

ALGORITMO HÍBRIDO DEL SIMPLEX CON LA ESTRATEGIA DE OPTIMIZACIÓN DE FUERZA CENTRAL (CFO) APLICADO A LA SOLUCIÓN DE SISTEMAS DE ECUACIONES NO LINEALES

AUTORES:

José Reyes García Delgado  
Diego Fernando Corredor Velandia

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS  
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES  
BUCARAMANGA  
2013

ALGORITMO HÍBRIDO DEL SIMPLEX CON LA ESTRATEGIA DE OPTIMIZACIÓN DE FUERZA CENTRAL (CFO) APLICADO A LA SOLUCIÓN DE SISTEMAS DE ECUACIONES NO LINEALES

AUTORES:

José Reyes García Delgado  
Diego Fernando Corredor Velandia

TRABAJO DE GRADO PARA OPTAR AL TITULO DE INGENIERO ELECTRÓNICO

CARLOS RODRIGO CORREA CELY, PHD  
DIRECTOR DE TRABAJO DE GRADO

IVÁN MAURICIO AMAYA CONTRERAS, PHD(C)  
CODIRECTOR DE TRABAJO DE GRADO

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS  
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES  
BUCARAMANGA  
2013

## **AGRADECIMIENTOS DIEGO**

A mis padres, que con su cariño, apoyo, ejemplo y sacrificio, me dotaron de la fortaleza para reponerme a los inconvenientes propios del ascetismo y el valor para forjarme a mi mismo y a mis convicciones.

A Sergio, mi hermano, que me ha acompañado desde niño, ha sido mi confidente, compañero de juegos, cómplice, consejero y amigo.

A Julian, mi hermanito, que me hace sentir que el mundo es sólo suyo y mio si su voz me dice: "Te Quiero".

A Carolina, que siempre se las arregló para hacerme sentir el hombre más afortunado.

A José Reyes, Anjher Gustavo y Sergio Augusto, amigos incondicionales y guerreros de la vida. Quizás las personas más valiosas que un hombre puede encontrar en su camino.

A mis compañeros de aventuras, Alejandro, Nando, Gerson, Espitia, Susan, Adriana, Pollo, agradezco los buenos momentos y las sonrisas.

A Maradona, el D10s, cuyo enorme talento me concedió un nombre que me agrada.

A Cerati, quien me enseñó lo que es la música.

## **AGRADECIMIENTOS JOSE**

A mis abuelos, quiénes se me educaron y me cuidaron desde el momento en que quede totalmente solo, soy el resultado de su esfuerzo por convertirme en un buen hombre.

A mi tío Oscar y la señora Cecilia Mateus, quiénes sin interés alguno creyeron en mí y me apoyaron de tantas maneras que una página de agradecimientos no bastaría para nombrarlas. Este título universitario es de ellos.

A mi hermano Gonzalo, él es el motor de mi vida y la razón por la cual trato de ser mejor persona para convertirme en un buen ejemplo de hermano mayor y luchar por un futuro promisorio para ambos.

A Eduardo, mi amigo, confidente y compinche de toda la vida a quien le deseo lo mejor en cada uno de sus planes.

A Tavo y Diego, que me apoyaron y me escucharon en los momentos en que sentía desfallecer, grandes personas a quienes les deseo muchos éxitos.

A mi tía Libia, a Martha Probst, a Daniel León y su familia, muchas gracias por toda la colaboración que me han prestado, ojalá la vida les depare lo mejor a tan gratas personas.

A Wen, que me ha mostrado el lado amable y lindo de la vida y que con sonrisas me ha alegrado la última etapa de mi vivencia académica.

# Índice de contenido

INTRODUCCIÓN.....	13
1. OBJETIVOS.....	14
1.1 GENERAL.....	14
1.2 ESPECÍFICOS.....	14
2. FORMULACIÓN DEL PROBLEMA.....	14
3.MARCO TEÓRICO:.....	15
3.1 Central Force Optimization:.....	15
3.2 IMPLEMENTACIÓN DE CFO.....	19
3.3 Algoritmo Simplex Nelder – Mead.....	20
3.4 Implementación del simplex.....	21
3.5 Solución a sistemas de ecuaciones no lineales mediante teorema de raíces reales:.....	23
4. EVALUACIÓN DEL ALGORITMO CON FUNCIONES TÍPICAS.....	24
4.1 FUNCIONES DE PRUEBA:.....	24
4.1.1 Función 1: Schewefel:.....	24
4.1.2 Función 2: Griewank Modificado:.....	25
4.1.3 Función 3: Ackley Modificado:.....	25
4.1.4 Función 4: Rastrigin Modificado:.....	26
4.1.5 Función 5: Función de Paso:.....	27
4.1.6 Función 6: Esfera Modificada:.....	28
4.1.7 Función 7: Función de Rosenbrock:.....	29
4.1.8 Función 8: Función de Colville:.....	29
4.1.9 Función 9: Six-Hump Camel-Back:.....	30
4.1.10 Función 10: Función de Branin:.....	31
4.1.11 Función 11: Agujeros invertidos de Shekel:.....	31
4.1.12 Función 12: Keane's Bump modificado:.....	32
4.2 Algoritmo Híbrido Central Force Optimization y Simplex Nelder-Mead, CFOS:.....	33
4.3 Tabla de evaluaciones de funciones típicas.....	37
5. ALGORITMO HÍBRIDO EN SISTEMAS DE ECUACIONES NO LINEALES:.....	41
5.1 SISTEMAS DE ECUACIONES NO LINEALES:.....	41
5.1.1 Sistema 1.....	41
5.1.2 Sistema 2.....	42
5.1.3 Sistema 3.....	42
5.2 RESULTADOS EVALUACIONES SISTEMAS NO LINEALES.....	43
5.3 CIRCUITOS ELECTRÓNICOS MODELADOS POR MEDIO DE SISTEMAS DE ECUACIONES NO LINEALES.....	44
5.3.1 Circuito 1.....	44
5.3.2 Circuito 2.....	46
5.3.3 Circuito 3.....	48
5.3.4 Circuito 4.....	50
6. CONCLUSIONES.....	53
BIBLIOGRAFIA.....	55

## Lista de figuras

Figura 1: Sonatas en el espacio de búsqueda.....	17
Figura 2: Diagrama de flujo algoritmo simplex.....	21
Figura 3: Función de Schewefel.....	24
Figura 4: Función de Ackley.....	25
Figura 5: Función de Rastrigin.....	26
Figura 6: Función de Paso.....	27
Figura 7: Función de esfera modificada.....	27
Figura 8: Función de Rosenbrock.....	28
Figura 9: Función de Six-Hump.....	29
Figura 10: Función de Branin.....	30
Figura 11: Función de agujeros invertidos de Shekel.....	31
Figura 12: Función de Keane.....	32
Figura 13: Diagrama de flujo híbrido simplex-CFO de Mahmoud.....	33
Figura 15: Circuito electrónico con un diodo.....	44
Figura 16: Circuito electrónico con dos diodo.....	46
Figura 17: Espejo de corriente de Widlar modificado.....	48
Figura 18: Circuito con MOSFET y diodo.....	51
Figura 19: Circuito con MOSFET y diodo en DC.....	51

## Índice de tablas

Tabla 1: Resultados para funciones de prueba CFO Formato 2007 junto a CFO básico implementado para este trabajo .....	37
Tabla 2: Resultados para funciones de prueba híbrido.....	39
Tabla 3: Especificaciones equipo de cómputo utilizado en este trabajo.....	40
Tabla 4: Resultados híbrido para sistemas de ecuaciones no lineales.....	42
Tabla 5: Resultados híbrido para circuito electrónico de prueba uno descrito mediante un sistema de ecuaciones no lineales contrastado con otras implementaciones.....	44
Tabla 6: Parámetros de entrada circuito electrónico de prueba dos.....	46
Tabla 7: Resultados híbrido para circuito electrónico de prueba dos descrito mediante un sistema de ecuaciones no lineales contrastado con otras implementaciones.....	46
Tabla 8: Parámetros de entrada circuito electrónico de prueba tres.....	48
Tabla 9: Resultados híbrido para circuito electrónico de prueba tres descrito mediante un sistema de ecuaciones no lineales contrastado con otras implementaciones.....	49
Tabla 10: Parámetros de entrada circuito electrónico de prueba cuatro.....	51
Tabla 11: Resultados híbrido para circuito electrónico de prueba cuatro descrito mediante un sistema de ecuaciones no lineales contrastado con otras implementaciones.....	52

## Índice de Anexos

Anexo A: Instrucciones de uso CFOS.....	56
Anexo B: Pseudocódigo para implementación de CFO.....	58

## RESUMEN

**TITULO:** ALGORITMO HÍBRIDO DEL SIMPLEX CON LA ESTRATEGIA DE OPTIMIZACIÓN DE FUERZA CENTRAL (CFO) APLICADO A LA SOLUCIÓN DE SISTEMAS DE ECUACIONES NO LINEALES.\*

**AUTORES:** José Reyes García Delgado, Diego Fernando Corredor Velandia.†

### **PALABRAS CLAVE:**

Optimización, Optimización por Fuerza Central (CFO), Simplex, Ecuaciones no lineales, Circuito no Lineal, híbrido.

### **CONTENIDO:**

Este documento tiene como objetivo proponer una estrategia que combine los métodos de Optimización de Fuerza central (CFO) y el método de Simplex y presentarlo como alternativa de solución a modelos matemáticos descritos con sistemas de ecuaciones no lineales, utilizando el teorema de las raíces reales. Para justificar el desarrollo de la estrategia híbrida y su funcionamiento, se evaluó el método utilizando funciones típicas de prueba. Adicionalmente, se aplica el método para resolver sistemas descritos con ecuaciones no lineales, como el espejo de corriente de Widlar o circuitos que contienen diodos y transistores. Finalmente, se contrastan los resultados con los métodos de Optimización por Enjambre de Partículas (PSO), Optimización por Enjambre de Partículas Unificado (UPSO), y Newton Raphson Multidimensional (NRM) implementados en trabajos previos al interior del grupo CEMOS.

Se encontró que el método híbrido es una alternativa viable para abordar los problemas representados con sistemas de ecuaciones no lineales, donde la habilidad de mapeo de CFO mejoró el desempeño del Simplex, al depender éste en menor medida de un punto inicial cercano al óptimo. Finalmente, queda abierta la posibilidad de tomar el algoritmo e implementarlo en diferentes soluciones de problemas de ingeniería descritos por ecuaciones no lineales distintos a los circuitos electrónicos.

---

\*Proyecto de grado

†Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Carlos Rodrigo Correa. Co director: Iván Mauricio Amaya.

## ABSTRACT

**TITLE:** HYBRID ALGORITHM OF SIMPLEX WITH THE STRATEGY OF CENTRAL FORCE OPTIMIZATION (CFO) APPLIED TO SOLVING NON-LINEAR SYSTEMS.\*

**AUTHORS:** José Reyes García Delgado, Diego Fernando Corredor Velandia.†

**KEY WORDS:** Optimization, Central Force Optimization (CFO), Simplex, non-linear equations, non-linear circuit, hybrid.

### CONTENT:

This document has as target to propose a new strategy that integrates the methods of Simplex and Central Force Optimization, CFO, and introduce it as an alternative to solving mathematics models described by non-linear equations systems, using the real roots theorem. In order to validate the hybrid strategy and its behavior, the algorithm was evaluated using typical functions of probe. Additionally, the hybrid was applied to solve circuits described with non-linear equations, like Widlar current mirror or circuits with diodes and transistors. Finally, the hybrid's results are compared with other optimization methods like Particle Swarm Optimization, PSO, Unified Particle Swarm Optimization, UPSO and Multidimensional Newton Raphson, NRM, developed in previous works at the group CEMOS.

It was found that the hybrid method is a valid alternative to solve some problems represented by non-linear equations systems, and that the ability of CFO's mapping improved the behavior of the Simplex method because its convergence began to independize of the position of the initial point (the simplex has a good answer if the initial point is near to the optimal). Finally, there is a big chance to apply for future works the algorithm, implementing it to solve other engineering problems modeled by non-linear equations different of electronic circuits.

---

\*Undergraduate thesis.

†Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Carlos Rodrigo Correa. Co director: Iván Mauricio Amaya.

## INTRODUCCIÓN

En matemáticas, la optimización consiste en encontrar los máximos o mínimos de una función específica. Generalmente, los programas computacionales lidian con problemas de optimización de la vida real, y su correcta optimización permite a los diferentes sectores minimizar diferentes variables como costos o emisión de contaminantes. Dos estrategias de optimización existentes en la actualidad son la de Fuerza Central, CFO, y la programación lineal usando el algoritmo Simplex.

CFO es un algoritmo basado en una analogía de las leyes gravitacionales de Newton, adaptadas para crear un método numérico en el cual se distribuyen unas sondas que “vuelan” a través del espacio de búsqueda hasta encontrar el punto óptimo. Por otra parte, el método Simplex hace uso de la propiedad de que la solución óptima de un problema de programación lineal se encuentra en un vértice o frontera del dominio de puntos factibles (esto último en casos muy especiales), por lo cual, la búsqueda secuencial del algoritmo se basa en la evaluación progresiva de estos vértices, hasta encontrar el óptimo.

En las secciones que se presentarán a continuación, se expondrán las bases teóricas que sustentan los algoritmos de CFO y Simplex, se explicará el funcionamiento de la estrategia híbrida desarrollada, se mostrarán los resultados obtenidos al evaluar CFO y CFO-Simplex con funciones típicas de prueba y se describirá el proceso para aplicar el algoritmo híbrido a la solución de sistemas que se describen con ecuaciones no lineales.

## 1. OBJETIVOS<sup>1</sup>

### 1.1 GENERAL

- Desarrollar un algoritmo híbrido entre los métodos Simplex y Central Force Optimization, CFO, y aplicarlo a la solución de sistemas de ecuaciones no lineales (páginas 31 a 49).

- 

### 1.2 ESPECÍFICOS

- Implementar un algoritmo híbrido (Simplex y CFO) en la plataforma MATLAB® u otro software computacional (páginas 31 a 34).
- Realizar pruebas al algoritmo con funciones de evaluación típicas obtenidas a través de la revisión bibliográfica (páginas 39 a 40).
- Aplicar el algoritmo híbrido desarrollado a la solución de sistemas de ecuaciones no lineales y contrastar resultados con los algoritmos de PSO, UPSO, NRM, CFO y el software PSPICE (páginas 40 a 50).

## 2. FORMULACIÓN DEL PROBLEMA

El uso del algoritmo Simplex ha servido para resolver muchos problemas de optimización durante los últimos 40 años[1][2]. Sin embargo, en problemas de múltiple dimensión, por ejemplo, puede llegar a perder parte de su efectividad, por lo cual se han creado algunas variantes como las expuestas en [3]. De igual manera, el método de Fuerza Central puede llegar a tardar en encontrar la respuesta a determinado problema, debido al intervalo especificado o a la naturaleza misma de la función a optimizar.

---

<sup>1</sup> Esto constituye un aporte intelectual consensuado entre los estudiantes García y Corredor junto con el profesor Rodrigo Correa e Iván Amaya.

Dado que no existe una estrategia infalible para resolver cada problema, el uso de una u otra puede llevar a no encontrar una respuesta o a tardar más de lo necesario. Esto, en la industria, puede representar dinero invertido, que conlleva a un incremento en el costo del producto o servicio final.

Por lo anterior se plantea desarrollar un algoritmo híbrido que involucre ambas estrategias con el objetivo de enlazar algunas de las ventajas de cada una y generar un algoritmo que llegue más eficientemente a la respuesta esperada.

### **3.MARCO TEÓRICO:**

#### **3.1 Central Force Optimization:**

La optimización de Fuerza Central, CFO por sus siglas en inglés, es una estrategia de optimización que al igual que el Ant Colony Optimization (ACO), o el Particle Swarm Optimization (PSO), se basa en una analogía de la naturaleza a la ciencia de la computación [1], [5] para lograr encontrar el máximo de una función objetivo utilizando las leyes de la dinámica de Newton. CFO fue introducido por Ricardo Formato como un algoritmo metaheurístico que podía ser aplicado a distintos tipos de problemas de optimización [6]. Desde su inicio, y a pesar de no tener gran fundamentación teórica, pues se esperaba fuera mejorando por trabajo experimental, CFO se mostraba como un algoritmo muy prometedor en optimización ya que se podía aplicar a diversos problemas con muy poca modificación del planteado originalmente y el poco tiempo que lleva en uso ha dado como cierto lo que se esperaba de él, encontrando aplicaciones electromagnéticas como el mejoramiento del rendimiento y diseño de antenas [7] [8] [9], ecualizadores, síntesis de arreglos lineales para distribución de elementos electromagnéticos [6] y entrenamiento de redes neuronales [10] .

La analogía de la naturaleza que utiliza el CFO consiste en el uso de las tres ecuaciones de Newton aplicadas al campo computacional:

$$F = \gamma \frac{m_1 m_2}{r^2} \quad (1)$$

$$a = -\gamma \frac{m_2}{r^2} \hat{r} \quad (2)$$

$$R(t + \Delta t) = R_0 + V_0 \Delta t + \frac{1}{2} a \Delta t^2 \quad (3)$$

La ecuación (1) establece la relación de la fuerza que ejercen entre sí dos objetos con masas  $m_1$  y  $m_2$  separados una distancia  $r$  influenciados por una aceleración gravitacional  $\gamma$ , mientras que la ecuación (2) establece la magnitud y dirección vectorial de la aceleración de un objeto con masa  $m_2$  hacia otro de masa  $m_1$ ; la ecuación (3) es la ecuación cinemática de la posición y establece el cambio de ésta en función de su posición, velocidad inicial y aceleración a la cual está sometido. Las ecuaciones (2) y (3) son denominadas ecuaciones de “movimiento” [11], [6].

Para comprender el uso de estas ecuaciones, Formato da como ejemplo la atracción gravitacional de los planetas: Si nos ubicáramos en el sistema solar sin conocerlo y quisiéramos encontrar el planeta más grande podríamos enviar un grupo de sondas,  $N_p$ , a lo largo de un espacio determinado, donde consideremos puede estar e ir monitorizando su movimiento en intervalos definidos de tiempo. Es de esperar que la mayoría de sondas vayan hacia el objeto con mayor atracción gravitacional que sería de acuerdo a las leyes de Newton el de mayor tamaño por lo cual las sondas después de un tiempo estarían agrupadas alrededor del mismo [6]. En este orden de ideas, CFO generaliza las ecuaciones de movimiento, teniendo en cuenta que cada sonda tendrá una aceleración  $a$ , una masa  $M$  y una posición  $R$  así:

$$\vec{a}_{j-1}^p = \frac{G \sum_{k=1, k \neq p}^{N_p} u(F) \cdot (F)^\alpha \vec{R}_{j-1}^k - \vec{R}_{j-1}^p}{(\vec{R}_{j-1}^k - \vec{R}_{j-1}^p)^\beta} \quad (4)$$

$$F = M_{j-1}^k - M_{j-1}^p \quad (5)$$

$$\vec{R}_j^p = \vec{R}_{j-1}^p + \frac{1}{2} \vec{a}_{j-1}^p \Delta t^2 \quad (6)$$

Se observan varios cambios en las ecuaciones de aceleración y posición usadas por el

algoritmo con respecto a las planteadas por Newton. A continuación señalamos estas diferencias y explicamos que significan cada uno de los nuevos parámetros:

- En (4) se hace uso de la función escalón,  $u(F)$ , esta función tiene el siguiente comportamiento:

$$u(F) = \begin{cases} 0 & \text{si } F < 0 \\ 1 & \text{si } F \geq 0 \end{cases} \quad (7)$$

El objetivo de utilizarla es evitar “masas negativas” dado que se perdería la analogía a las leyes de Newton y se generarían fuerzas repulsivas en lugar de atractivas [6], estas pueden surgir debido a la diferencia que implica (5); con el escalón se evita que este tipo de resultados ocurran.

- Las “masas” en el algoritmo están dadas por la evaluación de la función en la posición de la sonda, a esto se le llama *fitness* y se define como:  $M = f(x_i)$  donde  $x_i$  representa la posición de cada sonda dada por (6).
- $j$  corresponde al paso de tiempo, representa las iteraciones que realizará el algoritmo y es el análogo a  $\Delta t$  mientras que  $p$  y  $G$  representan la sonda que se está analizando y la aceleración gravitacional respectivamente.
- Las constantes  $\alpha$  y  $\beta$  en (4) no tienen un sentido físico como tal, sino que se derivan del análisis matemático realizado a la analogía y, generalmente, tiene un valor entre 1 y 3 [6], [12], mientras  $G$  representa la aceleración de la gravedad.
- En (6), que representa la ecuación de posición de las sondas, se ha eliminado el término de la velocidad y el paso de tiempo se ha igualado a 1 para simplificar el algoritmo [6]. Esta posición es multidimensional y se trata como un vector con las diferentes coordenadas de la sonda.

CFO está diseñado para buscar una solución óptima en un espacio de búsqueda determinado por el usuario, este se define como [13]:

$$\Omega = x_i^{min} < x_i < x_i^{max}, 1 \leq i \leq N_d \quad (8)$$

$\Omega$  representa el espacio de búsqueda,  $x_i$  son las variables de decisión y definen el intervalo donde se “moverán” las sondas mientras  $N_d$  representa el número de dimensiones de la función objetivo.

En ciertas iteraciones del algoritmo es posible que algunas sondas se salgan del espacio de búsqueda tal como lo muestra la figura 1.

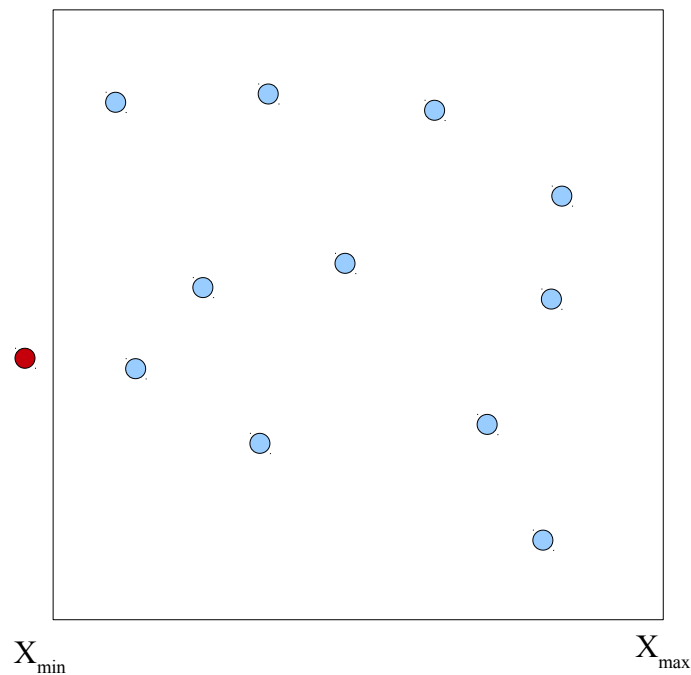


Figura 1: Sondas en el espacio de búsqueda

De acuerdo con la Figura 1, se tiene un espacio de decisión bidimensional con límites  $X_{min}$  y  $X_{max}$  respectivamente, las sondas azules se encuentran dentro del espacio determinado mientras que las rojas por acción de la suma de fuerzas de gravitación han quedado fuera del mismo, para regresarlas Formato planteó inicialmente dos formas [4] :

- Regreso a un punto medio entre la posición previa de la sonda y el límite del espacio de búsqueda:

$$R(p, i, j) = x_{\min}(i) + \frac{1}{2}(R(p, i, j-1) - x_{\min}(i)), \text{ si } R(p, i, j) < x_{\min}(i) \quad (9)$$

$$R(p, i, j) = x_{\max}(i) - \frac{1}{2}(x_{\max}(i) - R(p, i, j-1)), \text{ si } R(p, i, j) > x_{\max}(i) \quad (10)$$

- Regreso a un punto aleatorio en el espacio de búsqueda.

Finalmente existen dos criterios de parada planteados [6] [14]:

- Por número de iteraciones o pasos de tiempo,  $j$ .
- La distancia promedio normalizada entre la sonda con el mejor *fitness*,  $P_{bf}$ , y las demás sondas en cada paso de tiempo, esta distancia promedio se plantea como:

$$D_{avg} = \frac{1}{L(N_p - 1)} \sum_{p=1}^{N_p} \sqrt{\left( \sum_{p=1}^{N_p} x_i^{p,j} - x_i^{P_{bf},j} \right)} \quad (11)$$

donde:

$$L = \sqrt{\sum_{i=1}^{N_d} (x_i^{\max} - x_i^{\min})^2} \quad (12)$$

Una vez que (11) llegue a una tolerancia definida por el usuario el algoritmo se detendrá.

### 3.2 IMPLEMENTACIÓN DE CFO:

En la primera versión de CFO, Formato planteó un pseudocódigo sencillo para la implementación del algoritmo. En trabajos posteriores incluyó nuevas modificaciones al algoritmo con el objetivo de hacerlo más eficiente y mejorar su convergencia [13] [15]. Se destacan el *factor de reposicionamiento*,  $F_{rep}$ , el paso de cambio del *factor de*

*reposicionamiento*,  $\Delta F_{rep}$ , el punto de intersección de las sondas al distribuirse uniformemente a lo largo de los ejes dado por un factor  $\gamma$  y la *adaptación del espacio de búsqueda*. Un pseudocódigo que muestra la lógica de CFO se plantea en el anexo 2.

### 3.3 Algoritmo Simplex Nelder – Mead

Simplex es un algoritmo basado en programación lineal, desarrollado inicialmente por George B. Dantzig en 1955. Su objetivo es buscar una región de interés en la cual sea posible encontrar la solución del problema a optimizar [16]. Esta región es delimitada por la intersección de las rectas construidas a partir de las constantes lineales de condición impuestas por el problema. Su uso es muy extendido en casos de optimización que usan programación lineal y tiene algunas variantes, como el Nelder-Mead [3], [17], el concurrente, o el algoritmo genético del Simplex [18].

Un Simplex es una figura geométrica de N+1 vértices y N dimensiones que encierra la región de interés [17]. A manera de ejemplo, para dos dimensiones, el Simplex es un triángulo, mientras que para tres es un tetraedro, no necesariamente regular, que forma un volumen de N dimensiones en el cual el algoritmo debe trabajar.

El método Nelder-Mead [1], consiste en la búsqueda de un mínimo mediante la creación de un Simplex, que es una figura geométrica que consta de N dimensiones, y tiene N+1 vértices,  $x_0, x_1, x_2 \dots x_n$ . Si un punto cualquiera del simplex es tomado como el origen, los demás puntos definen las direcciones del vector que abarca las N dimensiones del espacio.

El simplex consta de los siguientes movimientos:

- Reflexión: Se denomina  $x_{ref}$  a la reflexión de  $x_w$ , cuyas coordenadas se definen como:

$$x_{ref} = (1 + \alpha) \cdot \bar{x} - \alpha \cdot x_w \quad (13)$$

- Expansión: La expansión de  $x_{ref}$  se define como:

$$x_{exp} = (1 - \gamma) \cdot \bar{x} + \gamma x_{ref} \quad (14)$$

- Contracción: La contracción de  $x_w$ ,  $x_{con}$ , se define como:

$$x_{con} = (1 - \beta) \cdot \bar{x} - \beta x_w \quad (15)$$

Las constantes  $\alpha$ ,  $\beta$  y  $\gamma$  definen el factor por el cual el volumen del simplex cambia debido a las operaciones mencionadas,  $x_w$  representa el peor punto evaluado en la función objetivo y

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \text{ representa el centroide de los } n+1 \text{ puntos utilizados [1] [17]}$$

Inicialmente el algoritmo refleja el peor punto, si  $x_b < x_{ref} < x_w$ , siendo  $x_b$  el mejor punto, se reemplaza  $x_w$  por  $x_{ref}$ . Si  $x_{ref} < x_b$  se expande el simplex mediante (14). Si  $x_{exp} < x_b$  reemplazamos  $x_w$  por  $x_{exp}$  y volvemos a iniciar, en caso contrario hemos obtenido una expansión errónea y debemos reemplazar  $x_w$  por  $x_{ref}$  y reiniciar el proceso. La contracción dada por (15) se realiza si el peor punto es al menos tan bueno como el punto reflejado, en caso de que  $x_{con} > \min(x_b, x_{ref})$  el punto contraído debe ser rechazado. Para todas las contracciones rechazadas reemplazamos todos los vértices de la siguiente manera:

$$x_i = \frac{x_i - x_b}{2} \quad (16)$$

El criterio de parada para el algoritmo se define mediante una tolerancia definida por el usuario y que se compara con el valor dado por:

$$tol \leq \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}} \quad (17)$$

### 3.4 Implementación del simplex:

La implementación del simplex es sencilla, y dada su antigüedad existe bastante información sobre el mismo, resumimos en la figura 2 y el procedimiento para su implementación.

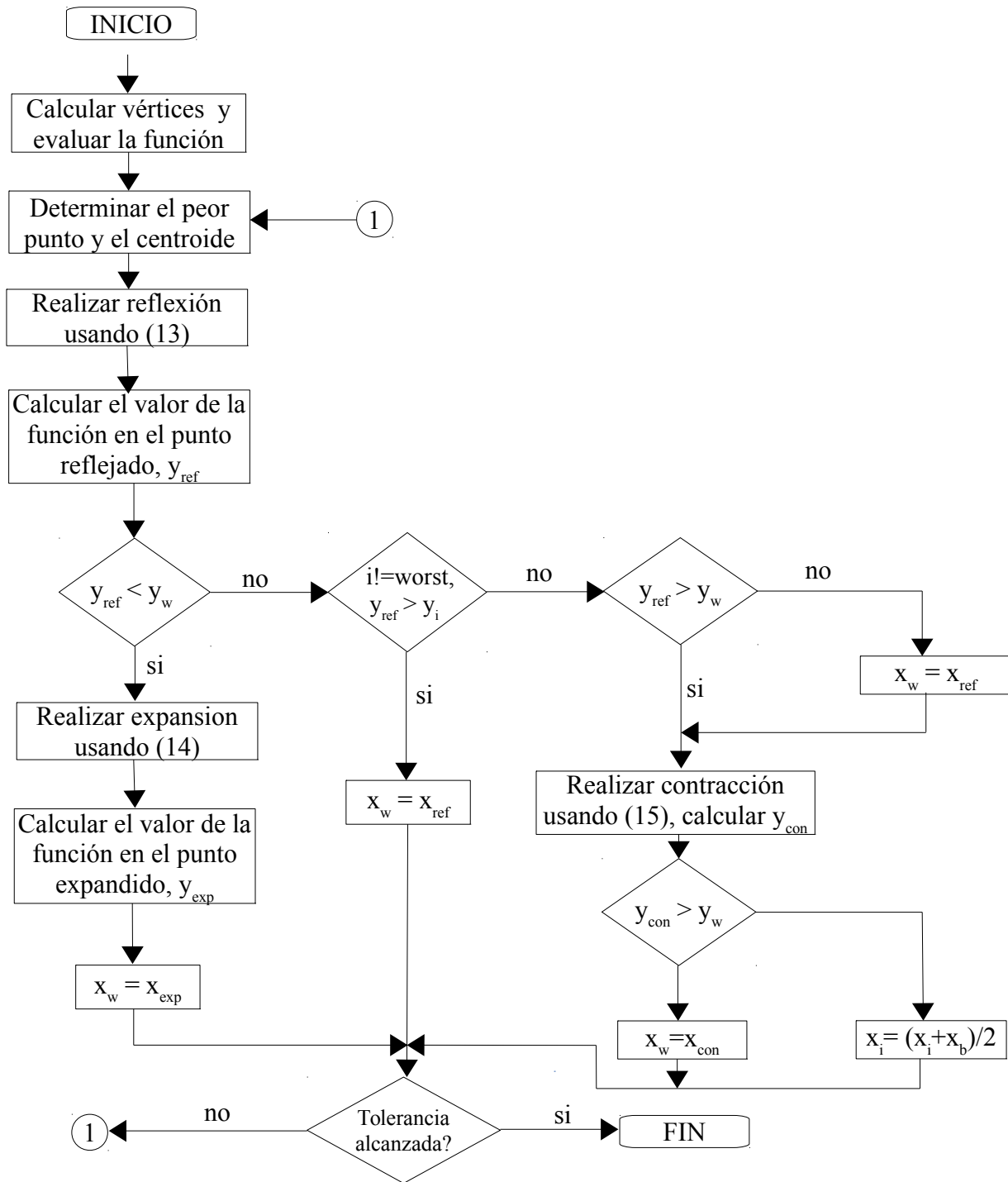


Figura 2: Diagrama de flujo algoritmo simplex

### 3.5 Solución a sistemas de ecuaciones no lineales mediante teorema de raíces reales:

La solución a problemas de diversas aplicaciones como hallar parámetros específicos de circuitos electrónicos con diodos o transistores se basa en el planteamiento de un sistema de ecuaciones no lineales de  $n \times n$  dimensiones, donde  $n$  es el número de variables de decisión a trabajar. Es posible dar solución a este sistema de ecuaciones mediante el uso del teorema de raíces reales y el criterio de optimización; para ello se parte del planteamiento del sistema de la siguiente manera [4]:

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ f_3(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ &\vdots \\ &\vdots \\ f_m(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \quad (18)$$

La ecuación (18) es importante para lograr adaptar el concepto de función objetivo a la solución de problemas mediante el uso de algoritmos de optimización. De acuerdo a investigaciones al interior del grupo CEMOS de la Universidad Industrial de Santander [4], [19], [20] es posible expresar la función objetivo de la siguiente manera:

$$\begin{aligned} f_{obj}(x) &= \sum_{i=1}^n (f(x_i))^2 \\ x &= (x_1, x_2, \dots, x_n) \end{aligned} \quad (19)$$

En (19) vemos que la función objetivo siempre será descrita como la suma de las funciones que describen el sistema de ecuaciones al cuadrado, lo cual nos garantiza que  $f_{obj}$  siempre será mayor a 0 y nos permite encontrar un mínimo  $a$  tal que satisfaga (18) y sea al mismo tiempo solución del sistema mediante la minimización de  $f_{obj}$  ya que el sistema estará acotado en el espacio de decisión  $[0, \infty)$ .

Finalmente es necesario plantear un subconjunto  $X$  de  $\mathbb{R}^n$  cuyo dominio contiene a cada

función  $f_i$ , entonces tendremos que  $x \in X$  es un posible subconjunto de solución al sistema. Una explicación mas detallada y demostración del teorema puede ser encontrada en [4], [19].

Con base en lo anterior, se puede definir un algoritmo de transformación del problema de la siguiente manera:

1. Entrada de un sistema de ecuaciones lineales como se muestra en (18) y definición de  $X$

2. Construir  $f_{obj}$  mediante (19) sobre  $x$ .

3. Sea  $a \in x$  es un mínimo para  $f_{obj}$ . Si  $f_{obj}(a) = 0$  entonces  $a$  satisface (18), de lo contrario, no existe solución en  $x$ .

#### **4. EVALUACIÓN DEL ALGORITMO CON FUNCIONES TIPICAS.**

##### **4.1 FUNCIONES DE PRUEBA:**

En este aparte se describen las funciones objetivo utilizadas para evaluar y validar inicialmente el algoritmo:

###### **4.1.1 Función 1: Schewefel:**

Se trata de una función de 30 dimensiones definida por (20):

$$f(x) = \sum_{i=1}^{30} x_i \sin \sqrt{|x_i|} \quad (20)$$

La función tiene un máximo de 12569.5 en  $x_i = 420.9687$  para  $i = 1, \dots, 30$ . El rango de búsqueda será  $-500 \leq x_i \leq 500$ . Se anexa el dibujo de la función para su variante en 2D en la Figura 3: Función de Schewefel.

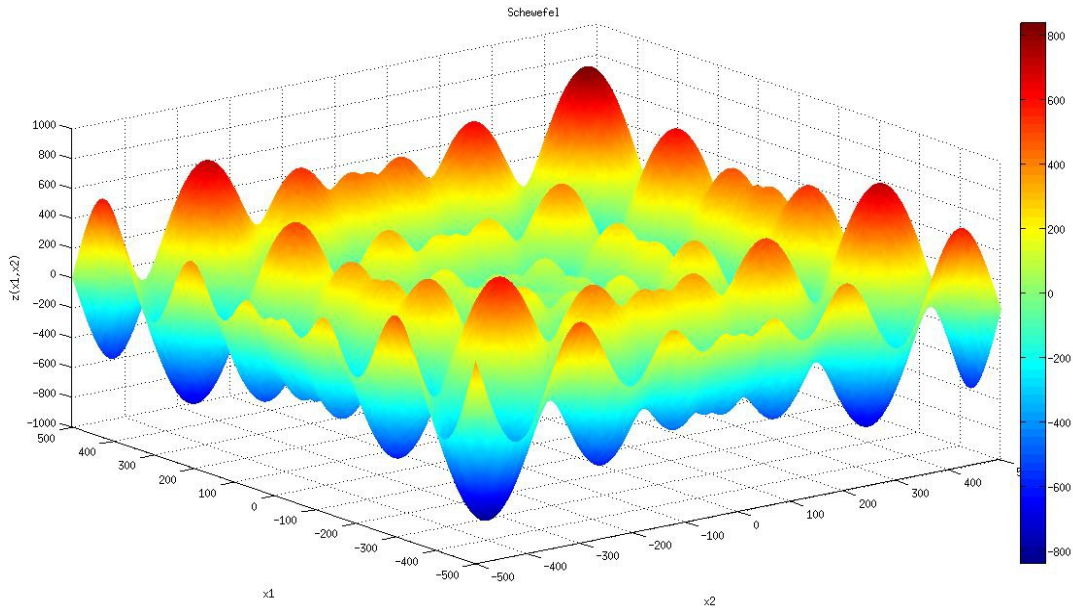


Figura 3: Función de Schewefel

#### 4.1.2 Función 2: Griewank Modificado:

Se trata de una función de 30 dimensiones definida por (21):

$$f(x) = \frac{-1}{4000} \sum_{i=1}^{30} (x_i - x_0)^2 + \prod_{i=1}^{30} \left\{ \frac{(x_i - x_0)}{\sqrt{i}} \right\} - 1 ; x_0 = 75.123 \quad (21)$$

La función tiene un máximo de 0 en  $x_i = x_0$  para  $i = 1, \dots, 30$ . El rango de búsqueda será  $-600 \leq x_i \leq 600$ .

#### 4.1.3 Función 3: Ackley Modificado:

Se trata de una función de 30 dimensiones definida por:

$$f(x) = 20 \exp \cdot \left( -0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} (x_i - x_0)} \right) + \exp \cdot \left( \frac{1}{30} \sum_{i=1}^{30} \cos \{ 2 \pi (x_i - x_0) \} \right) - 20 - e \quad (22)$$

con  $x_0 = 4.321$

La función tiene un máximo de 0 en  $x_i = x_0$  para  $i = 1, \dots, 30$ . El rango de búsqueda será  $-32 \leq x_i \leq 32$ . Se anexa el dibujo de la función para su variante en 2D en la Figura 4: Función de Ackley.

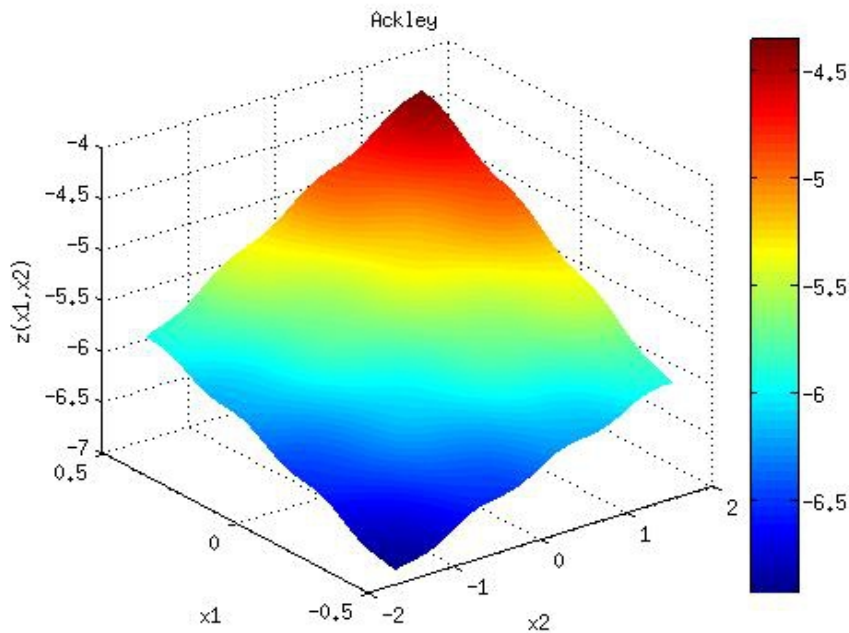


Figura 4: Función de Ackley

#### 4.1.4 Función 4: Rastrigin Modificado:

Se trata de una función de 30 dimensiones definida por:

$$f(x) = -\sum_{i=1}^{30} [y_i^2 - 10 \cos(2\pi y_i) + 10], \quad y_i = x_i - x_0 \quad (23)$$

$$\text{con } x_0 = 1.123$$

La función tiene un máximo de 0 en  $x_i = x_0$  para  $i = 1, \dots, 30$ . Posee muchos máximos locales cerca del valor máximo lo cual la hace una función difícil de optimizar. El rango de búsqueda será  $-5.12 \leq x_i \leq 5.12$ . Se anexa el dibujo de la función para su variante en 2D en la Figura 5: Función de Rastrigin.

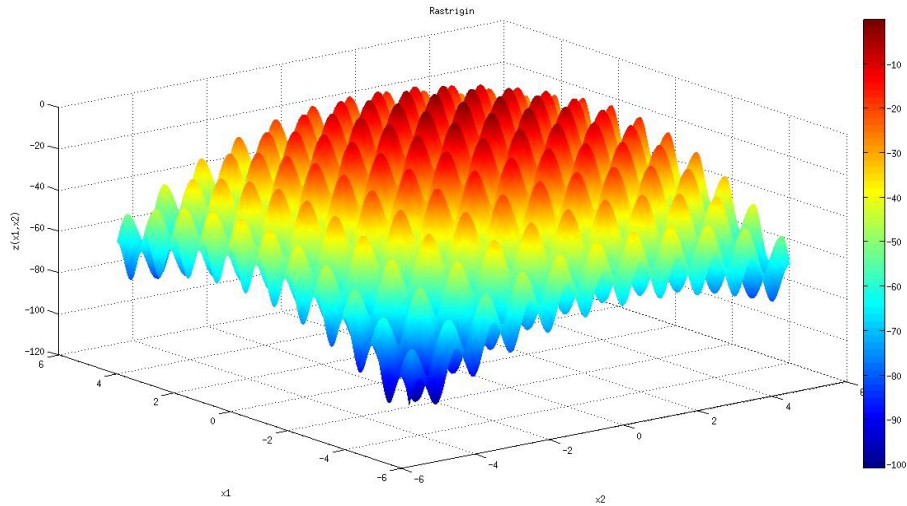


Figura 5: Función de Rastrigin

#### 4.1.5 Función 5: Función de Paso:

Se trata de una función de 30 dimensiones definida por:

$$f(x) = -\sum_{i=1}^{30} (\text{floor}[x_i - x_0 + 0.5])^2 \quad (24)$$

$$\text{con } x_0 = 75.123$$

La función tiene un máximo de 0 en  $x_i = x_0$  para  $i=1, \dots, 30$ . El rango de búsqueda será  $-100 \leq x_i \leq 100$ . Se anexa el dibujo de la función para su variante en 2D cerca de la vecindad del máximo en la Figura 6: Función de Paso.

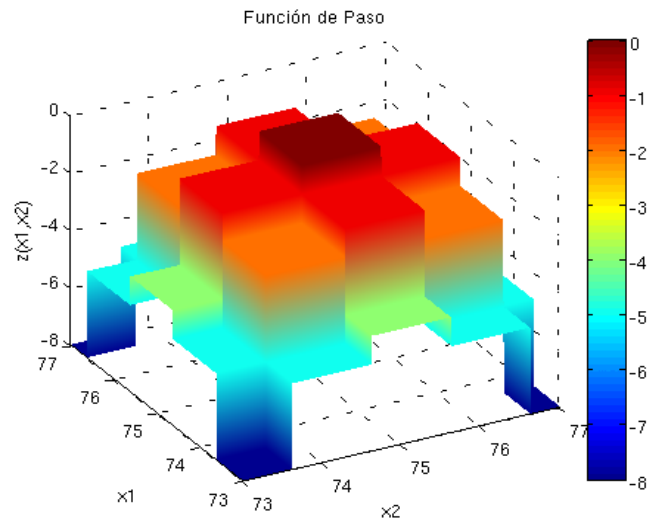


Figura 6: Función de Paso

#### 4.1.6 Función 6: Esfera Modificada:

Se trata de una función de 30 dimensiones definida por (25):

$$f(x) = -\sum_{i=1}^{30} (x_i - x_0)^2 \quad \text{con } x_0 = 75.123 \quad (25)$$

La función tiene un máximo de 0 en  $x_i = x_0$  para  $i = 1, \dots, 30$ . El rango de búsqueda será  $-100 \leq x_i \leq 100$ . Se anexa el dibujo de la función para su variante en 2D en la Figura 7: Función de esfera modificada.

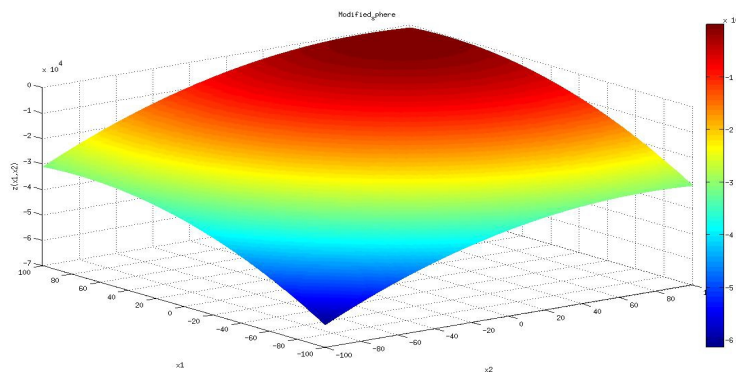


Figura 7: Función de esfera modificada

#### 4.1.7 Función 7: Función de Rosenbrock:

Se trata de una función de 30 dimensiones definida por (26):

$$f(x) = -\sum_{i=1}^{30} [100\{(x_{i+1}-x_0)-(x_i-x_0)^2\}^2 + \{(x_i-x_0)-1\}^2] \quad (26)$$

$$\text{con } x_0 = 25.123$$

La función tiene un máximo de 0 en  $x_i = 26.123$  para  $i = 1, \dots, 30$ . El rango de búsqueda será  $-30 \leq x_i \leq 30$ . Se anexa el dibujo de la función para su variante en 2D en la Figura 8: Función de Rosenbrock.

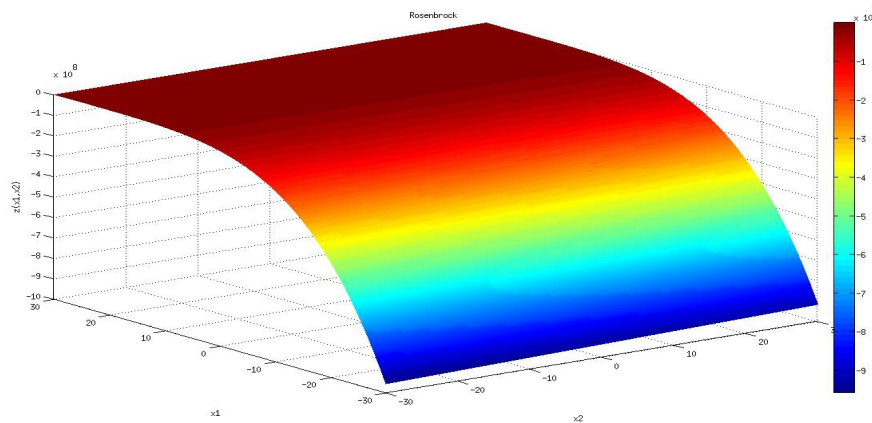


Figura 8: Función de Rosenbrock

#### 4.1.8 Función 8: Función de Colville:

Esta función esta definida por (27):

$$\begin{aligned} f(x_1, x_2, x_3, x_4) &= -100 \cdot (x_2' - x_1'^2)^2 - (1 - x_1')^2 - 90 \cdot (x_4' - x_3'^2) - (1 - x_3')^2 \\ &= -10.1 \cdot ((x_2' - 1)^2 - (x_4' - 1)^2) - 19.8 \cdot (x_2' - 1) \cdot (x_4' - 1) \end{aligned} \quad (27)$$

$$\text{con } x_i' = x_i - 7.123$$

La función tiene un máximo de 0 en  $x_i = 8.123$  para  $i = 1, \dots, 4$ . El rango de búsqueda será  $-10 \leq x_i \leq 10$ . Debido a que Colville es una función en 4 dimensiones por definición no puede

ser dibujada.

#### 4.1.9 Función 9: Six-Hump Camel-Back:

Esta función es 2D y esta definida por (28):

$$\begin{aligned} f(x_1, x_2) &= -4 \cdot (x_1 - 1)^2 + 2.1 \cdot (x_1 - 1)^4 - \frac{1}{3} \cdot (x_1 - 1)^6 - (x_1 - 1)(x_2 - 1) \\ &= +4 \cdot (x_2 - 1)^2 - 4 \cdot (x_2 - 1)^4 \end{aligned} \quad (28)$$

La función tiene un máximo de 1.0316285 en  $x_i = [1.08983, 0.2874]$  y  $x_i = [0.91017, 1.7126]$  para  $i=1,2$ . El rango de búsqueda será  $-5 \leq x_i \leq 5$ . Se anexa el dibujo de la función en 2D en la Figura 9: Función de Six-Hump.

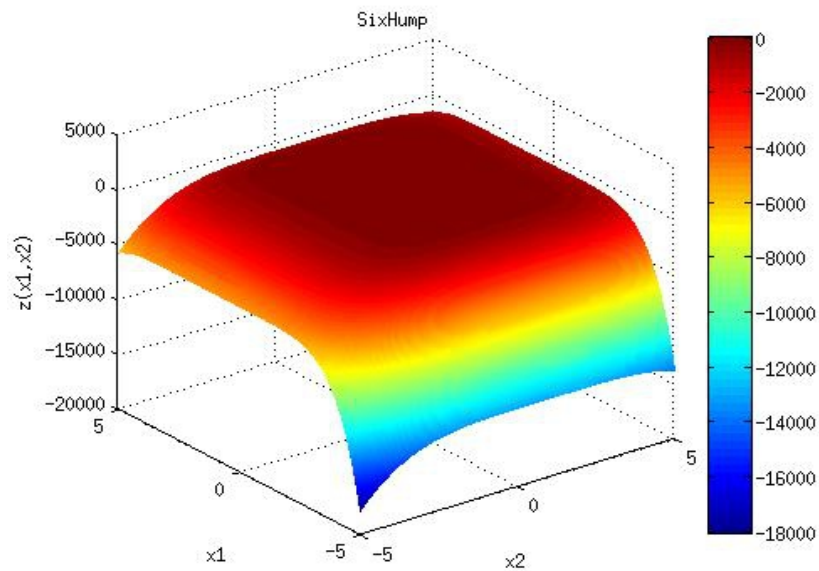


Figura 9: Función de Six-Hump

#### 4.1.10 Función 10: Función de Branin:

Esta función es 2D y esta definida por (29):

$$f(x_1, x_2) = -\left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi} - 6\right)^2 - 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) - 10 \quad (29)$$

La función tiene un máximo de -0.398 en  $x_i = [-3.142, 12.275]$ ,  $x_i = [3.142, 2.275]$  y  $x_i = [9.425, 2.425]$  para  $i=1,2$ . El rango de búsqueda será  $-5 \leq x_i \leq 15$ . Se anexa el dibujo de la función en la Figura 10: Función de Branin.

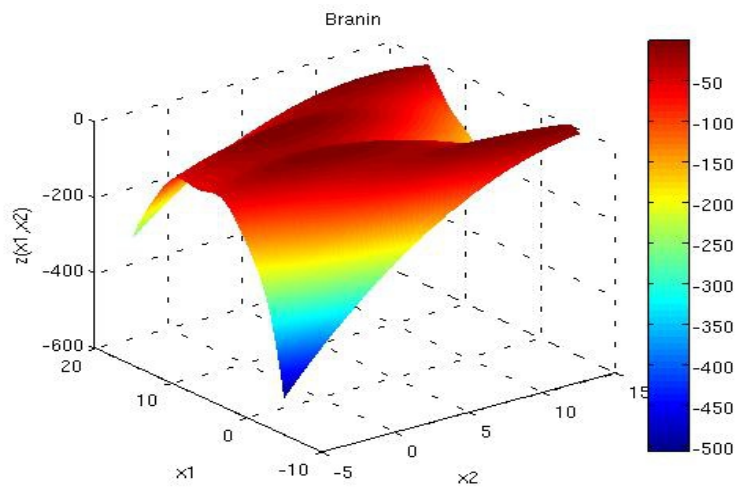


Figura 10: Función de Branin

#### 4.1.11 Función 11: Agujeros invertidos de Shekel:

Esta función es 2D y esta definida por (30):

$$f(x) = -\left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}\right]^{-1} \quad (30)$$

donde

$$a_{ij} = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 & 32 & 32 \end{bmatrix}$$

La función tiene un máximo de -1 en  $x_i = [-32, -32]$ . El rango de búsqueda será  $-65.536 \leq x_i \leq 65.536$ . Se anexa el dibujo de la función en la figura 11..

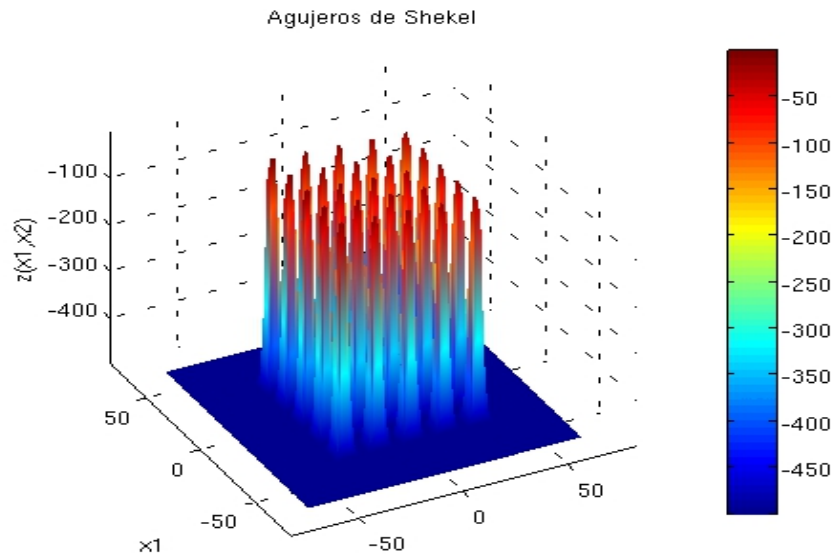


Figura 11: Función de agujeros invertidos de Shekel

#### 4.1.12 Función 12: Keane's Bump modificado:

Esta función es a trazos y se define como:

$$f(x_1, x_2) = \begin{cases} 0 & \text{para } x_1 + x_2 \geq 15 \text{ o } x_1 \cdot x_2 \leq 0.75; \\ \frac{\text{sino}(\cos^4(x_1) + \cos^4(x_2) - 2\cos^2(x_1)\cos^2(x_2))}{\sqrt{x_1^2 + 2x_2^2}} & \text{de otro modo} \end{cases} \quad (31)$$

La función tiene un valor máximo desconocido y su rango de búsqueda se encuentra en  $-5 \leq x_i \leq 5$ . Se anexa el dibujo de la función en la Figura 12: Función de Keane.

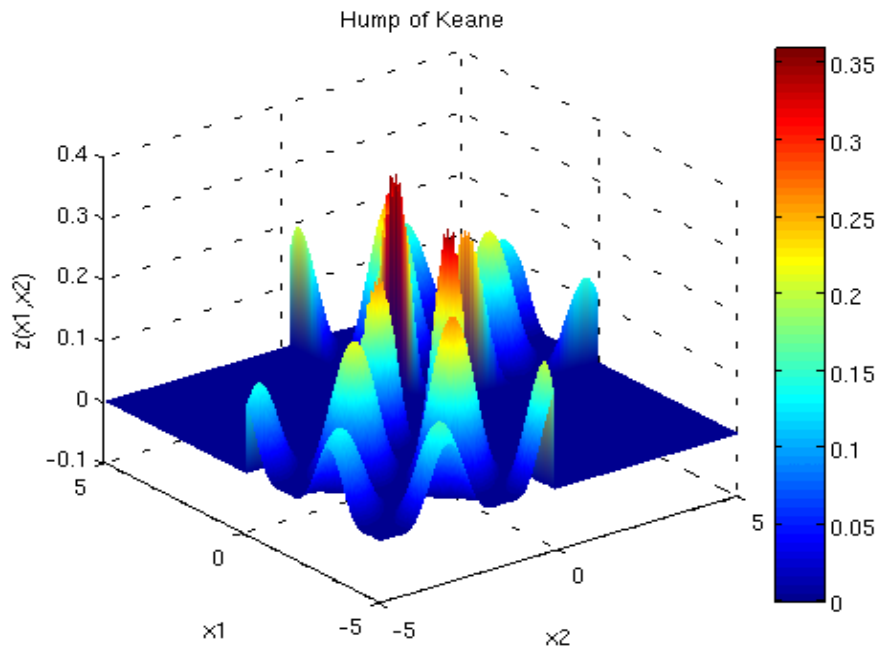


Figura 12: Función de Keane

#### 4.2 Algoritmo Híbrido Central Force Optimization y Simplex Nelder-Mead, CFOS:

CFO posee una potencialidad única para la exploración y mapeo de la topología de la función objetivo: las sondas enviadas al espacio de búsqueda se agrupan cerca de zonas donde se presenten máximos (locales y absoluto). Esta característica permite discriminar e identificar posibles zonas donde se pueda encontrar el máximo de la función objetivo así como mínimos locales dando una ventaja sensible sobre otros algoritmos como PSO y ACO que simplemente llevan todos sus hijos genéticos hacia un único y mejor punto, a criterio de los autores, este comportamiento puede ser explotado para mejorar el rendimiento del algoritmo simplex NM, al tener un mapeo previo de la función y determinar cuales áreas son sensibles e importantes para su exploración.

Existe una variación de CFO con simplex propuesta por R. Mahmoud [21], que consiste en ejecutar el CFO con un criterio de parada dado por número de iteraciones y tolerancia máxima, una vez que el algoritmo arroje un punto donde se encuentra el máximo, se toma este como inicial para determinar los vértices del simplex, para el caso de dos dimensiones,

los vértices serían:  $\{(x_1, x_2), (x_s, 0), (0, x_s)\}$  donde  $x_s = \sum_{i=1}^{N_d} x_i$  (31) tal como se plantea en [1].

En la figura 13 se muestra un sencillo diagrama de flujo de su lógica, si el lector desea consultarlo en detalle puede ser visto en [21]:

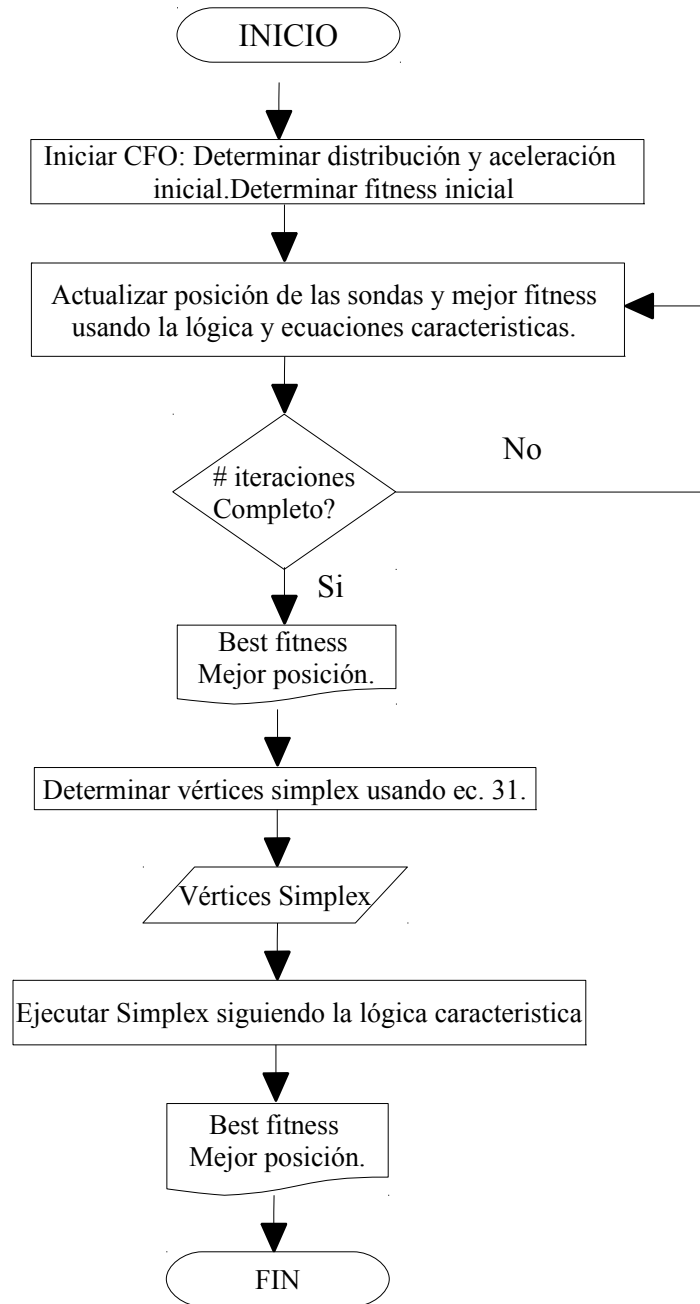


Figura 13: Diagrama de flujo híbrido simplex-CFO de Mahmoud

Este método usa el mapeo de CFO en una sola ejecución para determinar un área sensible de estudio, sin embargo, es posible que existan otras áreas de interés que pueden ser encontradas por CFO y explotadas con simplex.

En este trabajo se propone un híbrido diferente: Será CFO el que determina los vértices no solo de un simplex sino de varios a medida que se va ejecutando, con esta modalidad se estudiará un mayor número de áreas prometedoras, con lo que se espera generar un conjunto mayor de puntos de análisis y lograr encontrar, además de un máximo global, varios locales. La capacidad de encontrar máximos locales resulta útil a la hora de la solución de problemas de ingeniería, ya que el diseño y fabricación en el mundo real es mas eficiente (en términos de costos, por ejemplo) con una solución “sub-óptima” en lugar de la óptima. También es de esperar que al tener mayores áreas de análisis se disminuya el sensible problema del punto de inicio de ejecución de simplex, que hace que genere soluciones poco fiables en algunos casos, ya que se forman y analizan varios a medida que el algoritmo híbrido se ejecuta.

El diagrama de flujo de la Figura 14: Diagrama de flujo híbrido simplex-CFO planteado en esta tesis explica la solución planteada.

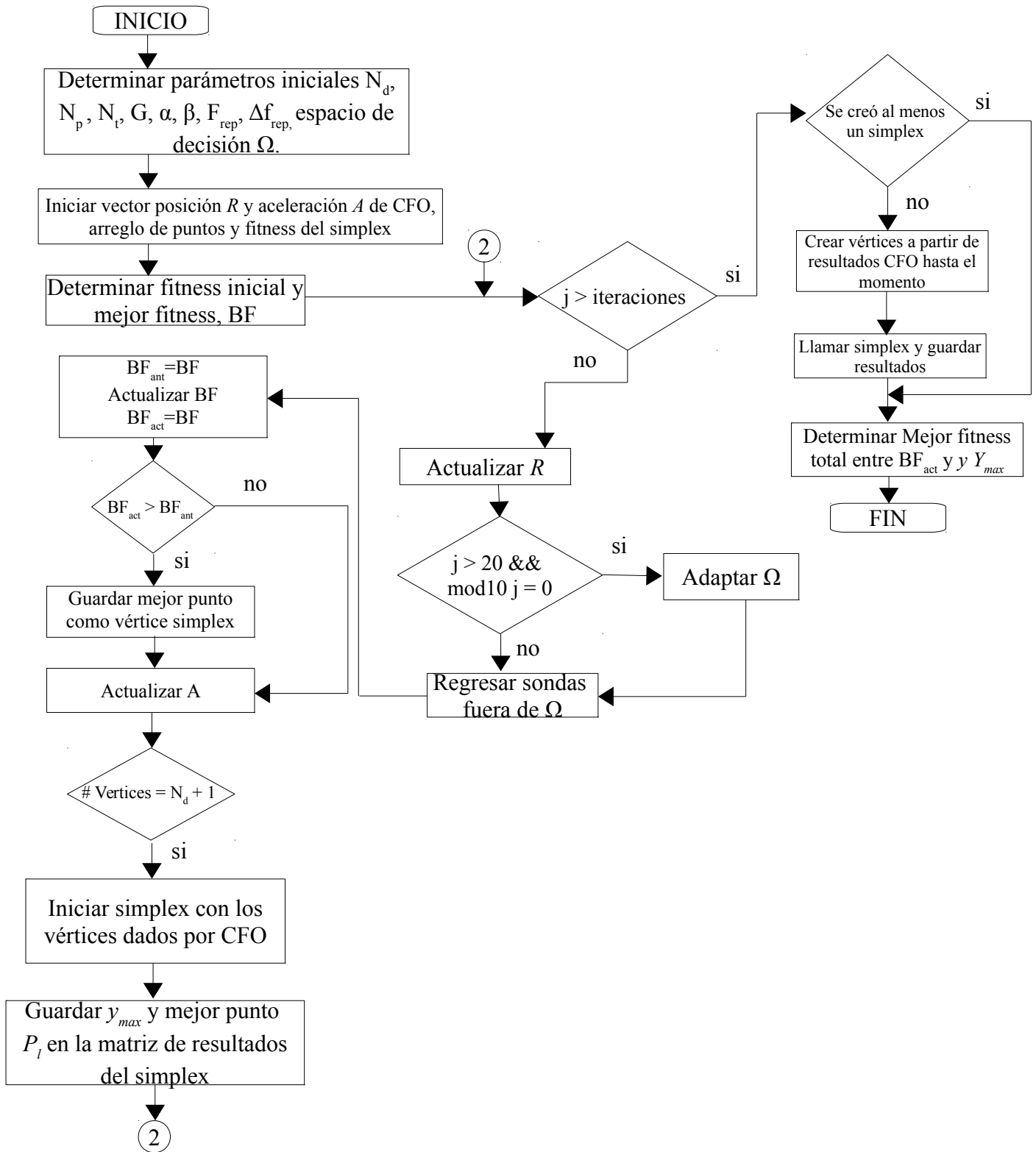


Figura 14: Diagrama de flujo híbrido simplex-CFO planteado en esta tesis

### 4.3 Tabla de evaluaciones de funciones típicas:

A continuación anexamos tablas de resultados del CFO de Formato desarrollado en 2007, CFO implementado para este trabajo con mejoras posteriores dadas por el autor en [13][22] y CFOS que se propone en esta tesis.

La Tabla 1 muestra los resultados obtenidos por el CFO con mejoras y el de Formato del 2007.

Claramente se observa que las mejoras planteadas incrementan la eficacia del algoritmo, siendo las funciones de Colville y Rastrigin los mejores ejemplos de mejoramiento del punto óptimo. Se mantienen constantes e iguales a 2 en esta y las siguientes implementaciones de CFO y del híbrido los valores de alfa, beta y la gravedad. La función de la esfera modificada puede hacernos creer que el algoritmo ha empeorado para encontrar su óptimo; sin embargo, en la implementación de Formato original se necesitaron 15000 sondas lo cual hace que el tiempo de computo sea absurdo y tarde varias horas resolviendo el problema tal cual se especifica en [22], mientras que para la nueva implementación sólo fueron necesarias 100 sondas reduciendo drásticamente el tiempo de ejecución para llegar a una respuesta razonable.

A pesar de las mejoras, el algoritmo sigue siendo muy sensible a las variaciones de gamma, por ello se especifica en la última columna los valores con los cuales dio una mejor respuesta luego de variarlo entre 0 y 1 en pasos de 0.1. Igualmente el factor de reposición y el delta de cambio del mismo hacen que el algoritmo llegue a desviarse del óptimo en caso de escogerse mal. Aún no se encuentra una forma efectiva en la literatura de CFO para escoger este parámetro y generalmente se determina a prueba y error luego de varias iteraciones.

Con esta implementación logramos comprobar que el algoritmo mejorado es mucho mas efectivo que el original, por lo cual, es el candidato a utilizar para determinar los vértices del simplex en el híbrido propuesto.

Tabla 1: Evaluación de CFO

Nombre de la función	Nt	Np	Neva	Máximo CFO	CFO Formato 2007	Máximo Teórico	alpha	beta	G	Posicion Mejor Sonda	Frep_ini	DeltaFrep	Y
Rosenbrock 30D	100	100	10000	-0.1697	-3,8	0	2	2	2	(26.1153-26.1153)	0,5	0	0,7
Esfera Modificada	100	100	10000	-0.1573	-0,0836	0	2	2	2	(75.1954-75.1954)	0,5	0	0
Colville	99	100	9900	-7.1959e-04	-19,387	0	2	2	2	(8.1239 - 8.1253)	0,5	0	0
Step	5	60	300	0	-1	0	2	2	2	(75-75)	0,5	0	0
Rastrigin	71	100	7100	-0.1646	-30,5308	0	2	2	2	(1.1180 - 1.1337)	0.5	0	0,9
Ackley	50	120	6000	-0.1199	-1,0066	0	2	2	2	(7.9883 - 7.9885)	0,5	0	0
Griewank	90	100	9000	-0.0136	-0,0459	0	2	2	2	(75.0368 - 75.1240)	0,5	0	0,5
Schewefel	186	120	22320	12569.1	12569,1	12569.5	2	2	2	(420.2912 - 422.2917)	0.5	0	0.6
Branin	20	400	8000	-0.4208	-0,398	-0,398	2	2	2	(-3.1910 12.5000)	0,99	0,2	0
	200	400	80000	-0.3990	-0,398	-0,398	2	2	2	(3.1365 2.3100)	0,99	0.35	0
	-	-	-	No encontrado	No encontrado	-0,398	-	-	-	No encontrado	0.5	0	0
Six Hump Camel Back	48	260	12480	1.0303	1,0316	1.0316	2	2	2	[1.0946 , 02748]	0,5	0	0.1
	30	480	14400	1.0135	1,0316	1.0316	2	2	2	[0.8438 , 1.7037]	0,5	0	0.7

Para las pruebas del híbrido se plantean las mismas funciones objetivo de prueba típicas descritas anteriormente. La Tabla 2 muestra los resultados obtenidos.

Analizando el máximo obtenido por el híbrido para las diferentes funciones se ve una notable mejora hallando el óptimo con respecto a CFO mejorado, los mejores ejemplos de ello son las funciones Step, Rosenbrock o Colville, cuyos máximos son prácticamente cero debido a que están en el orden de  $10e-19$ .

Caso de estudio aparte merecen las mejoras vistas en la esfera modificada y Branin, mientras para la primera Formato utilizó un número de sondas prohibitivo para un algoritmo iterativo, la mejora propuesta por él posteriormente logró reducir el tiempo de computo llegando a una respuesta aceptable; sin embargo, CFOS ha mejorado este fitness drásticamente alcanzando el punto óptimo teórico. Mientras que con Branin CFO nunca llegó a hallar los tres puntos con ningún arreglo de factor de reposición y gamma, CFOS logra encontrarlos con un error relativo menor al 0.002%.

Agregar el simplex mejora la dependencia de las constantes del factor de reposición, el delta de éste y el gamma de la distribución inicial de sondas (fue necesaria la estrategia de punto medio para la mayoría de funciones con gamma igual a cero). Sin embargo, no la elimina como se ve, por ejemplo, en el desarrollo de Branin, donde fue necesario modificarlas para encontrar los distintos óptimos.

Tabla 2 :Evaluación CFOS.

Nombre de la función	Nt	Np	Neva	Máximo híbrido	Máximo Teórico	Posición Mejor Sonda	Frep_ini	DeltaFrep	Y
Rosenbrock 30D	200	200	40000	-2.9331e-19	0	(26.1230 – 26.1230)	0,5	0	0
Esfera Modificada	200	200	40000	-3.0678e-19	0	(75.1230 – 75.1230)	0,5	0	0
Colville	200	200	40000	-1.0785e-18	0	(8,123 – 8,123)	0,5	0	0
Step	200	200	40000	0	0	(75-75)	0,5	0	0
Rastrigin	80	600	48000	0.9950	0	(0,1228 - 1.123)	0.5	0	0
Ackley	200	200	40000	-4,44E-015	0	( 4,321 – 4,321)	0,5	0	0
Griewank	200	200	40000	-0.0065	0	(74.9941 – 75.2935)	0,5	0	0.1
Schewefel	100	100	10000	12569.1	12569.5	( 421.2917 – 421.2917)	0.5	0	0.7
Branin	200	200	40000	-0.4208	-0,3978	( -3.14159 12.2750)	0,5	0	0
	200	200	40000	-0.3990	-0,3978	(3.14159 2.2749)	0,99	0.35	0
	200	200	40000	-0.3990	0,3978	( 9.4247 2,475)	0,99	0,3	0,3
Six Hump Camel Back	200	200	40000	1.0303	1.0316	[1.0898 0.2873]	0,5	0	0
	200	200	40000	1.0135	1.0316	[ 0.910 , 1.712]	0.5	0	0

Otro punto a favor que podemos inferir de la Tabla 2 es el uso del mismo número de sondas y de iteraciones para la mayoría de implementaciones. Independizar el resultado de la variación de los factores propios del algoritmo de exploración, es tan importante como evitar un punto de inicio alejado de la solución a la hora de usar únicamente simplex. Al realizar el híbrido se han disminuido estas sensibilidades, logrando obtener un algoritmo más eficiente aprovechando las ventajas de cada una de las estrategias que lo conforman.

Las implementaciones anteriores del algoritmo, y las demás consignadas en este trabajo, se hicieron en un equipo con las especificaciones dadas en la Tabla 3.

Tabla 3 : Especificaciones equipo utilizado para implementar el algoritmo.

Fabricante	ASUS®
Modelo	A45Vd
Procesador	Intel® Core i7-3630QM a 2.4GHz
RAM	8 GB
Sistema Operativo	Linux-Ubuntu

## 5. ALGORITMO HIBRIDO EN SISTEMAS DE ECUACIONES NO LINEALES:

Se plantea probar el algoritmo en sistemas de ecuaciones no lineales de orden 2 y 5 planteados en [4] y en la resolución de circuitos electrónicos mediante estrategias de optimización de investigaciones al interior del grupo CEMOS descritas en [19], [20] para darle una aplicación práctica.

### 5.1 SISTEMAS DE ECUACIONES NO LINEALES:

#### 5.1.1 Sistema 1:

El sistema 1 consta de dos ecuaciones con dos incógnitas:

$$\begin{aligned}
 f_1 &= 3x_1^2 - x_2^2 \\
 f_2 &= 3x_1^2 x_2^2 - x_1^3 - 1 \\
 f_{obj}(x_1, x_2) &= (3x_1^2 - x_2^2)^2 + (3x_1^2 x_2^2 - x_1^3 - 1)^2 = 0
 \end{aligned}
 \tag{32}$$

Las raíces de este sistema se encuentran en  $[x_1, x_2]=[0.5, \pm 0.86]$ .

### 5.1.2 Sistema 2:

El sistema 2 consta de dos ecuaciones con dos incógnitas:

$$\begin{aligned} f_1 &= x_1(1-x_1) + 4x_2 - 12 \\ f_2 &= (x_1-2)^2 + (2x_2-3)^2 - 25 \\ f_{obj}(x_1, x_2) &= [x_1(1-x_1) + 4x_2 - 12]^2 + [(x_1-2)^2 + (2x_2-3)^2 - 25]^2 = 0 \end{aligned} \quad (33)$$

Las raíces de este sistema se encuentran en  $[x_1, x_2]=[-1, 3.25]$  y  $[2.55, 3.98]$ .

### 5.1.3 Sistema 3:

Finalmente se considera un sistema de 5 ecuaciones con 5 incógnitas.

Las raíces no se especifican en la fuente de consulta.

El sistema esta descrito por las ecuaciones de (34) dadas a continuación:

$$\begin{aligned} f_1 &= 2.3x_1 + x_2^2 - 2x_4 + 0.01x_5 - 1.45 \\ f_2 &= -x_2 + 1.3x_5 + 9 \\ f_3 &= x_2x_3 - x_5^2 + 9 \\ f_4 &= x_1^3 - 2x_4x_3 + x_5^2 - 0.8 \\ f_5 &= -5x_3 - x_5 - 3x_5x_4x_3 + 3.6 \\ f_{obj}(x_1, x_2, x_3, x_4, x_5) &= [2.3x_1 + x_2^2 - 2x_4 + 0.01x_5 - 1.45]^2 + [-x_2 + 1.3x_5 + 9]^2 \\ &\quad + [x_2x_3 - x_5^2 + 9]^2 + [x_1^3 - 2x_4x_3 + x_5^2 - 0.8]^2 \\ &\quad + [-5x_3 - x_5 - 3x_5x_4x_3 + 3.6]^2 = 0 \end{aligned} \quad (34)$$

## 5.2 RESULTADOS EVALUACIONES SISTEMAS NO LINEALES

La Tabla 4 muestra los resultados de la solución de los sistemas no lineales descritos en la sección anterior por (32) , (33) y (34); el espacio de búsqueda nuevamente se tomó de forma arbitraria entre -30 y 30 para comenzar la solución en puntos alejados del óptimo y probar si el algoritmo era capaz de llegar a respuestas adecuadas.

Tabla 4: Resultados híbrido sistemas de ecuaciones no lineales.

Frep_ini	DeltaFrep	Y	Función	Nt	Np	Neva	Máximo híbrido	Máximo Teórico	Posición Mejor Sonda
0,5	0	0	Sistema 1	100	100	10000	-2.9148e-20	0	[0.5000 -0.8660]
0,5	0	0.1	Sistema 1	100	100	10000	-1.3397e-19	0	[0.5000 0.8660]
0,5	0	0	Sistema 2	100	100	10000	-9.7959e-19	0	[2.5469 3.9850]
0,5	0	0.1	Sistema 2	100	100	10000	-2.7781e-20	0	[-1.0000 3.5000]
0,5	0	0.7	Sistema 3	100	100	10000	-4.1939e-19	0	[-2.1086 5.1868 -0.0764 10.2870 -2.9332]
0,5	0	0.8	Sistema 3	100	100	10000	-1.0232e-19	0	[-2.0065 12.9023 0.0008 80.2171 3.0018]
0,5	0	1	Sistema 3	100	100	10000	-1.0793e-19	0	[-2.6886 2.8523 4.6853 0.2271 -4.7290]

El híbrido demuestra una exactitud bastante alta encontrando el óptimo de los diferentes sistemas y dando como puntos de solución las raíces teóricas de los dos primeros. Para el tercero logró encontrar 3 valores diferentes variando el factor de gamma. Cabe destacar que para los sistemas uno y dos, sólo fue necesario variar este factor una vez para llegar a las respuestas deseadas, mientras que, para el sistema tres se consignan los mejores fitness para una variación de gamma entre 0 y 1.

El sistema tres no posee solución teórica, a pesar de no poder comparar con un valor al cual llegar, se ve que el fitness que encuentra está en el orden de  $10e-19$  que da a entender la destreza para resolver el problema.

Se destaca de la Tabla 4, el bajo costo computacional que utiliza el algoritmo al enviar solo 100 sondas y realizar el mismo número de iteraciones. En algunos casos, durante la implementación, se llegó a un buen resultado en iteraciones anteriores a la 100; sin embargo, el tiempo de cómputo nunca supero los 3 segundos para resolver los tres sistemas, por lo cual, dejamos este número de sondas e iteraciones como estándar para el problema.

Finalmente, aumentar el número de sondas y de iteraciones no cambió la respuesta a la que

se llegaba; sin embargo, se sigue presentando (aunque en menor medida) una clara dependencia del factor gamma para llegar a determinada respuesta del problema; pero, lograr que los resultados no dependan del factor de reposición, o el número de sondas, es un avance significativo que mejora la exactitud de CFO en la solución problemas específicos de ingeniería como resolver circuitos con transistores y diodos, campo donde el algoritmo presenta varios fallos [20].

### 5.3 CIRCUITOS ELECTRÓNICOS MODELADOS POR MEDIO DE SISTEMAS DE ECUACIONES NO LINEALES:

#### 5.3.1 Circuito 1: Circuito Electrónico con 1 diodo.

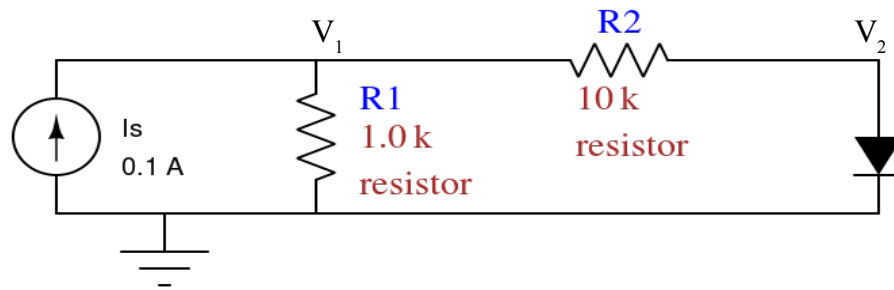


Figura 15: Circuito electrónico con un diodo

El primer circuito a analizar consta de un diodo como elemento no lineal como se muestra en la Figura 15: Circuito electrónico con un diodo, dos elementos pasivos y una fuente de corriente independiente. Realizando un análisis de nodos podemos encontrar las ecuaciones de (35) que modelan el circuito tal cual se especifica en [23].

Los parámetros propios del circuito serán:

$$I_{sat} = 1 \times 10^{-15} ; V_t = 25.875 \times 10^{-3}$$

$$I_s = 0.01 [A] ; R_1 = 100 \Omega ; R_2 = 10 \times 10^3 \Omega$$

Las ecuaciones que lo modelan se describen en (35):

$$\begin{aligned}
\text{Nodo 1: } & \left(\frac{1}{R_1} + \frac{1}{R_2}\right) \cdot v_1 - \left(\frac{1}{R_2}\right) \cdot v_2 - I_s = 0 \\
\text{Nodo 2: } & -\frac{1}{R_2} \cdot v_1 + \left(\frac{1}{R_2} + G_{eq}\right) \cdot v_2 + I_{eq} = 0 \\
f_{obj}(v_1, v_2) & = \left[ \left(\frac{1}{R_1} + \frac{1}{R_2}\right) \cdot v_1 - \left(\frac{1}{R_2}\right) \cdot v_2 - I_s \right]^2 + \left[ -\frac{1}{R_2} \cdot v_1 + \left(\frac{1}{R_2} + G_{eq}\right) \cdot v_2 + I_{eq} \right]^2 = 0 \quad (35) \\
I_{DI} & = I_{sat} \left( e^{\frac{V_{DI}}{V_t}} - 1 \right); \quad G_{eq} = \frac{I_{sat}}{V_t} e^{\frac{V_{DI}}{V_t}}; \quad I_{eq} = I_{DI} - G_{eq} V_{DI}; \quad V_{DI} = v_2
\end{aligned}$$

Usando la ecuación objetivo dada en (35) se comparan los resultados de los algoritmos PSO, UPSO, NRM y CFO con el híbrido CFOS propuesto por nosotros. Para esta y las demás pruebas con circuitos se varía el factor  $\gamma$  desde 0 a 1 en pasos de 0.1,  $\alpha$ ,  $\beta$  y  $G$  serán iguales a 2 y se consignan los mejores resultados de CFO y CFOS junto a los obtenidos por Cruz en [19].

En la Tabla 5 se pueden ver los resultados de las diferentes implementaciones. Para CFO el mejor óptimo se obtuvo para  $\gamma = 0.1$  mientras para CFOS variar  $\gamma$  fue indiferente llegando al mismo punto siempre. El espacio de búsqueda se definió arbitrariamente entre -30 y 30.

Tabla 5: Resultados híbrido para circuito electrónico uno.

	<b>PSO</b>	<b>UPSO</b>	<b>NRM</b>	<b>CFO</b>	<b>CFOS</b>	<b>PSPICE</b>
v1	10	9,908	9,908	9.3939	9.9080	9,908
v2	0	0.7128	0.7128	-24	0.7128	0.7125
Precisión	2.00e -06	7.61e -06	0.00e+0.00	-1.8557e -05	9.6332e-21	
Tiempo [s]	0.28	0.06	0.16	5.012550	0.212895	0.00
Sondas	10	10		200	10	
Iteraciones	433	98	369	100	35	

CFOS incrementa notablemente el resultado de CFO, que al parecer ha quedado atrapado en un máximo local. Lo interesante de la prueba, viene al comparar los resultados con los demás algoritmos. Mientras PSO se mantiene alejado del punto deseado vemos que UPSO, NRM y CFOS llegan al mismo pero con precisiones (fitness) diferentes, debido a las cifras decimales de las respuestas encontradas. CFOS esta claramente en ventaja frente a las demás estrategias implementadas llegando, al igual que UPSO y NRM, prácticamente a la misma respuesta del programa de simulación comercial PSPICE, pero con una precisión mayor, usando el mismo número de sondas pero con un número de iteraciones de mapeo

menor y un tiempo de ejecución corto (menor a 1 segundo).

### 5.3.2 Circuito 2: Circuito Electrónico con 2 diodos.

La siguiente topología consta de elementos pasivos y dos diodos como se muestra en la Figura 16: Circuito electrónico con dos diodos. Esta configuración hace que las ecuaciones que lo describen sean no lineales y más complejas en solución. De acuerdo a lo planteado por Baldick en [24] las ecuaciones que describen el circuito y  $f_{obj}$  pueden ser planteadas así:

$$\begin{aligned}
 f_1 &= \left(\frac{1}{R_a}\right)v_1 + I_{sat} \left(e^{\frac{qV_b}{nk_{BT}}} - 1\right) - I_1 = 0 \\
 f_2 &= -I_{sat} \left(e^{\frac{qV_b}{nk_{BT}}} - 1\right) + \left(\frac{1}{R_c} + \frac{1}{R_d}\right)v_2 - \left(\frac{1}{R_d}\right)v_3 = 0 \\
 f_3 &= -\left(\frac{1}{R_d}\right)v_2 + \left(\frac{1}{R_d} + \frac{1}{R_e} + \frac{1}{R_f}\right)v_3 - \left(\frac{1}{R_f}\right)v_4 = 0 \\
 f_4 &= -\left(\frac{1}{R_f}\right)v_3 + \left(\frac{1}{R_f}\right)v_4 + I_{sat} \left(e^{\frac{qV_g}{nk_{BT}}} - 1\right) = 0 \\
 f_{obj}(v_1, v_2, v_3, v_4) &= \left\{ \left(\frac{1}{R_a}\right)v_1 + I_{sat} \left(e^{\frac{qV_b}{nk_{BT}}} - 1\right) - I_1 \right\}^2 + \left\{ -I_{sat} \left(e^{\frac{qV_b}{nk_{BT}}} - 1\right) + \left(\frac{1}{R_c} + \frac{1}{R_d}\right)v_2 - \left(\frac{1}{R_d}\right)v_3 \right\}^2 \\
 &\quad + \left\{ -\left(\frac{1}{R_d}\right)v_2 + \left(\frac{1}{R_d} + \frac{1}{R_e} + \frac{1}{R_f}\right)v_3 - \left(\frac{1}{R_f}\right)v_4 \right\}^2 + \left\{ -\left(\frac{1}{R_f}\right)v_3 + \left(\frac{1}{R_f}\right)v_4 + I_{sat} \left(e^{\frac{qV_g}{nk_{BT}}} - 1\right) \right\}^2 = 0
 \end{aligned} \tag{36}$$

donde  $v_b = v_1 - v_2$ ,  $v_g = v_4$ .

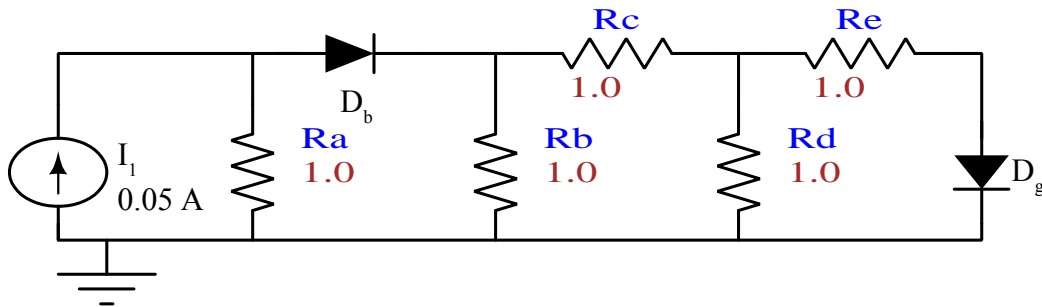


Figura 16: Circuito electrónico con dos diodo

Los parámetros de entrada del circuito se describen en la Tabla 6.

Tabla 6: Parámetros de entrada circuito electrónico de prueba dos.

Isat	1x10 <sup>-6</sup>
Ra, Rb, Rc, Rd, Re, Rf	1 [ Ω ]
$\frac{q}{nk_{BT}}$	40 [ V <sup>-1</sup> ]
I1	0.05 [ A ]

En la Tabla 7 se pueden ver los resultados de las diferentes implementaciones. Para CFO el mejor óptimo se obtuvo para  $\gamma = 0.5$  mientras para CFOS variar  $\gamma$  fue indiferente llegando al mismo punto de solución siempre. El espacio de búsqueda se definió arbitrariamente entre -30 y 30.

Tabla 7: Resultados híbrido circuito electrónico dos.

	PSO	UPSO	NRM	CFO	CFOS	PSPICE
v1	4,80E-002	4.99936e-02	5,00E-002	-0.6122	0.04999	0.05
v2	2.62309e-13	4.25725e-06	4.25725e-06	0	4.2574e-6	3.93900e-6
v3	1.59533e-13	2.12858e-06	2.12858e-06	0	2.1288e-6	1.9700E-06
v4	3.18486e-13	2.12850e-06	2.12850e-06	0	2.1288e-6	1.9690E-06
Precisión	3.81e-06	7,06E-043	8,99E-022	-0.4386	6.36851e-20	
Tiempo [s]	1.86	1.28	0.01	9.694499	0.452033	0.02
Sondas	50	100		200	60	
Iteraciones	516	292	5	100	35	

Mientras PSO y UPSO tienen un rango de búsqueda entre 0 y 1 [V], CFOS logra llegar prácticamente a la misma respuesta de los anteriores en un espacio de búsqueda mas amplio. A pesar de esto, el mejor fitness encontrado es el de UPSO, pero el orden de 10e-20 encontrado por CFOS es aceptable y los puntos de solución se acercan a los obtenidos por PSPICE.

Nuevamente la implementación con CFO se aleja bastante de los resultados de los demás algoritmos y su precisión no es satisfactoria para resolver el ejercicio.

Otro parámetro importante es el número de sondas utilizado, con tan solo 60 en 35 iteraciones se logra llegar a una respuesta comparable a la de los demás algoritmos genéticos de la Tabla 7, que junto a la no dependencia del factor de reposición y de gamma habla muy bien de la implementación del híbrido.

### 5.3.3 Circuito 3: Espejo de Corriente Widlar.

Esta topología con espejo de corriente Widlar planteada por Roa y Ramírez [19] consta de dos transistores mosfet unidos por sus compuertas y con una resistencia  $R_s$  conectada al drain del mosfet dos que hace que la corriente de salida sea distinta a la de entrada dado que los potenciales puerta-surtidor de los dos transistores no serán los mismos. El hecho de colocar esta resistencia hace que el circuito no sea un espejo de corriente que dependa solo del ancho de los transistores unidos y tenga que ser descrito con las ecuaciones de mallas descritas en (37). El circuito puede ser visto en la Figura 17: Espejo de corriente de Widlar modificado

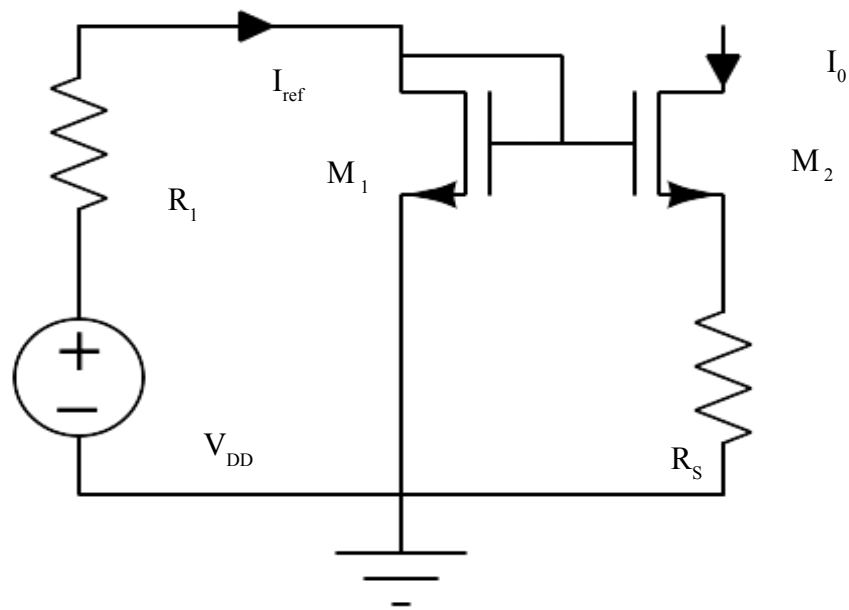


Figura 17: Espejo de corriente de Widlar modificado

$$\begin{aligned}
f_1 &= -V_{GS1} + V_{GS2} + R_s I_o = 0 \\
f_2 &= \frac{1}{2} k_n \left( \frac{W}{L} \right)_2 (V_{GS2} - V_t)^2 = 0 \\
f_{obj}(V_{GS2}, I_o) &= [-V_{GS1} + V_{GS2} + R_s I_o]^2 + \left[ \frac{1}{2} k_n \left( \frac{W}{L} \right)_2 (V_{GS2} - V_t)^2 \right]^2 = 0 \quad (37)
\end{aligned}$$

donde

$$I_{ref} = \frac{1}{2} k_n \left( \frac{W}{L} \right)_1 [(R_s \cdot I_o + V_{GS2}) - V_t]^2 ;$$

Se comprobará la eficacia del algoritmo hallando el valor de  $I_{ref}$  con el punto de solución de la función objetivo. Los parámetros de entrada se encuentran en la Tabla 8 mientras en la Tabla 9 se consignan los resultados obtenidos por Roa y Ramírez con un algoritmo CFO adaptado especialmente al problema, nuestro algoritmo CFO base, el híbrido propuesto y la corriente de referencia calculada con los puntos óptimos obtenidos. Cruz no plantea este ejercicio por lo cual no se dejan los resultados de PSO y UPSO con su método de solución.

Los valores de gamma para CFO y CFOS son 0.8 y 0 respectivamente. En esta ocasión CFOS varió algunos puntos al modificar gamma, sin embargo, algunas de las respuestas dadas no tenían un sentido físico coherente y el resto variaban en el orden de  $1 \times 10^{-7}$  con respecto a la consignada por lo cual las consideramos iguales. Roa y Ramírez no especifican con que valor de este parámetro implementaron su solución.

Tabla 8: Parámetros de entrada circuito electrónico de prueba tres

Iref	$100 \times 10^{-6}$
Rs	$1 \times 10^3 [\Omega]$
$k_n \left( \frac{W}{L} \right)_1 = k_n \left( \frac{W}{L} \right)_2$	$19.4 \times 10^{-3} [A/V^2]$
Vgs1	0.8015[V]
Vt	0.7 [V]

Tabla 9: Resultados híbrido circuito electrónico dos

	<b>CFO (R &amp; R)*</b>	<b>CFO</b>	<b>CFOS</b>
Vgs2 [ V ]	7.6300e-1	0.4073	0.80055
Io [ A ]	3.8499e-5	0.3929e-3	9.51099e-7
Precisión	-3.3378e-11	-1.9032e-06	6.36851e-20
Tiempo [s]	No especifica	5.130859	0.452033
Sondas	No especifica	200	60
Iteraciones	No especifica	200	20
<b>Iref</b>	99.9299e-06	97.3402e-06	99.9318e-6

(\*) Los valores de CFO(R&R) son promedios de varios resultados obtenidos en 10 iteraciones distintas, el rango de puntos hallados es el siguiente:  $v_{GS2}: [7.5746e-1 - 7.6425e-1]$ ,  $I_o: [3.7558e-5 - 4.4077e-5]$ .

Se ve de nuevo la dependencia del factor gamma y del número de sondas en CFO para llegar a una respuesta adecuada. Tanto CFO(R&R) y CFO básico, llegan a un resultado para el cálculo de la corriente de referencia bastante bueno; sin embargo, CFOS mejora este resultado y la precisión de las otras dos implementaciones con pocas sondas y un número de pasos menor.

Importante destacar que es CFOS la estrategia que mas se acerca al valor teórico de la corriente de referencia de  $100e-6$  [ A ] planteado por el ejercicio. CFO obtiene por sí solo una precisión del orden de  $1e-6$ , lo cual nos sugiere una adecuada respuesta; sin embargo, la corriente de referencia calculada con sus valores da alejada del punto teórico. CFO(R&R) es un promedio de resultados (buenos y no tan buenos) de varias iteraciones del programa, a pesar de que este promedio da un resultado adecuado no se puede, asegurar que para un número mayor de iteraciones este se mantenga.

#### 5.3.4 Circuito 4: Circuito con MOSFET y diodo.

El siguiente circuito es un convertidor DC-DC que reduce nivel de tensión muy utilizado debido a su buen comportamiento frente al ruido [19]. La Figura 18: Circuito con MOSFET y diodo muestra el circuito del convertidor, el transistor se comporta como un interruptor analógico que da paso de corriente durante el ciclo positivo de la señal de control  $V_{pwm}$  que hace que el

diodo se polarice inversamente permitiendo cargar los diferentes elementos almacenadores de energía. Durante el ciclo negativo de la señal de control el diodo se polarizará inversamente cerrando el circuito y haciendo que la bobina y condensador previamente cargados entreguen su energía a la carga.

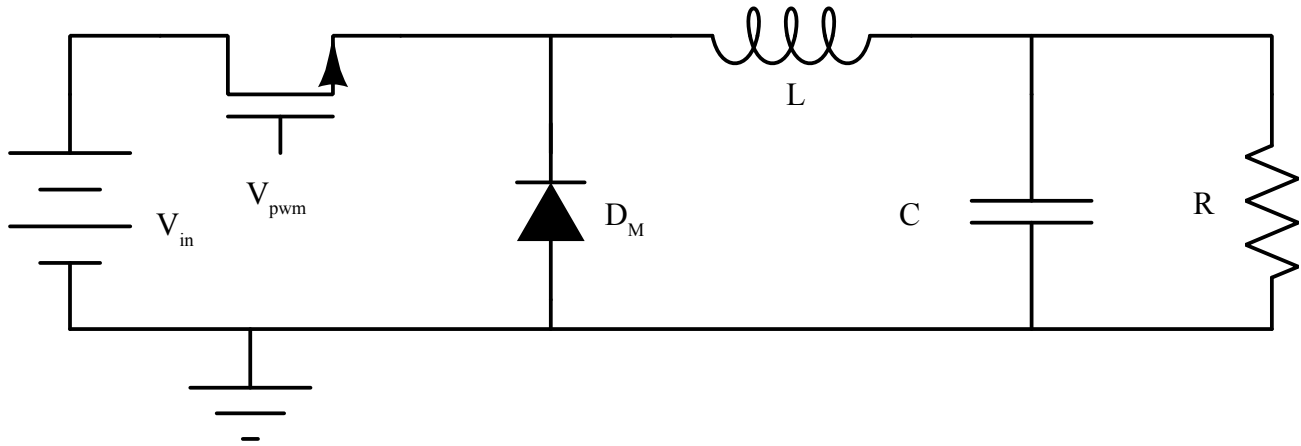


Figura 18: Circuito con MOSFET y diodo

Por simplicidad, y dado que el caso de estudio es obtener las tensiones de polarización, el circuito se simplifica como muestra la Figura 19: Circuito con MOSFET y diodo en DC y se realiza su análisis en DC modelando la bobina y el capacitor como un puente y un circuito abierto respectivamente. Aparecen nuevas resistencias que representan el valor propio de las mismas en la bobina y la fuente.

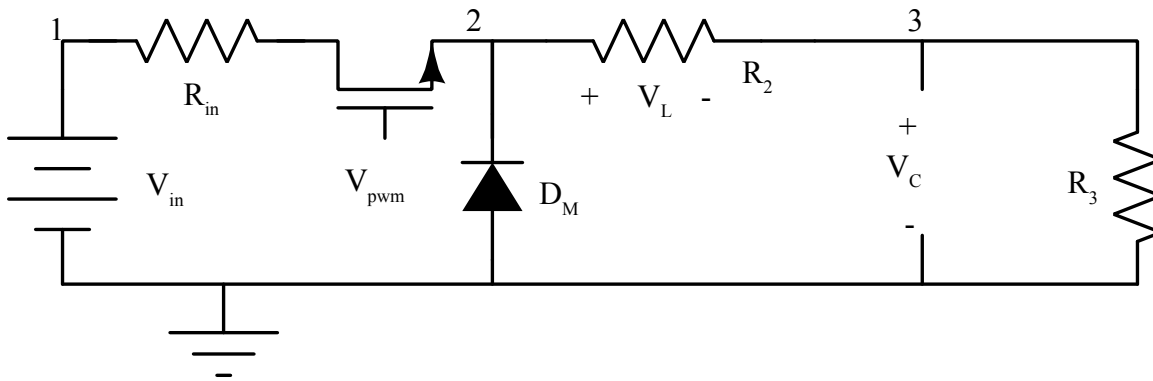


Figura 19: Circuito con MOSFET y diodo en DC

Las ecuaciones que describen el circuito para implementar la estrategia de optimización son las siguientes:

$$\begin{aligned}
f_1 &= \frac{1}{R_1} v_1 - \frac{1}{R_1} V_s - I_M = 0 \\
f_2 &= I_M + I_{D_M} - \frac{1}{R_2} v_2 + \frac{1}{R_2} v_3 = 0 \\
f_3 &= \left( \frac{1}{R_2} + \frac{1}{R_3} \right) v_3 - \frac{1}{R_2} v_2 = 0 \\
f_{obj}(v) &= \left( \frac{1}{R_1} v_1 - \frac{1}{R_1} V_s - I_M \right)^2 + \left( I_M + I_{D_M} - \frac{1}{R_2} v_2 + \frac{1}{R_2} v_3 \right)^2 + \left( \left( \frac{1}{R_2} + \frac{1}{R_3} \right) v_3 - \frac{1}{R_2} v_2 \right)^2 = 0
\end{aligned} \tag{38}$$

donde:

$$i_M(V_{GS}, V_{DS}) = \begin{cases} k'_p \left( \frac{W}{L} \right) \cdot (1 + \lambda V_{DS}) \cdot \left( V_{OV} \cdot V_{DS} - \frac{1}{2} V_{DS}^2 \right), & V_{DS} \leq V_{OV}, V_{OV} \geq 0; \\ \frac{1}{2} k'_p \left( \frac{W}{L} \right) \cdot (1 + \lambda V_{DS}) \cdot V_{OV}^2, & V_{DS} \geq V_{OV}, V_{OV} \geq 0; \\ 0, & V_{OV} \leq 0 \end{cases} \tag{39}$$

$$V_{OV} = V_{GS} - V_{TH} \tag{40}$$

$$i_D(V_D) = I_S \left( e^{\frac{q \cdot V_D}{nkT}} - 1 \right)$$

$$\begin{aligned}
I_M &= i_M(V_{pwm} - v_2, v_1 - v_2); \\
I_{D_M} &= i_d(-v_2);
\end{aligned} \tag{41}$$

Los parámetros propios del circuito se describen en la Tabla 10 .

Tabla 10: Parámetros de entrada circuito electrónico cuatro.

$V_s, V_{pwm}$	5[V]	$k'_p$	$150 \times 10^{-6} \text{ [A/V}^2\text{]}$
$\frac{q}{nkT}$	40 [V <sup>-1</sup> ]	$\lambda$	0.02 [V <sup>-1</sup> ]
R1	10 [Ω]	VTH	1.45 [V]
R2	0.1 [Ω]	W	$30 \times 10^{-6} \text{ [m]}$
R3	$500 \times 10^3 \text{ [Ω]}$	L	$10 \times 10^{-6} \text{ [m]}$
Is	$1 \times 10^{-6} \text{ [A]}$		

En la Tabla 11: Resultados híbrido circuito electrónico cuatro. se consignan los resultados de Cruz, Roa y Ramírez y el híbrido propuesto. El espacio de búsqueda usado para PSO y UPSO se encuentra entre 0 y 5 [V] mientras para CFOS se continua entre -30 y 30 y gamma

igual a 0.

Tabla 11: Resultados híbrido circuito electrónico cuatro.

	<b>PSO</b>	<b>UPSO</b>	<b>NRM</b>	<b>CFO(R&amp;R)*</b>	<b>CFOS</b>	<b>PSPICE</b>
v1	5.0	4.99936e-02	4.99992	5.0002	5.0001	4,99992
v2	5.0	3.42271	3.36754	3.5067	3.5477	3.36754
v3	5.0	3.42271	3.36754	3.5067	3.5477	3.36754
Precisión	8.6300e-11	2.3500e-11	4.7100e-14	-94.7080e-06	1.7180e-19	
Tiempo [s]	2.58	2.81	0.0232	No especifica	0.220434	0.02
Sondas	100	100	100	No especifica	60	
Iteraciones	104	106	23	No especifica	20	

(\*) Los valores de CFO(R&R) son promedios de varios resultados obtenidos en 4 iteraciones distintas, el rango de puntos hallados es el siguiente:  $v_1: [4.9866 - 5.0134]$ ,  $v_2: [2.1319 - 4.1216]$ ,  $v_3: [2.1319 - 4.1216]$ .

Para esta topología, CFOS no logra igualar la exactitud alcanzada por NRM, siendo superado por éste en precisión y tiempo de computo. Sin embargo, la aproximación al punto de PSPICE es bastante buena, y dado que los óptimos hallados por Roa y Ramírez son promedios de cuatro iteraciones, no se puede asegurar que resultado final se obtendría para varias iteraciones del algoritmo debido al factor aleatorio de disposición de sondas incluido por estos autores. Por lo anterior y dado que CFOS siempre llega a una misma respuesta al ser totalmente determinístico, los resultados para esta topología siguen siendo buenos.

Respecto al esfuerzo de computación, es claro que CFOS logra llegar a su óptimo consumiendo pocos recursos informáticos y utilizando un número menor de sondas e iteraciones que UPSO y NRM. El fitness encontrado por el híbrido es mejor que el de los otros algoritmos genéticos, por lo cual podría pensarse que la solución al sistema no es la óptima sino una “sub”-óptima dado que la función objetivo es la misma para todos.

## 6. CONCLUSIONES

- Se cumplieron los objetivos del proyecto.
- La mejora que agrega el simplex al CFO aumenta su precisión para funciones objetivo típicas y aplicaciones prácticas, como circuitos electrónicos descritos mediante

ecuaciones no lineales, dejando claro la importancia del mapeo previo de  $f_{obj}$ .

- Se propuso una nueva estrategia para un algoritmo híbrido entre Simplex y CFO con resultados satisfactorios para funciones de prueba y sistemas no lineales llegando a precisiones de  $1e-20$  en algunos casos.
- CFOS logró independencia para varias implementaciones de parámetros como el número de sondas, iteraciones, factor de reposición y gamma, propios de CFO. Dado que estos factores no tienen actualmente una forma de decisión teórica (se toman en base a prueba y error) el híbrido mejora de forma indirecta los problemas propios de una topología sin una fundamentación teórica desarrollada.
- CFOS logró llegar a respuestas cercanas y mejores a las de UPSO, PSO y NRM, siendo en todos los casos el espacio de búsqueda usado por el híbrido mayor al de las demás implementaciones, lo cual sugiere una potencialidad para encontrar óptimos con un intervalo de respuesta factible desconocido.
- El método de las raíces reales funciona para resolver sistemas de ecuaciones no lineales mediante el uso de una estrategia de optimización, siendo extendido a la solución de circuitos electrónicos en DC.
- CFOS es una estrategia determinística que no se ve afectada por el punto de inicio del simplex (principal desventaja de esta estrategia) debido al complemento de los dos algoritmos por separado: un determinístico de mapeo (CFO) y un determinístico de finalización (Simplex).

## BIBLIOGRAFIA

- [1] J. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, 1965.
- [2] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, N. York, and P. Chester, "Minimization or Maximization of Functions," in *Numerical Recipes in C The art of Scientific Computing*, 2nd ed., 1992, pp. 394–444.
- [3] X. Hongfeng and T. Guanzheng, "High-Dimension Simplex Genetic Algorithm and Its Application to Optimize Hyper-high Dimension Functions," *2009 WRI Global Congress on Intelligent Systems*, pp. 39–43, 2009.
- [4] R. Correa, I. Amaya, and J. Cruz, "Real Roots of nonlinear systems of equations through a metaheuristic algorithm," *Revista DYNA*, pp. 15–23, 2011.
- [5] R. A. Formato, "Central Force Optimization: A New Nature Inspired Computational Framework for Multidimensional Search and Optimization," vol. 238, no. 2008, pp. 221–238.
- [6] P. In, "CENTRAL FORCE OPTIMIZATION: A NEW METAHEURISTIC WITH APPLICATIONS IN APPLIED ELECTROMAGNETICS R. A. Formato," pp. 425–491, 2007.
- [7] G. M. Qubati, R. A. Formato, and N. I. Dib, "Antenna benchmark performance and array synthesis using central force optimisation," vol. 4, no. April 2009, pp. 583–592, 2010.
- [8] R. A. Formato, "Improving Bandwidth of Yagi-Uda Arrays," vol. 2012, no. January, pp. 18–24, 2012.
- [9] R. A. Formato, "Variable Z 0 -," vol. 0, pp. 18–23, 2012.
- [10] L. Wang and M. Alam, "Training Neural Networks using Central Force Optimization and Particle Swarm Optimization: Insights and Comparisons."
- [11] G. M. Qubati, R. a. Formato, and N. I. Dib, "Antenna benchmark performance and array synthesis using central force optimisation," *IET Microwaves, Antennas & Propagation*, vol. 4, no. 5, p. 583, 2010.
- [12] R. C. Green, L. Wang, M. Alam, and R. A. Formato, "Central Force Optimization on a GPU: A case study in high performance metaheuristics using multiple topologies," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, 2011, no. 2, pp. 550–557.
- [13] R. Formato, "Parameter-free deterministic global search with simplified central force optimization," *Advanced Intelligent Computing Theories and ...*, pp. 309–318, 2010.
- [14] R. P. Attorney and C. Engineer, "CENTRAL FORCE OPTIMIZATION APPLIED TO THE PBM SUITE OF ANTENNA BENCHMARKS Richard A. Formato," vol. 425, pp. 405–425, 2010.
- [15] R. A. Formato, "Pseudorandomness in Central Force Optimization," vol. 2, no. February 2010, pp. 1–92.
- [16] G. B. Dantzig, A. Orden, and P. Wolfe, "The generalized simplex method for minimizing a linear form under linear inequality restraints," *Pacific Journal of Mathematics*, vol. 5, no.

2, pp. 183–195, 1955.

- [17] R. Chelouah and P. Siarry, “A hybrid method combining continuous tabu search and Nelder–Mead simplex algorithms for the global optimization of multim minima functions,” *European Journal of Operational Research*, vol. 161, no. 3, pp. 636–654, Mar. 2005.
- [18] H. Lin and K. Yamashita, “Hybrid simplex genetic algorithm for blind equalization using RBF networks,” *Mathematics and Computers in Simulation*, vol. 59, no. 4, pp. 293–304, Jun. 2002.
- [19] J. Cruz, “SOLUCIÓN DEL MODELO MATEMÁTICO DE UN CIRCUITO ELECTRÓNICO D.C. NO LINEAL MEDIANTE UNA ESTRATEGIA DE OPTIMIZACIÓN,” Universidad Industrial de Santander, 2012.
- [20] O. Roa and F. Ramirez, “SOLUCIÓN DE LAS ECUACIONES QUE DESCRIBEN EL MODELO MATEMÁTICO DE UN CIRCUITO ELECTRÓNICO COMPUESTO DE ELEMENTOS NO LINEALES MEDIANTE EL OPTIMIZADOR DE FUERZA CENTRAL,” Universidad Industrial de Santander, 2012.
- [21] K. R. Mahmoud, “Central Force Optimization: Nelder-Mead Hybrid Algorithm for Rectangular Microstrip Antenna Design,” *Electromagnetics*, vol. 31, no. 8, pp. 578–592, Nov. 2011.
- [22] R. Formato, “Central force optimization: A new nature inspired computational framework for multidimensional search and optimization,” *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)*, vol. 238, no. 2008, pp. 221–238, 2008.
- [23] J. Cruz, R. Correa, and I. Amaya, “SOLUCIÓN DEL MODELO MATEMÁTICO DE UN CIRCUITO ELECTRÓNICO NO LINEAL EN CORRIENTE DIRECTA MEDIANTE OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS,” *Revista DYNA*, pp. 77–84, 2012.
- [24] R. Baldick, “Applied Optimization: Formulation and Algorithms for Engineering Systems,” *First. Cambridge: Cambridge University Press*, p. 786, 2006.

## ANEXOS

### Anexo A: Instrucciones de uso CFOS

Para utilizar el código realizado es necesario agregar al path de MATLAB® u Octave las ubicaciones de las carpetas “Core” y “Funciones Objetivo” (Estos archivos se dejan al interior del grupo CEMOS y pueden ser solicitados a los autores al mail [reyes2000jose@gmail.com](mailto:reyes2000jose@gmail.com) [diferco99@gmail.com](mailto:diferco99@gmail.com) ).

Dentro de la carpeta Core existe un archivo denominado “cfosimplex” que trae guardados los punteros a las diferentes funciones del programa. A manera de ejemplo se deja, consignado un script básico de aplicación donde se explica la manera de optimizar la función de la esfera modificada con CFOS:

```
clear all; close all; clc;
tic
Np=100;           %Numero de sondas a enviar
Nd=2;            %Numero de dimensiones
Nt=100;          %Numero de paso de tiempo
G=2;             %Aceleracion Gravitacional
alpha=2;         %Factor alpha propio CFO
beta=2;          %Factor beta propio CFO
Frep_ini=0.5;    %Factor de reposicion inicial
deltaFrep=0;     %Delta Factor de reposicion
funcion='Modified_sphere'; %Funcion a optimizar
gamma=0;         %Valor de gamma
Xmi=-30*ones(1,Nd); %Minimo del Espacio de busqueda
Xma=30*ones(1,Nd); %Maximo del espacio de busqueda

%Llamando algoritmo CFOS

[fitness, punto] = cfosimplex(alpha,beta, G, gamma, Np, Nd, Nt,
Frep_ini, deltaFrep, Xmi, Xma, funcion);

fitness
punto
toc
```

## Anexo B: PSEUDOCÓDIGO PARA IMPLEMENTACIÓN DE CFO

### 1. Inicialización de Datos :

- 1.1 Crear los arreglos de posición,  $R(p, i, j)$  y aceleración  $A(p, i, j)$ ,  
donde  $1 \leq p \leq N_p$ ,  $1 \leq i \leq N_d$ ,  $0 \leq j \leq N_t$ ,  
siendo  $[N_p, N_d, N_t] = [\text{total sondas}, \text{total dimensiones}, \text{total Pasos de tiempo}]$ .
- 1.2 Crear los arreglos de evaluación o fitness:  $M(p, j) = f(R(p, i, j))$ .
- 1.3 Hallar el mejor fitness,  $M_{best}$

### 2. Inicialización

En el paso  $j=0$

- 2.1 Sondas distribuidas uniformemente en cada eje de coordenadas.  
( $\frac{N_p}{N_d}$ , número de pruebas / dimensión, es especificado por el usuario)

Encontrar punto de intersección de las sondas :

For  $i=1:N_d$

For  $p=1:N_p$

$$R(p, i, 0) = x_i^{min} + \gamma \cdot (x_i^{max} - x_i^{min})$$

end

end

For  $i=1$  to  $N_d$ ,  $n=1$  to  $\frac{N_p}{N_d}$  :

$$p = n + \frac{(i-1) \cdot N_p}{N_d}, \quad R(p, i, 0) = x_i^{min} + \frac{(n-1)c}{\frac{N_p}{N_d} - 1}$$

### 2.2 Aceleración inicial

$$A(p, i, 0) = 0, \quad 1 \leq p \leq N_p, \quad 1 \leq i \leq N_d$$

### 2.3 Fitness inicial

$$M(p, 0) = f(R(p, i, 0)), \quad 1 \leq p \leq N_p, \quad 1 \leq i \leq N_d$$
$$\text{Mejor Fitness} = \text{MAX}(M(p, 0), \quad 1 \leq p \leq N_p)$$

### 2.4 Factor de reposicionamiento e incremento

$$F_{rep}^{init}, \Delta F_{rep} \quad (\text{determinados por el usuario})$$

### 3. Lazo de tiempo

#### 3.1 Calcular las nuevas posiciones de las sondas:

for  $p=1$  to  $N_p$ ,  $i=1$  to  $N_d$ :

$$R(p, i, j) = R(p, i, j-1) + \frac{1}{2} A(p, i, j-1) \Delta t^2, \Delta t^2 = 1$$

#### 3.2 Regresar las sondas:

if  $R(p, i, j) > x_i^{max}$  then

$$R(p, i, j) = x_i^{min} + F_{rep} (R(p, i, j-1) - x_i^{min})$$

if  $R(p, i, j) < x_i^{min}$  then

$$R(p, i, j) = x_i^{max} - F_{rep} (R(p, i, j-1) - x_i^{max})$$

#### 3.3 Actualizar fitness para iteración actual:

$$\text{for } p=1:N_p : M(p, j) = f(R(p, i, j))$$

#### 3.4 Actualizar aceleración para la siguiente iteración:

for  $p=1$  to  $N_p$ ,  $i=1$  to  $N_d$ :

$$A(p, i, j) = \sum_{k=1, k \neq p}^{N_p} U(M(k, j) - M(p, j)) \cdot (M(k, j) - M(p, j))^\alpha \frac{R(k, i, j) - R(p, i, j)}{|(R(k, i, j) - R(p, i, j))^\beta|}$$

donde

$$|(R(k, i, j) - R(p, i, j))^\beta| = \sqrt{\sum_{m=1}^{N_d} (R(k, m, j) - R(p, m, j))^2}$$

#### 3.5 Decrecer el factor de reposición

$$F_{rep} = F_{rep} - \Delta F_{rep}; \text{ if } F_{rep} \leq 0 \text{ then } F_{rep} = F_{rep}^{init}$$

#### 3.6 Si han pasado los primeros 20 pasos de tiempo $\rightarrow$ actualizar cada 10 pasos el espacio de búsqueda:

if  $j=20$  &&  $j \bmod 10 = 0$ :

$$x_i'^{min} = x_i^{min} + \frac{\vec{R}_{best} \cdot \hat{e}_i - x_i^{min}}{2}; \quad x_i'^{max} = x_i^{max} - \frac{x_i^{max} - \vec{R}_{best} \cdot \hat{e}_i}{2}$$

4. Aumentar el lazo de tiempo,  $j = j + 1$ , y repetir los pasos hasta que se cumpla el número de iteraciones o el criterio de parada.