas metaheurísticas creadas a partir de “Optimización Gravitatoria”**

Zapatero Moreno, M<sup>a</sup> José; Alegre Martínez, Jesús y Pacheco Bonrostro, Joaquín  
*Departamento de Economía Aplicada  
Universidad de Burgos*

## **RESUMEN**

En este trabajo proponemos una nueva metaheurística para obtener óptimos globales de funciones no lineales, algunas multimodales. Dicha metaheurística surge uniendo dos algoritmos: el primero, “Optimización Gravitatoria”, concibe el espacio de soluciones análogamente al espacio-tiempo relativista, en el que la métrica es modificada por las diferentes partículas en él inmersas. En la heurística el papel de la atracción gravitatoria lo juega la función objetivo; el óptimo se encontraría en el punto donde se encuentre la mayor masa. Como esta posición se desconoce, se procede a medir la variación de la geometría; tal como en relatividad general la variación de la geometría nos lleva a la mayor masa, en la heurística nos conduce al óptimo global. El segundo algoritmo es el conocido simplex de Nelder-Mead que consiste en, partiendo de un simplex no degenerado inicial en el espacio de soluciones, variar dicho poliedro para optimizar la función objetivo.

**Palabras claves:** Metaheurística; optimización no lineal; Optimización Gravitatoria; S.G.O.; función multimodal; algoritmo Nelder-Mead.

**Área temática:** Optimización.

## **ABSTRACT**

In this work we propose a new metaheuristic for global optimization of nonlinear functions, some multimodal. This metaheuristic arises from the union of two algorithms: the first, "Optimising Gravity", sees the solution space analogous to the relativistic space-time, in which the metric is modified by the different particles embedded in it. In the heuristic the role of the gravitational pull is played by the objective function, the optimum would be at the point where the greatest mass is. As this position is unknown, it is necessary to measure the change of geometry; the same as in general relativity the change in geometry leads to the greater mass in the heuristic leads to the global optimum. The second algorithm is the known Nelder-Mead simplex that consists on, starting from an initial nondegenerate simplex in the solution space, change the polyhedron in order to optimize the objective function.

**Keywords:** Metaheuristics, nonlinear optimization, Space Gravitational Optimization SGO; function multimodal Nelder-Mead algorithm.

## 1. INTRODUCCIÓN

En trabajos anteriores ya hemos analizado el algoritmo S.G.O. (ver Zapatero M.J. y otros 2009 y 2010), ahora lo complementamos con el algoritmo Nelder-Mead usado como técnica de intensificación.

*Optimización gravitatoria, SGO*, es una heurística introducida recientemente por Hsiao, Chuang, Jiang and Chien (2005). En ella se concibe el espacio de soluciones de forma similar al espacio-tiempo relativista.

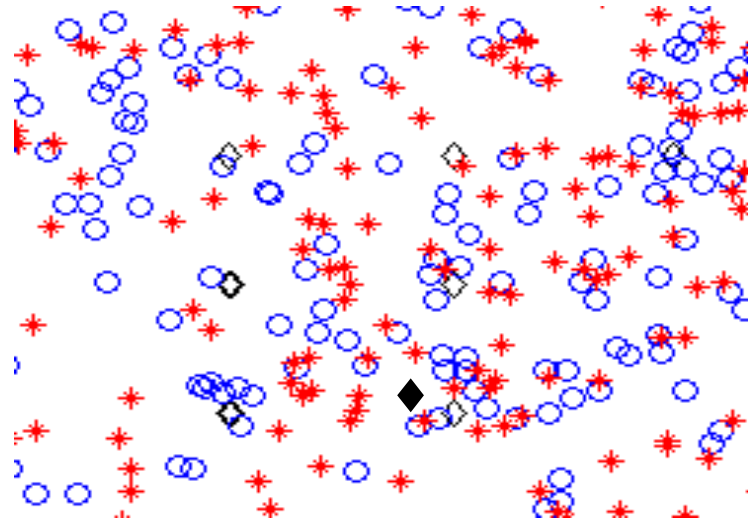
Hemos implementado esta novedosa heurística para obtener los óptimos globales de un conjunto de funciones de optimización difícil, funciones multimodales con un número elevado o muy elevado de óptimos locales.

Según la teoría de la relatividad general el espacio se curva por efecto del campo gravitatorio. Las partículas (asteroides, agentes de búsqueda, etc...) se aceleran hacia las masas grandes por las variaciones en la geometría del espacio tiempo.

Consideraremos en este procedimiento unos agentes de búsqueda, denominamos asteroides. Estos se están moviendo en el Universo, que será el espacio de búsqueda. Cuando un asteroide es capturado por la gravedad de una masa grande tiene dos posibilidades: salir disparado como una honda y, en ese caso, seguir buscando otra masa pesada en el universo o chocar contra su superficie. Para evitar esto último en el algoritmo se supone que el cuerpo real de las masas pesadas no existe realmente. Así cuando un asteroide se acerca al centro de una masa pesada, se incrementa considerablemente la energía cinética de la partícula por la gravedad de la masa, y el asteroide adquiere suficiente velocidad para escapar de la región del espacio que ocupa la masa pesada (óptimo local).

Un algoritmo así diseñado requiere poca memoria y poder computacional. La posibilidad de caer en óptimos locales es muy pequeña y los asteroides tienen muchas posibilidades de salir disparados fuera del óptimo local. Tampoco quedarán atrapados por el óptimo global, en el caso que lleguen a hasta él, sino que seguirán la búsqueda.

En el ejemplo de la figura que se muestra a continuación se han considerado tan solo dos asteroides, representados uno de ellos por círculos azules y otro por asteriscos rojos y observamos como los asteroides han sido atraídos hacia los diferentes óptimos locales ( $\diamond$ ), han seguido su trayecto y han pasado cerca del óptimo global ( $\blacklozenge$ ).



El algoritmo *Nelder\_Mead* parte de un simplex no degenerado en cuyos vértices tenemos aproximaciones al óptimo global. Dicho simplex se va modificando, bien al cambiar alguno de sus vértices por otro en el que la función objetivo haya disminuido, o bien al contraer todo el poliedro si no se ha conseguido mejorar la función objetivo con el simplex inicial. Es por tanto un algoritmo de carácter local que carece además de componente aleatoria.

De esta forma ambos algoritmos son complementarios ya que, en primer lugar se ejecuta el S.G.O. con lo que tenemos una exploración del espacio de soluciones y un primer acercamiento al óptimo global. Una vez que disponemos de suficientes aproximaciones localizadas por los asteroides, escogemos las mejores posiciones para construir el simplex inicial no degenerado sobre el que trabajará el algoritmo *Nelder-Mead* para llevarnos al óptimo global.

## 2. DESCRIPCIÓN DE LOS ALGORITMOS

### 2.1 Optimización gravitatoria

#### 2.1.1 Descripción general

El espacio de soluciones está formulado como un espacio tiempo curvado de acuerdo al concepto de Einstein. Un número  $n$  de asteroides o agentes de búsqueda se mueve en el espacio de soluciones y va buscando el cuerpo con mayor masa (óptimo

global). Denotaremos por  $P[i]$ ,  $V[i]$ ,  $A[i]$  con  $i = 1, \dots, n$  la posición, velocidad y aceleración de cada asteroide, y por  $f$  a la función objetivo.

**Paso I:** Inmersión de los asteroides en el espacio de búsqueda:

A los asteroides inicialmente se les asigna una posición aleatoria,  $P_0[i]$  ( si el número de variables de la función es *dimension\_espacio*, cada componente será denotada por  $P_0[i][k]$ ,  $k = 1..dimension\_espacio^1$ ), dentro de su región factible, y así mismo se les asigna una velocidad aleatoria,  $V_0[i]$ , con valores en la región factible, para evitar, en lo posible, que el asteroide se "escape" a posiciones no factibles.

De partida los asteroides tienen una aceleración nula,  $A_0[i]$ , y a partir de ese momento quedan sometidos únicamente a la atracción gravitatoria generada por  $f$  (ley de la gravedad de Newton). Se va a almacenar la mejor solución por la que pasa cada asteroide  $i$ , que inicialmente se supone  $f\_best[i] = +\infty$ .

Los asteroides tratan de buscar los objetos de masa más pesada. Los que no queden atrapados en óptimos locales se dirigirán a la masa más pesada del Universo, el óptimo global.

**Paso II:** Buscando la variación del espacio tiempo:

El espacio tiempo del universo está curvado por el campo gravitacional, y la variación de la geometría del espacio-tiempo modifica la aceleración de los asteroides. Como desconocemos la posición del óptimo (masa mayor) para determinar la trayectoria de los asteroides debemos detectar la variación en la geometría del espacio tiempo: para ello se emplea un parámetro llamado rango de detección  $r_d$ . Para calcular la aceleración de cada asteroide por variación espacio-tiempo, se emplea la siguiente fórmula que para el asteroide  $i$ , en su componente  $k$  es:

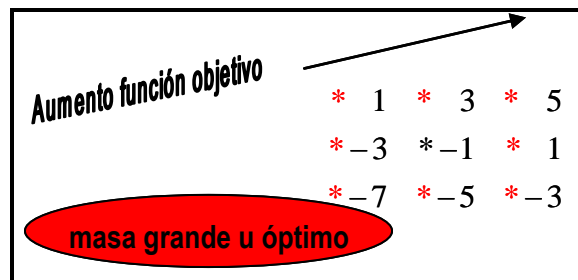
$$A[i][k] = G(\{f(P[i]) - f(P[i][1], \dots, P[i][k] + r_d, \dots)\} + \{f(P[i][1], \dots, P[i][k] - r_d, \dots) - f(P[i])\}) = G\{f(P[i][1], \dots, P[i][k] - r_d, \dots) - f(P[i][1], \dots, P[i][k] + r_d, \dots)\}$$

donde  $G$  es un nuevo parámetro que representa la intensidad del campo gravitatorio. En la fórmula, para calcular cada componente de la aceleración, se multiplica  $G$  por la suma de las variaciones parciales de la función objetivo en el ámbito del rango de detección para dicha componente.

---

<sup>1</sup> En lo que resta de trabajo asumimos que la dimensión de las magnitudes vectoriales será de *dimension\_espacio*.

En el ejemplo del gráfico observamos la tendencia hacia posiciones con menor valor de la función  $f(x, y) = 4y + 2x - 7$ . Los asteriscos representan diversos puntos, cuya distancia en sentido horizontal y vertical es un rango de detección (en este caso considerando  $r_d=1$ ) y a su lado el valor en ellos de la función objetivo. El asterisco abajo a la izquierda estaría situado en el punto (0,0), el de encima de él en el punto (0,1), etc...



Así, en este caso para el punto central la aceleración producida por la variación de la geometría sería (suponiendo que  $G=1$ ) en cada componente de  $A_x = -2-2 = -4$  y  $A_y = -4-4 = -8$ . Así si la velocidad de partida fuese (10, 5) pasaría a ser (6, -3)

**Paso III:** Actualizando la velocidad y posición de los asteroides:

Tras cada iteración hay que recalcular la velocidad, y actualizar la posición de los asteroides a partir de las nuevas velocidades: así para el asteroide  $i$  su velocidad es  $V[i]=V[i]+A[i]$ , y la posición del asteroide  $P[i]=P[i]+V[i]$

Podría ocurrir que algún asteroide se escape del espacio factible, en ese caso hemos recuperado su factibilidad, y para ello el procedimiento que hemos utilizado consiste en devolver al asteroide, de forma aleatoria, a la región factible. Simultáneamente se han ido contando las infactibilidades, y su cómputo se ha tenido en cuenta a la hora de dar valores a los parámetros, procurando que el número de infactibilidades no sea muy elevado.

**Paso IV:** Actualización de la mejor solución de cada asteroide: En una matriz  $posicion\_best[i]$  almacenamos la mejor posición de cada asteroide y en otra matriz,  $f\_best[i]$ , el mejor valor de la función objetivo. Tras cada iteración si la solución obtenida por el asteroide  $i$  es mejor que la mejor conocida hasta ese momento son actualizadas  $posicion\_best[i]$  y  $f\_best[i]$ .

### 2.1.1 Modificaciones del algoritmo

De la misma forma que en el trabajo de Hsiao y otros (2005) consideramos las siguientes modificaciones del algoritmo:

1.- Efecto de los asteroides en la geometría: añadimos el efecto de los asteroides en la modificación de la geometría del espacio tiempo; éstos van a producir un campo gravitatorio que debe de ser considerado. En la fórmula que nos servía para el cálculo de la aceleración hay que añadir  $\alpha.C[k]/d_i^2$ , donde C es el centro de masas de todos los asteroides, C[k] su componente k-ésima y  $d_i$  es la distancia entre el asteroide  $i$  y el centro de masas C. En cada iteración ejecutaremos un procedimiento llamado *Calcular\_posicion\_centro\_masas\_asteroides*. Para calcular el centro de masas

utilizaremos que  $C = \frac{\sum_{i=1}^n P[i]}{n}$ .

2.- Contracción o expansión del Universo: en cada componente la actualización de la velocidad viene dada por  $velocidad[i] := \beta * velocidad[i] + aceleracion[i]$ , donde  $\beta$  será un parámetro mayor que 1 si se supone que el universo se contrae (cuando todo parece aproximarse a los asteroides) y menor que 1 si se expande (cuando todo parece alejarse de los asteroide).

El pseudocódigo del algoritmo es.

***Procedimiento Gravita (mejorsolucion)***

*Inicializar aleatoriamente todos los asteroides*

*Mientras  $t < max\_iter$  hacer*

*Calcular posición centro masas asteroides;*

*Para cada asteroide  $i$  hacer*

*Calcular aceleración asteroide por variación espacio-tiempo;*

*Calcular velocidad tras aceleración;*

*Calcular posición tras variación velocidad ;*

*Si no factible  $i$  HacerFactible;*

*Obtener funcion\_objetivo;*

*Si Valor\_funcion  $< f\_best[i]$  entonces actualizar posicion\_best[i] y  $f\_best[i]$*

*$t := t + 1$*

**2.1 El algoritmo Nelder-Mead**

El simplex de Nelder-Mead (también denominado simplex no-lineal) es un algoritmo heurístico propuesto por John Nelder y Roger Mead en 1965 (Lagarias J. y otros 1998) para la optimización de funciones objetivo sin restricciones.

Dicho algoritmo, que sólo usa valores de la función, no de su derivada, se basa en conceptos geométricos y consiste en, partiendo de  $dimension\_espacio + 1$  (dim, a

partir de ahora) puntos distintos de  $\mathbb{R}^{\dim}$  (donde se localizan las  $\dim+1$  aproximaciones iniciales al óptimo buscado) construir su envolvente convexa en el espacio  $\dim$ -dimensional, i.e el poliedro más simple, no degenerado, formado por dichos vértices y posteriormente ir modificando el poliedro de forma que en los nuevos vértices la función objetivo mejore, es decir, disminuya.

Ejemplos de simplex serían: el segmento que une dos puntos en  $\mathbb{R}$ , el triángulo que une 3 puntos no alineados en  $\mathbb{R}^2$ , el tetraedro que une 4 puntos no coplanarios en  $\mathbb{R}^3$  etc.

La deformación y variación del poliedro se hace en base a 4 operaciones básicas: **reflexión, expansión, contracción y encogimiento**, que llevan asociados 4 parámetros respectivamente que denotaremos:  $\rho, \chi, \gamma$  y  $\sigma$ .

Siguiendo el algoritmo Nelder-Mead, dichos parámetros deben satisfacer

$$\rho > 0, \quad \chi > 1, \quad 0 < \gamma < 1, \quad 0 < \sigma < 1 \quad \chi > \rho$$

Para nuestras experiencias computacionales hemos asumido los valores

$$\rho = 1, \quad \chi = 2, \quad \gamma = \frac{1}{2}, \quad \sigma = \frac{1}{2}$$

que son los valores usados en el algoritmo estándar.

### Desarrollo del algoritmo

Veamos cómo realizar la iteración  $k$ -ésima:

Partimos de  $\dim+1$  puntos que forman un simplex no degenerado de  $\mathbb{R}^{\dim}$ , si  $k = 1$  tendremos los  $\dim+1$  vértices iniciales.

El objetivo de la iteración es obtener un nuevo simplex variando uno o más de los vértices que lo forman.

#### Paso 1: ordenación

En primer lugar ordenamos los puntos de forma que los valores de la función objetivo en dichos puntos estén ordenados de forma creciente, i.e.

$$x_1^{k-1}, x_2^{k-1}, \dots, x_{\dim}^{k-1}, x_{\dim+1}^{k-1} \quad k \geq 1 \quad \text{tal que}$$

$$f(x_1^{k-1}) \leq f(x_2^{k-1}) \leq \dots \leq f(x_{\dim}^{k-1}) \leq f(x_{\dim+1}^{k-1})$$

Puesto que nuestro problema es *minimizar la función objetivo* entendemos que el primer punto es el mejor, el último el peor, etc.

#### Paso 2: reflexión



Hallamos el punto de reflexión que será el punto simétrico, ponderado por  $\rho$ , del peor respecto al centro de gravedad que forman los  $\text{dim}$  mejores, esto es:

$$\text{Dado}^2 \quad \bar{x} = \frac{1}{\text{dim}} \sum_{i=1}^{\text{dim}} x_i^{k-1}$$

$$\text{Entonces} \quad x_{\text{refle}} = \bar{x} + \rho(\bar{x} - x_{\text{dim}+1})$$

Evaluamos la función objetivo  $f(x_{\text{refle}})$

**Caso I: el punto de reflexión ha mejorado al mejor**

$$f(x_{\text{refle}}) < f(x_1)$$

**Paso 3: expansión**

Calculamos el punto de expansión respecto al centro de gravedad para ver si podemos mejorar aún más

$$x_{\text{expan}} = \bar{x} + \chi(x_{\text{refle}} - \bar{x})$$

De nuevo evaluamos la función objetivo  $f(x_{\text{expan}})$

Si hemos mejorado, i.e.  $f(x_{\text{expan}}) < f(x_{\text{refle}})$  aceptamos el punto de expansión y sacamos al peor punto para formar el nuevo simplex.

Si no hemos mejorado, i.e.  $f(x_{\text{expan}}) \geq f(x_{\text{refle}})$  aceptamos el punto reflejado y sacamos al peor punto para formar el nuevo simplex. **Fin de la iteración**

**Caso II: el punto de reflexión no ha mejorado al mejor pero sí ha mejorado a alguno de los  $n$  mejores.**

$$f(x_1) \leq f(x_{\text{refle}}) < f(x_{\text{dim}})$$

Aceptamos el punto reflejado y sacamos al peor punto para formar el nuevo simplex. **Fin de la iteración**

**Caso III: el punto de reflexión no ha mejorado a ninguno de los  $n$  mejores**

$$f(x_{\text{refle}}) \geq f(x_{\text{dim}})$$

**Paso 4: contracción**

La contracción se hace hacia el punto reflejado o hacia el peor punto dependiendo de cual sea mejor de los dos

---

<sup>2</sup> Para no entorpecer la notación omitiremos a partir de este momento el superíndice  $k$ .

**-Hacia fuera:** si el punto reflejado es mejor que el peor i.e.

$$f(x_{refle}) < f(x_{dim+1})$$

$$x_{con-f} = \bar{x} + \gamma(x_{refle} - \bar{x})$$

Evaluamos la función objetivo en el punto contraído  $f(x_{con-f})$ .

Si hemos mejorado, i.e. ,  $f(x_{con-f}) < f(x_{refle})$  aceptamos el punto contraído y sacamos al peor para formar el nuevo simplex. **Fin de la iteración**

Si no hemos mejorado, i.e.  $f(x_{con-f}) \geq f(x_{refle})$  vamos al **paso 5**.

**- Hacia dentro:** si el punto reflejado iguala o empeora el peor i.e.

$$f(x_{refle}) \geq f(x_{dim+1})$$

$$x_{con-d} = \bar{x} - \gamma(\bar{x} - x_{dim+1})$$

Evaluamos la función objetivo en el punto contraído  $f(x_{con-d})$ .

Si hemos mejorado i.e.  $f(x_{con-d}) < f(x_{dim+1})$  aceptamos el punto contraído y sacamos al peor para formar el nuevo simplex **Fin de la iteración**

Si no hemos mejorado i.e.  $f(x_{con-d}) \geq f(x_{dim+1})$  vamos al **Paso 5**.

### **Paso 5: encogimiento**

Tomamos n puntos nuevos de la forma

$$v_i = x_1 + \sigma(x_i - x_1) \quad i = 2, \dots, \text{dim}+1$$

El nuevo simplex tiene como vértices  $x_1, v_2, \dots, v_{\text{dim}+1}$ . **Fin de la iteración**

En la figura siguiente, obtenida de Lagarias 1998, tenemos cinco gráficos de simplex: en el primero vemos cómo ha variado tras una reflexión, en el segundo tras una reflexión – expansión, en el tercero y cuarto tras una contracción (hacia fuera y hacia dentro respectivamente) y en el último tras un encogimiento. En todos los casos el simplex inicial aparece con líneas punteadas.

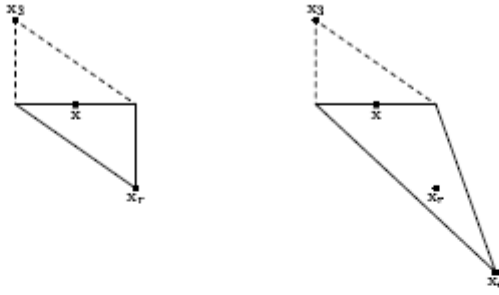
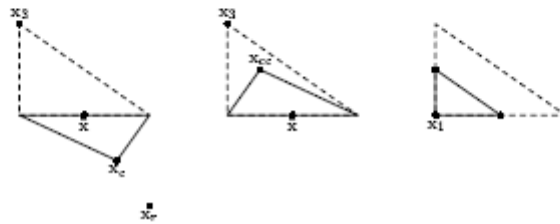
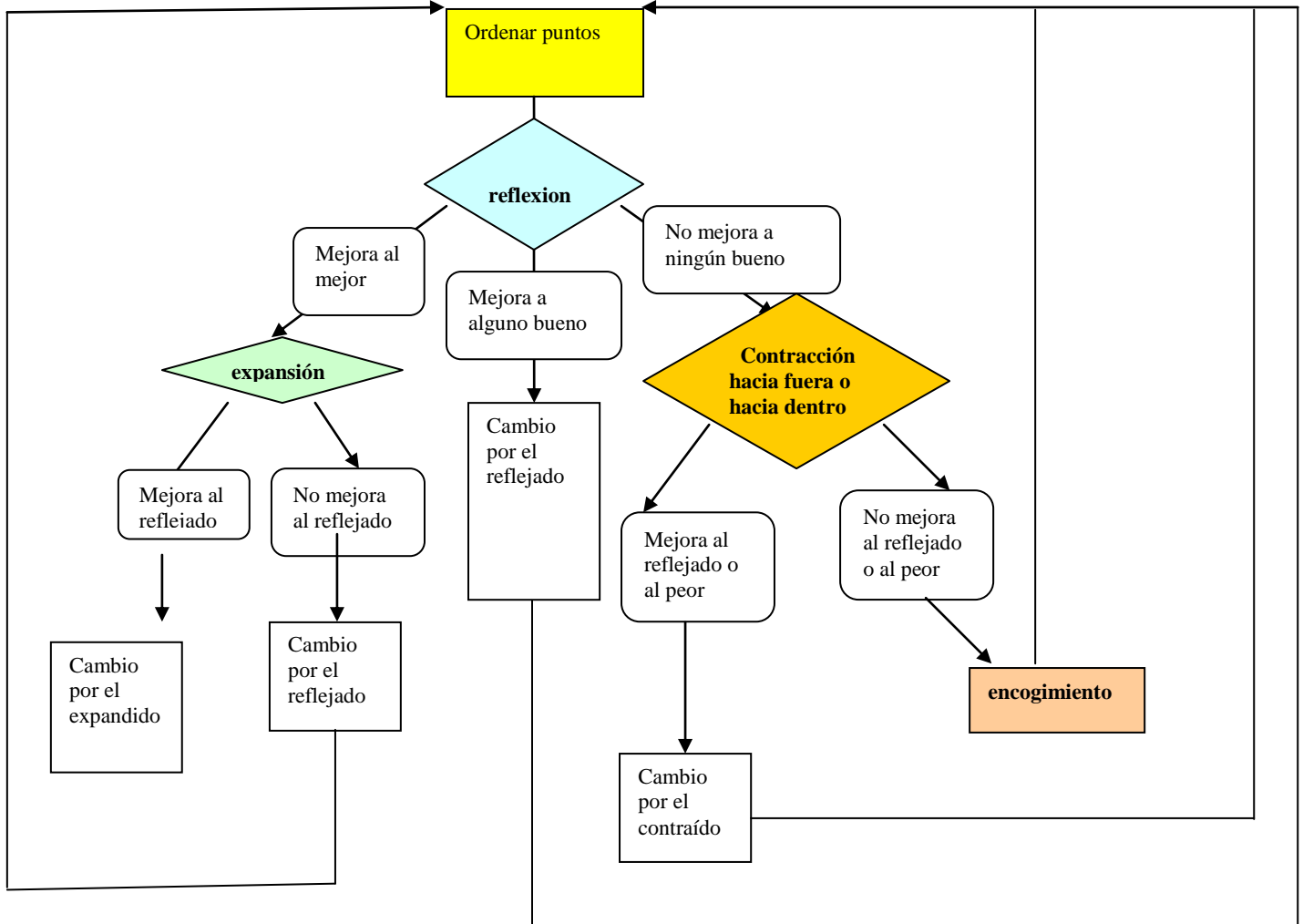


FIG. 1. Nelder-Mead simplices after a reflection and an expansion step. The original simplex is shown with a dashed line.



Un organigrama del método de Nelder-Mead es



### 3. EXPERIENCIAS COMPUTACIONALES

Dado que, a priori, puede ocurrir que el mejor valor permanezca estable durante varias iteraciones, hemos optado por ejecutar el algoritmo un número fijo de veces, es decir  $k=1$  hasta  $n$ -iter.

Hemos probado la metaheurística en diversas funciones no lineales de dimensiones que varían desde 2 hasta 6, algunas de ellas multimodales. Estas funciones también han sido utilizadas, por ejemplo, como funciones test para algoritmos Scatter Search en un trabajo de Laguna y Marti (2005), y para un algoritmo genético, con carácter estocástico, en un trabajo de Tu y Lu (2004).

Las pruebas han sido realizadas en un Pentium 4, 2.8 GHz. Las implementaciones de los algoritmos de este trabajo, así como las representaciones gráficas, han sido realizadas con Matlab 6.5.

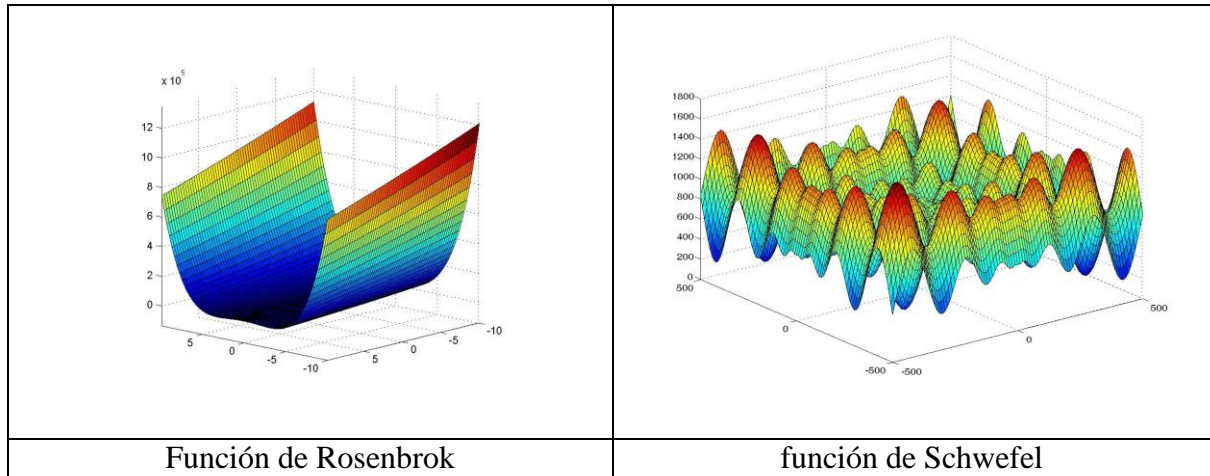
Mostramos en la tabla que viene a continuación las funciones y sus principales características.

Problema	dim	nombre	$x^*$ ( óptimo)	$f(x^*)$
$g_1(x)$	2	Space_paper	(-2.8362,-2.8362)	-130.8323
$g_2(x)$	2	B2	(0,0)	0
$g_3(x)$	3	De Joung	(0,0,0)	0
$g_4(x)$	4	Rastrigin 4	(0,0,0,0)	0
$g_5(x)$	4	Rosenbrok 4	(1,1,1,1)	-186.7309
$g_6(x)$	5	Michalewics 5		-4.687658
$g_7(x)$	6	Rosenbrok 6	(1,1,1,1,1,1)	0
$g_8(x)$	6	Schwefel 6		0

La función que denominamos  $g_1(x)$ , Space\_paper, es una función optimizada en el trabajo de Hsiao, Chuang, Jiang Chien (2005) donde se introduce la heurística Optimización gravitatoria.

Algunas de estas funciones objetivo han sido construidas de forma que su óptimo es conocido, pero que no puede ser hallado por procedimientos generales de búsqueda. Además, los problemas no pueden ser trivialmente resueltos por procedimientos de búsqueda que no exploren la estructura especial de cada función. Varias de estas funciones son multimodales con muchos óptimos locales y representan la clase de problemas con mayor dificultad, sobre todo cuando aumentamos la dimensión del espacio de soluciones. Dos ejemplos de estas últimas funciones son la

función de Schwefel y la función de Rosenbroks cuya representación gráfica en el caso de dos variables podemos contemplar en las figuras que vienen a continuación.



Los resultados obtenidos sólo con S.G.O. son muy buenos en algunos casos, lo que hace pensar en su gran capacidad para encontrar el óptimo global, siempre que se elijan adecuadamente los parámetros. Pero, como se ve a continuación, la metaheurística propuesta disminuye el error en todos los casos.

Funcion	Aprox. S.G.O.	Error S.G.O.	Aprox. Metaheur.	Error Metaheur.
$g_1(x)$	130.832322	-6.443289e-007	-130.832323	5.684341e-014
$g_2(x)$	3.388344e-006	3.388344e-006	0	0
$g_3(x)$	9.184064e-008	9.184064e-008	0	0
$g_4(x)$	3.960606e-004	3.960606e-004	0	0
$g_5(x)$	1.950390e-004	1.950390e-004	1.035379e-030	1.035379e-030
$g_6(x)$	-4.668336	1.932110e-002	-4.687658	1.790881e-007
$g_7(x)$	2.480493e-004	2.480493e-004	4.269709e-029	4.269709e-029
$g_8(x)$	-1.633879e-006	-1.633879e-006	-1.346024e-007	1.3460248e-007

#### 4. CONCLUSIONES

- El algoritmo S.G.O. es una buena herramienta para optimizar funciones de dimensión 2 (ver más ejemplos en Zapatero M.J. y otros 2009). Sin embargo, para mayores dimensiones se hace necesario realizar un estudio previo de los parámetros utilizados para obtener resultados aceptables. Este estudio es muy costoso aunque posible si se conoce el óptimo global a priori pero resulta muy difícil en un problema desconocido. Ésta es posiblemente la dificultad más seria del algoritmo S.G.O. Sin embargo, añadir el algoritmo Nelder-Mead como estrategia de intensificación hace que podamos obtener resultados muy buenos partiendo de aproximaciones menos buenas

obtenidas a partir de S.G.O. Así, el estudio inicial de los parámetros de S.G.O. no es tan importante para asegurar el éxito y el metaheurístico puede emplearse en problemas nuevos con garantías.

- El algoritmo Nelder-Mead busca óptimos locales, por tanto por sí sólo no se puede garantizar que encuentre el óptimo global. Sin embargo, si sus datos iniciales están cerca del óptimo global, i.e. son, por ejemplo, las aproximaciones ya encontradas por S.G.O., su efectividad se potencia en gran medida.
- La capacidad de no quedar atrapado en óptimos locales de S.G.O. y la capacidad de localizar el óptimo en “dominios pequeños” de Nelder-Mead se complementan perfectamente de forma que ninguno “se mete en la tarea” del otro.
- En definitiva, estamos ante una metaheurística muy prometedora que resuelve de forma satisfactoria los problemas planteados desde 2 dimensiones hasta 6. Queda abierta su evaluación en dimensiones mayores que 6.

## 5. APÉNDICE

Lista de funciones empleadas:

$$g_1 = x_1^4 - 16 \cdot x_1^2 + 0.5 \cdot x_1 + x_2^4 - 16 \cdot x_2^2 + 0.5 \cdot x_2 \quad \text{función del Space\_paper}$$

Dominio de optimización:  $[-15, 15] \times [-15, 15]$ , que denotaremos  $[-15, 15]^2$

$$g_2 = \left( \sum_{i=1}^5 i \cos((i+1)x_1 + i) \right) \left( \sum_{i=1}^5 i \cos((i+1)x_2 + i) \right), \quad \text{función B2}$$

Dominio de optimización:  $[-50, 100]^2$

$$g_3 = \sum_{i=1}^3 x_i^2 \quad \text{función De Joung}$$

Dominio de optimización:  $-2.56 \leq x_i \leq 5.12 \quad i = 1, 2, 3.$

$$g_4 = 418.9829n + \sum_{i=1}^n (-x_i \sin \sqrt{|x_i|}) \quad \text{función Rastrigin 4}$$

Dominio de optimización:  $x_i \in [0, \pi] \quad i = 1, \dots, 5$

$$g_5 = \sum_{i=1}^3 [100(x_{i+1} - x_i^2)^2 - (x_i - 1)^2] \quad \text{función de Rosenbrock 4}$$

Dominio de optimización:  $[-500, 500]^4$

$$g_6 = -\sum_{j=1}^5 \text{sen}(x_j) \left( \text{sen} \left( \frac{jx_j^2}{\pi} \right) \right)^{20}, \text{ función de Michalewics}$$

Dominio de optimización:  $x_i \in [0, \pi] \ i = 1, \dots, 5$

$$g_7 = \sum_{i=1}^5 [100(x_{i+1} - x_i^2)^2 - (x_i - 1)^2] \text{ función de Rosenbrocks } 6$$

Dominio de optimización:  $[-500, 500]^6$

$$g_8 = 2513.897322 + \sum_{i=1}^6 -x_i \sin \sqrt{|x_i|} \text{ función de Schwefel } 6$$

Dominio de optimización:  $[-30, 30]^2$

## 6. REFERENCIAS BIBLIOGRÁFICAS

- Hsiao, Chuang, Jiang and Chien (2005) "A Novel Optimization Algorithm: Space Gravitational Optimization". Systems, Man and Cybernetics, IEEE International Conference on Volume 3, Issue , 10-12 Oct. 2005 Page(s): 2323 - 2328 Vol. 3
- Lagarias J., Reeds J., Wright M. and Wright P.(1998) "Converge properties of the Nelder-Mead simplex Method in low dimensions". SIAM J. OPTIMIZATION, No 1, pp. 112-147.
- Laguna M. and Marti R. (2005) "Experimental Testing of Advanced Scatter Search Designs for Global Optimization of Multimodal Functions". Journal of Global Optimization Volume 33 , Issue 2 , Oct. 2005, pages: 235 – 255.
- Martínez González, J.L. (2001) "Optimización y ajuste de parámetros mediante el método simplex (Nelder-Mead)" 1ª Reunión de Usuarios de EcosimPro, UNED, Madrid 3-4 Mayp 2001, C19.
- Tu Z. and Lu Y. (2004). "A Robust Stochastic Genetic Algorithm (StGA) for Global Numerical Optimization". IEEE Transactions on Evolutionary Computation, vol. 8, no. 5, october 2004.
- Zapatero M.J., Pacheco J. y Alegre J. "Uso de la heurística "Optimización Gravitatoria" con funciones no lineales" *Rect@* 2009. Vol: Actas\_17 Issue: 1 Pages/record No.: 201.

- Zapatero M.J., Alegre J., y Pacheco J. ““Optimización gravitatoria” y “Optimización por enjambre de partículas”. Comportamiento en funciones no lineales ” XXVIII jornadas ASEPUMA. VI Encuentro Internacional. Anales de ASEPUMA nº 18 303.

#### Webs

- información sobre las funciones: [http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar\\_files/go.htm](http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/go.htm);  
<http://solon.cma.univie.ac.at/glopt.html>.