

**SOLUCIÓN DE UN SISTEMA DE ECUACIONES NO LINEALES, UTILIZANDO
UNA ESTRATEGIA BASADA EN EL ALGORITMO DE HARMONY SEARCH**

JHONATAN CONTRERAS DUARTE

CARLOS ANDRES VILLANUEVA VASQUEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS

ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA

Y DE TELECOMUNICACIONES

BUCARAMANGA

2013

**SOLUCIÓN DE UN SISTEMA DE ECUACIONES NO LINEALES, UTILIZANDO
UNA ESTRATEGIA BASADA EN EL ALGORITMO DE HARMONY SEARCH**

**JHONATAN CONTRERAS DUARTE
CARLOS ANDRÉS VILLANUEVA VASQUEZ**

**Trabajo de grado para optar al título de
Ingeniero Electrónico**

**DIRECTOR
PhD. CARLOS RODRIGO CORREA CELY**

**CODIRECTOR
PhD(c) IVAN MAURICIO AMAYA**

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

2013

A Dios a MA, a mis padres y hermanas, a Don Jaime y el servicio de comedores
quienes finalmente son los responsables de un gran número de grados.

J.C.D

TABLA DE CONTENIDO

INTRODUCCIÓN	13
1 DESCRIPCIÓN DEL TRABAJO DE GRADO	14
1.1 OBJETIVO GENERAL	14
1.2 OBJETIVOS ESPECIFICOS.....	14
2 FUNDAMENTOS TEORICOS	15
2.1 TEOREMA DE RAICES REALES.....	15
2.2 Harmony Search original.....	16
2.3 HS FRENTE A OTRAS ESRATEGIAS DE OPTIMIZACION	19
2.4 <i>IMPROVED HARMONY SEARCH (IHS)</i>	19
2.5 <i>HARMONY SEARCH CON ANCHO DE BANDA AJUSTABLE (ABHS)</i>	21
3 COMPARACION HS, IHS Y ABHS (V1 Y V2)	26
3.1 PRUEBA FUNCION ORDEN 2	27
3.2 PRUEBA FUNCION ORDEN 6.....	30
3.3 PRUEBA FUNCION ORDEN 10.....	33
3.4 ANALISIS DE RESULTADOS.....	34
4 EVALUACIÓN DEL ALGORITMO A TRAVES DE EJEMPLOS	35
4.1 FUNCIÓN DE PRUEBA UNO	36
4.2 FUNCIÓN DE JONG	38
4.3 FUNCIÓN DEL VALLE ROSENBROCK.....	41
4.4 FUNCIÓN DE WOOD	43
4.5 FUNCIÓN DE BOOTH	44
5 EXPERIMENTOS CON CIRCUITOS CON COMPONENTES NO LINEALES	47
5.1 CIRCUITO DE PRUEBA UNO	47
5.2 CIRCUITO DE PRUEBA DOS	51
5.3 CIRCUITO DE PRUEBA TRES	54
5.4 CIRCUITO ESPEJO DE CORRIENTE WIDLAR.....	57
6 CONCLUSIONES	60
7 RECOMENDACIONES	61
8 REFERENCIAS BIBLIOGRÁFICAS	62
9 ANEXOS	64
9.1 ANEXO A.....	64
9.2 ANEXO B.....	66

LISTA DE FIGURAS

Figura 1. Diagrama de flujo para el Algoritmo Harmony Search	18
Figura 2. Variación del HCMR con respecto al número de iteraciones	22
Figura 3 Variación del PAR con respecto al número de iteraciones	22
Figura 4. Versión 1 de la variación del BW con respecto al número de iteraciones.	23
Figura 5. Versión 2 de la variación del parámetro ancho de banda BW.	24
Figura 6. Versión 3 de la variación del ancho de banda BW	24
Figura 7. Variación 1 del ancho de banda BW con respecto al número de iteraciones.....	25
Figura 8. Variación 2 del parámetro ancho de banda con respecto al número de iteraciones	26
Figura 9. Gráfico 3D de función de prueba 1	36
Figura 10. Gráfico 3D de función de Jong.....	39
Figura 11. Búsqueda de la convergencia por parte del algoritmo	40
Figura 12. Grafica 3D de función de valle de Rosenbrock	43
Figura 13. Gráfica 3D con de la función de Booth.	46
Figura 14. Búsqueda del punto de convergencia por parte del algoritmo.	46
Figura 15. Circuito de prueba 1.	47
Figura 16. Circuito de prueba 2	51
Figura 17. Circuito de prueba conversor DC-DC	55
Figura 18. Espejo de corriente Widlar	57

LISTA DE TABLAS

Tabla 1 Características de equipo de cómputo donde se realiza las pruebas.....	26
Tabla 2. Datos para el HS original con función de prueba 1	27
Tabla 3. Datos para HS mejorado con función de prueba 1	28
Tabla 4. Datos para versión 1 del ABHS con función de prueba 1	29
Tabla 5. Datos para la versión 2 del ABHS con función de prueba 1	29
Tabla 6. Datos para HS original con función de Jong.....	30
Tabla 7. Datos para HS mejorado con función de Jong	31
Tabla 8. Datos para la versión 1 del HS con ancho de banda adaptativo con función de Jong.....	32
Tabla 9. Datos para la versión 2 del HS con ancho de banda adaptativo con función de Jong.....	32
Tabla 10 Datos para HS mejorado con función Rastrigin	33
Tabla 11 Datos ABHS con función Rastrigin.....	33
Tabla 12 Datos para HS mejorado con función Rastrigin mejor resultado encontrado	34
Tabla 13. Características de equipo de cómputo donde se realiza las pruebas.	35
Tabla 14. Parámetros por defecto.	36
Tabla 15. Datos correspondientes a la función de prueba 1.	37
Tabla 16. Datos correspondientes a la función de Jong.	39
Tabla 17. Datos correspondientes a la función de Rosenbrock	42
Tabla 18. Datos correspondientes a la función de Wood.	44
Tabla 19. Datos correspondientes a la función de Booth.	45
Tabla 20. Valores de los componentes del circuito para el ensayo 1	48
Tabla 21. Datos para el circuito de prueba 1.	49
Tabla 22. Tabla comparativa para los siguientes métodos.....	49
Tabla 23. Valores de los componentes del circuito de prueba 1 para el ensayo 2.....	50
Tabla 24. Datos para el circuito de prueba 1 variando V_t e I_s	51
Tabla 25. Valores de los componentes del circuito de prueba 2 para el ensayo 1	52
Tabla 26. Datos para el circuito de prueba 2 variando el valor del parámetro HMS.....	53
Tabla 27. Valores de los componentes del circuito de prueba 2 para el ensayo 2.	53
Tabla 28. Datos para el circuito de prueba 2 variando $q/(n \cdot k_b \cdot t)$	54
Tabla 29. Valores de los componentes del circuito conversor DC-DC.....	55
Tabla 30 Datos tomados por parte del algoritmo ABHS V2 para el circuito de prueba 3 ..	57
Tabla 31. Parámetros constantes para el espejo de corriente	58
Tabla 32. Datos para prueba 1 del espejo de corriente Widlar	59
Tabla 33. Datos para ensayo 2 del espejo de corriente Widlar	59
Tabla 34. Datos para ensayo 3 del espejo de corriente Widlar	60
Tabla 35. Circuito de prueba uno CFO1.....	64
Tabla 36. Circuito de prueba uno CFO2.....	64
Tabla 37. Circuito de prueba uno CFOFORM.....	65
Tabla 38 resultados función Griewank orden 100 ABHS V2	66

RESUMEN

TÍTULO: SOLUCIÓN DE UN SISTEMA DE ECUACIONES NO LINEALES, UTILIZANDO UNA ESTRATEGIA BASADA EN EL ALGORITMO DE HARMONY SEARCH.

AUTORES: Jhonatan Contreras Duarte y Carlos Andrés Villanueva Vásquez.

PALABRAS CLAVES: determinístico, metaheurístico, no lineal, Harmony Search

CONTENIDO:: En este trabajo de grado se muestran los resultados obtenidos en la solución de algunos modelos matemáticos que representan circuitos electrónicos no lineales en corriente directa (D.C) y de problemas de optimización comúnmente usados para comprobar métodos numéricos, utilizando un método basado en el algoritmo metaheurístico Harmony Search. Asimismo, se utilizó el teorema de Raíces Reales, para transformar un sistema de ecuaciones no lineales en un problema de optimización.

En primer lugar, el algoritmo Harmony Search original se comparó con algunas de sus variantes utilizando como medio de comparación algunas funciones de optimización estándares con resultados conocidos. Después de este paso, se eligió el método que mejores resultados obtuvo con respecto al tiempo y la precisión, este algoritmo fue llamado Harmony Search con Ancho de banda Ajustable o Adjustable Bandwidth Harmony Search por sus siglas en inglés, Este algoritmo seleccionado se evaluó con cinco funciones de prueba más con el fin de verificar su efectividad. Posterior de la evaluación, este método se implementó en la solución de algunos sistemas que describen el comportamiento de los circuitos electrónicos no lineales en DC. Logrando resolver todos los problemas propuestos. Nuestros resultados muestran que Harmony Search es un método viable para resolver circuitos no lineales en DC

ABSTRACT

TITLE: SOLUTION TO A SYSTEM OF NONLINEAR EQUATIONS, USING A STRATEGY BASED ON THE HARMONY SEARCH ALGORITHM.

AUTHORS: Jhonatan Contreras Duarte y Carlos Andrés Villanueva Vásquez.

KEYWORDS: Deterministic, metaheuristic, nonlinear, Harmony Search

DESCRIPTION: In this paper we show the results obtained when solving some mathematical models representing nonlinear electronic circuits with direct current (D.C). We further investigate optimization problems commonly found with numerical methods. Using a method based on Harmony Search metaheuristic algorithm. Likewise, we used the real roots theorem to transform a system of non-linear equations in an optimization problem.

Firstly, the original Harmony Search algorithm was compared with some of its variants comparison means using as some optimization functions standards with known results. After this step, we chose the best performing method with respect to time and accuracy. This algorithm was Adjustable Bandwidth Harmony Search. The selected algorithm was evaluated with five more test functions in order to verify its effectiveness. Following the evaluation, this method was implemented in solving some systems that describe the behavior of nonlinear electronic circuits in DC. Our results show that the Harmony Search metaheuristic method is a viable approach in solving nonlinear circuits in DC.

INTRODUCCIÓN

Los métodos numéricos tradicionales de solución de sistemas de ecuaciones no lineales se basan en algoritmos que hacen uso de operaciones matriciales, cuyos elementos pertenecen a las derivadas de sus funciones. Una estrategia diferente consiste en convertir un sistema de ecuaciones no lineales en un problema de optimización, ya que estos pueden ser resueltos por métodos determinísticos, estocásticos y metaheurísticos, presentando diversas variaciones y enfoques. Una técnica determinística que puede ser utilizada para este fin es Newton-Raphson y sus variantes. Estos métodos, pese a ser muy utilizados, tienen la desventaja de requerir mucho tiempo para converger, y sólo ofrecen una raíz (en su forma tradicional). Adicionalmente tiene un elevado costo computacional, debido, principalmente, al cálculo y almacenamiento del Jacobiano (derivadas de primer orden) y del Hessiano (derivadas de segundo orden), matrices que deben ser evaluadas en cada punto candidato para cada iteración, por lo que la cantidad de memoria necesaria para obtener una nueva dirección de búsqueda es elevada. Además, estos métodos son fácilmente atrapados en mínimos locales, o en puntos de silla, por lo que se requiere un punto de partida cerca de la solución, lo que limita su aplicación en situaciones de la vida real.

Una estrategia diferente de optimización apareció hace algún tiempo y se conoce como algoritmo metaheurístico. Éste hace uso de procesos estocásticos, que rara vez son atrapados por la presencia de mínimos locales. Uno de estos métodos, cuya popularidad ha ido en aumento es "*Harmony Search*" (HS), (Búsqueda Armónica) utilizado extendidamente para la optimización, desarrollado en 2001 por Zong Woo Geem, Joong Hoon Kim y G. V. Loganathan, inspirado en el proceso de composición de los músicos, mediante el cual, generan armonías a partir de la improvisación. Durante esta investigación se utilizó el algoritmo para encontrar las raíces de sistemas no lineales.

1 DESCRIPCIÓN DEL TRABAJO DE GRADO

Este informe de investigación se realizó con el fin de recopilar los resultados obtenidos y evidenciar el cumplimiento de los objetivos, que se retoman a continuación.

1.1 OBJETIVO GENERAL

Analizar y desarrollar un algoritmo fundamentado en la estrategia de optimización “*Harmony Search*” orientado a la solución de un sistema de ecuaciones no lineales, presente en un circuito electrónico específico tomado como ejemplo.

1.2 OBJETIVOS ESPECIFICOS

El cumplimiento del objetivo general del trabajo de grado comprende:

1. Implementar el algoritmo de optimización “Harmony Search” para solucionar el sistema de ecuaciones que describe el modelo de un circuito electrónico seleccionado.
2. Analizar la respuesta del sistema “Harmony Search” ante la variación de los parámetros que definen el (los) componente(s) no lineal(es) del circuito electrónico tomado como ejemplo.
3. Se contrastará la eficacia del algoritmo de optimización Harmony Search, en la solución del sistema de ecuaciones que describe un circuito electrónico no lineal, utilizando medidas de control como el tiempo de cómputo, la convergencia del método y la estimación de error, con los resultados obtenidos previamente con otro método numérico.

2 FUNDAMENTOS TEORICOS

El algoritmo Harmony Search (HS) es un algoritmo meta heurístico inspirado en el proceso de composición de los músicos, en donde estos generan armonías a partir de la improvisación. Fue desarrollado en 2001 por Zong Woo Geem, Joong Hoon Kim y G. V. Loganathan [1].

La improvisación musical es un proceso utilizado por toda clase de músico, en el cual la experiencia y el conocimiento de armonías previamente utilizadas aportan a la calidad de la pieza que se está tocando. Un buen músico recuerda las mejores notas y descarta aquellas cuyo resultado no haya sido el deseado.

Para poder resolver un sistema de ecuaciones con un algoritmo de optimización, es necesario modificar el sistema, transformándolo en una función objetivo a través del teorema de las raíces reales.

2.1 TEOREMA DE RAICES REALES

Se supone un sistema no lineal de ecuaciones con m funciones e igual número de incógnitas como se observan en las ecuaciones (1), (2), (3).

$$F_1(x_1, x_2, x_3 \dots, x_m) = 0 \quad (1)$$

$$F_2(x_1, x_2, x_3 \dots, x_m) = 0 \quad (2)$$

...

$$F_m(x_1, x_2, x_3 \dots, x_m) = 0 \quad (3)$$

Sea $X \in R^n$ y vector solución del sistema de ecuaciones, Considerando el siguiente sistema:

$$\text{Sistema} \begin{cases} f_1(x) = 0 \\ f_2(x) = 0 \\ \vdots \\ f_m(x) = 0 \end{cases}$$

Donde para cada $i = 1, 2, \dots, m$, f_i es una función cuyo dominio contiene a X , y con rango en los reales. Sea $f: X \rightarrow \mathbb{R}$ definida como sigue

$$f(x) = \sum_{i=1}^m (f_i(x))^2 \quad (4)$$

Siendo $X = (x_1, x_2, \dots, x_m)$, vector solución del sistema de ecuaciones, existe un $a \in X$ donde $a = (a_1, a_2, a_3, \dots, a_n)$, entonces a satisface el sistema si y solo si a minimiza a $f(x)$.

Entonces $f_i(a) = 0$ para cada $i = 1, 2, \dots, m$ luego $f(a) = 0$ y como $f(x) \geq 0$ para todo vector $z \in X$, entonces a es un punto de mínimo para f .

2.2 Harmony Search original

El algoritmo HS es un algoritmo metaheurístico, que utiliza de forma adecuada los conceptos de explorar el espacio de búsqueda, e intensificar en las mejores soluciones previamente halladas, para encontrar la solución.

El desarrollo del algoritmo se lleva a cabo en cinco pasos que serán enumerados a continuación.

Paso 1. Inicializar los parámetros del problema y los parámetros de HS

Definir los límites superior (LS) e inferior (LI), tal que $LI < x_i < LS$, donde (x_i) , es una solución candidata que pertenece al vector solución de N variables.

Además de los límites, los parámetros que se especifican en este paso son el tamaño de la memoria *Harmony Memory (HM)*, *Harmony Memory Size (HMS)*, la tasa de consideración de la memoria HM, llamado HMCR (*Harmony Memory Considering Rate*), la tasa de ajuste del tono PAR (*Pitch Adjust Rate*), el ancho de banda de ajuste de tono BW y el número de improvisaciones NI.

Paso 2 Inicializar la memoria HM

La HM inicial se genera a partir de una distribución uniforme en los rangos $[LS, LI]$, se realiza de la siguiente manera $x_i = LI + rand(LS - LI)$ donde $rand$ es un número aleatorio entre cero (0) y uno (1)

Paso 3 improvisar una nueva armonía

El proceso de generación de un nuevo vector solución es llamado improvisación, el nuevo vector se genera utilizando las siguientes reglas: consideración de la memoria HMCR, ajuste de tono BW, selección aleatoria PAR

Paso 4 Actualizar la memoria HM

La nueva armonía generada, reemplazará la peor armonía almacenada en la memoria si el valor de la función objetivo es mejor que esta.

Paso 5 verificar el criterio de parada

El algoritmo termina cuando el número máximo de improvisaciones alcanza NI.

Los parámetros HMCR y PAR ayudan al método en la búsqueda de mejores soluciones globales y locales respectivamente, PAR y BW tienen un efecto directo en el desempeño del algoritmo tanto en la búsqueda de la respuesta como en el tiempo de cómputo y es por esto que el ajuste de estos parámetros es muy

importante el valor de estos parámetros varía dependiendo el área de búsqueda y el problema específico.

En la **¡Error! No se encuentra el origen de la referencia.** [3] se muestra el diagrama el funcionamiento del algoritmo HS

Figura 1. Diagrama de flujo para el Algoritmo Harmony Search

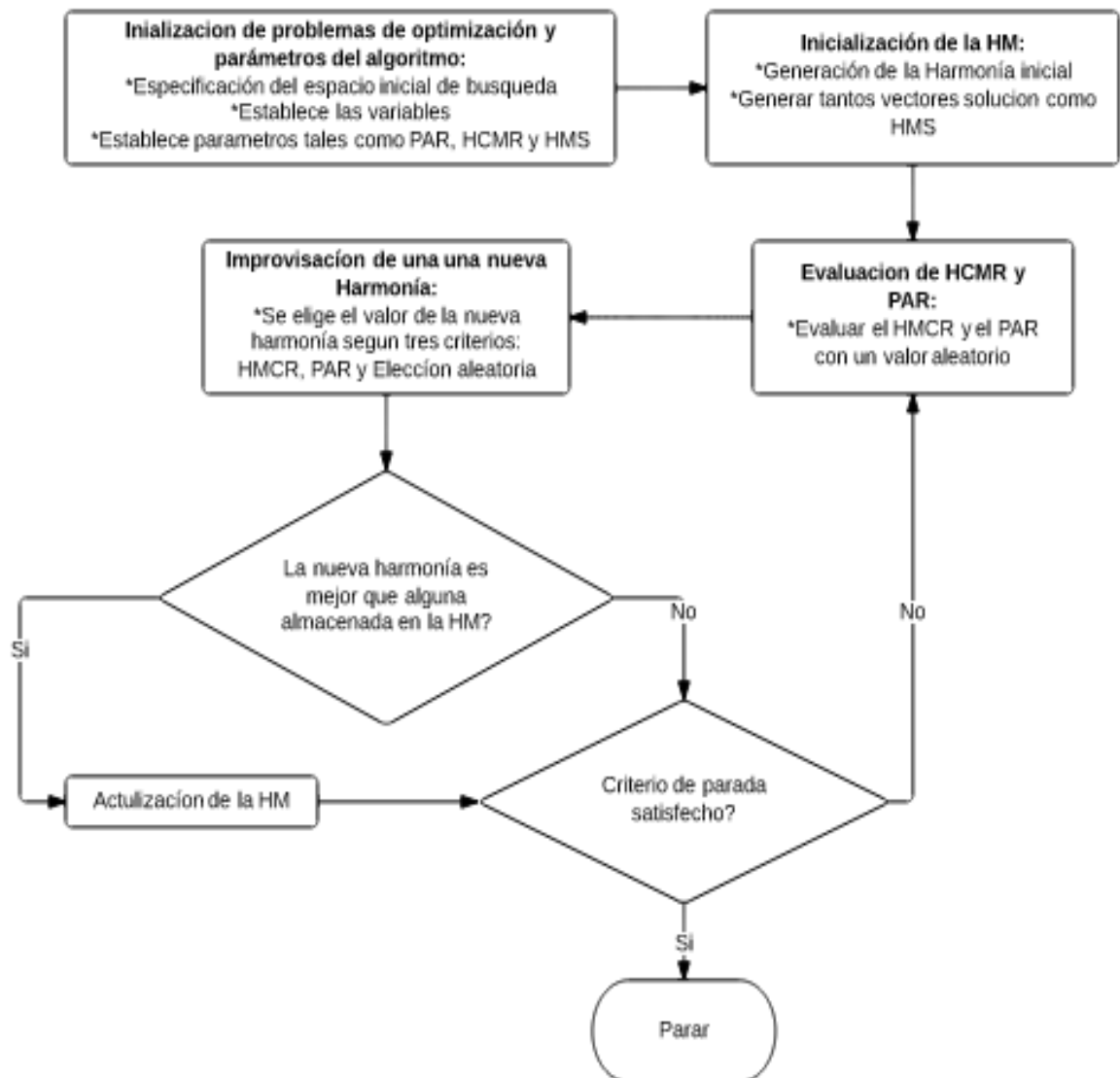


Ilustración 1

2.3 HS FRENTE A OTRAS ESTRATEGIAS DE OPTIMIZACION

Una ventaja de los algoritmos metaheurísticos se basa en su simplicidad, la cual puede ayudar a encontrar errores lógicos y solucionarlos de una mejor manera. La simulación de procesos naturales es razón por la cual estos algoritmos son preferidos frente a algoritmos que emplean cálculos y operaciones matemáticas [5].

Estos tipos de algoritmos hacen uso de técnicas de alto nivel y algunos procesos de ensayo y error para encontrar la solución a un problema determinado. Se destacan dos características principales: la *diversificación* y la *intensificación*, también conocidos como exploración y explotación, los cuales son conceptos opuestos. Con un adecuado balance entre los dos se establece el éxito de un algoritmo meta heurístico. La diversificación es el proceso por el cual el algoritmo explora en muchos lugares y regiones del espacio de soluciones, con el fin de evitar que el algoritmo no caiga en óptimos locales. La intensificación pretende aprovechar la historia y la experiencia del proceso de búsqueda asegurando la velocidad de convergencia, cuando sea necesario reduciendo la aleatoriedad y limitando la diversificación.

En HS la intensificación es controlada por el parámetro PAR y la diversificación por el parámetro HMCR, con lo cual se busca refinar los valores obtenidos, es decir encontrar nuevos valores cercanos a los mejores valores ya obtenidos. Mientras el algoritmo explora el espacio de búsqueda global de manera efectiva.

Un valor alto HMCR indica que se reutilizarán más los valores almacenados en la memoria, y si es demasiado bajo la convergencia se tornará demasiado lenta.

2.4 IMPROVED HARMONY SEARCH (IHS)

IHS (*Improved Harmony Search*) o *Búsqueda Armónica Mejorada* en su traducción al castellano fue propuesto en el año 2007 por Mangdavi et al [4]. A diferencia del HS original, genera nuevas armonías variando los parámetros PAR y BW logrando mejorar la precisión y la velocidad de convergencia, en comparación con el algoritmo original. Esta variante modifica el paso 3 del algoritmo original, cambiando PAR y BW dinámicamente con el número de iteraciones, que se calculan como se observa en las ecuaciones (5) y (6).

$$PAR = PAR_{min} + \frac{(PAR_{max} - PAR_{min})}{NI} * (iter) \quad (5)$$

Donde

PAR tasa de ajuste del tono para cada iteración

PARmin Tasa mínima de ajuste de tono

PARmax Tasa máxima de ajuste de tono

NI Numero de iteraciones máximas

Iter Número de iteración actual

$$Bw = Bw_{max} * e^{(c*iter)} \quad (6)$$

Donde

$$c = \frac{\ln\left(\frac{Bw_{min}}{Bw_{max}}\right)}{NI} \quad (7)$$

BW Ancho de banda

BWmin Ancho de banda mínimo

BWmax Ancho de banda máximo

El parámetro PAR crece linealmente con el número de iteraciones y el parámetro BW decrece exponencialmente en este algoritmo con respecto al número de iteraciones. El algoritmo IHS tiene como criterio de parada el número máximo de iteraciones en el cual PAR y BW obtienen sus valores máximos y mínimos respectivamente.

2.5 HARMONY SEARCH CON ANCHO DE BANDA AJUSTABLE (ABHS)

En este trabajo de grado, se propuso una modificación a la variante IHS, llamada ABHM (*Adjustable Bandwidth Harmony Search*), la principal razón de la transformación del código es la incertidumbre en el número de iteraciones necesarias para resolver un problema dado (por ejemplo, puede requerir 6000 iteraciones, como puede requerir 40000).

Los parámetros PAR y BW descritos en el algoritmo IHS dependen directamente del número de iteraciones, un adecuado número de iteraciones permite encontrar la respuesta buscada, al limitar el número de iteraciones, por debajo de las necesarias, no se puede garantizar una respuesta satisfactoria, por otro lado si estas se exceden los parámetros no variarán de forma adecuada, haciendo que el algoritmo demore más al encontrar la respuesta. La solución propuesta fue declarar un “*máximo de iteraciones*” en el cual se modifican los valores de PAR y de HMCR como se observan en la **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** se aclara que “*máximo de iteraciones*” es solo un valor en el algoritmo y no un criterio de parada, el cual nos dice el instante en que estos dos parámetros obtienen un valor fijo como en HS original.

Figura 2. Variación del HCMR con respecto al número de iteraciones

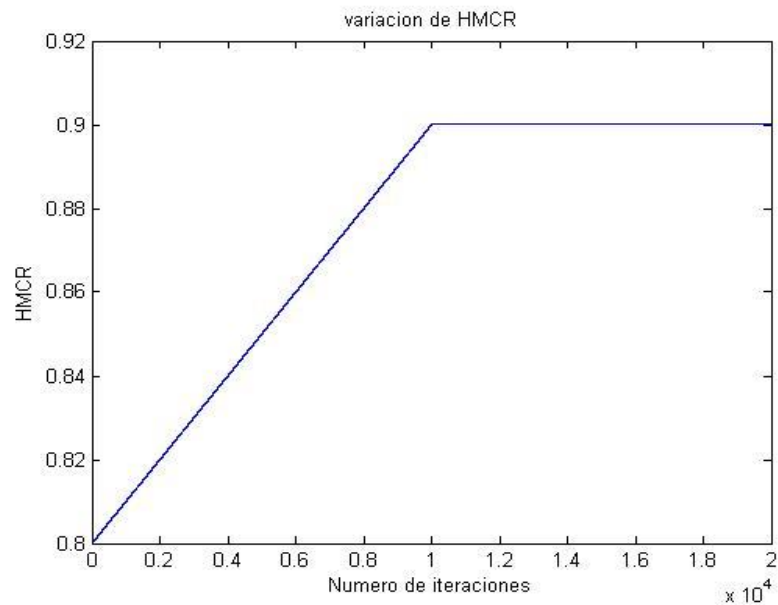
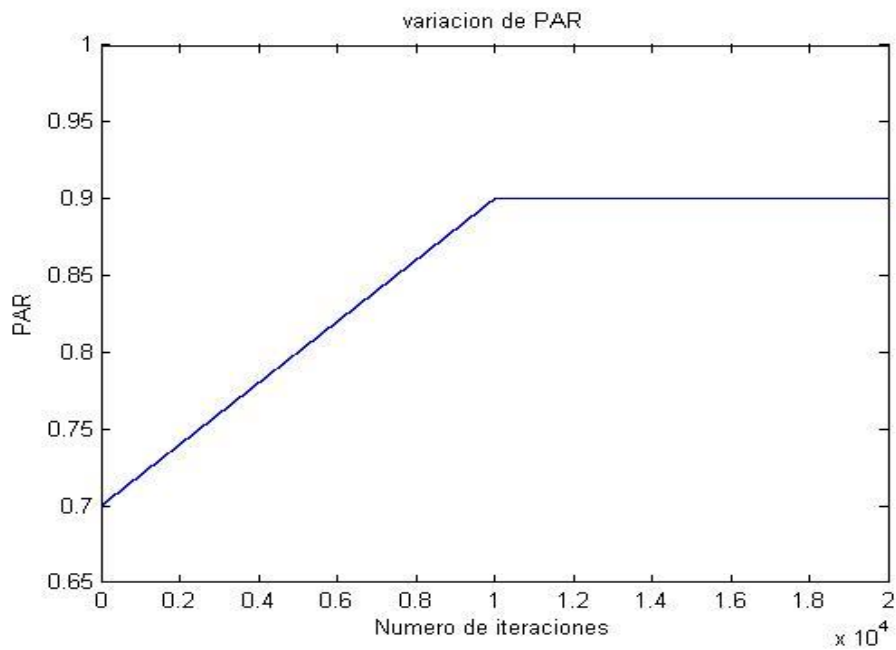


Figura 3 Variación del PAR con respecto al número de iteraciones



Los parámetros PAR y HCMR varían linealmente desde un valor mínimo hasta un valor máximo, el valor máximo se encuentra en un número de iteraciones fijo, a partir de este valor permanecen constantes.

El parámetro BW se ajustó de dos formas, la versión 1 (V1) y la versión 2 (V2), la primera opción V1, presenta como valor inicial y de retorno una constante de BW al igual que la versión original de H.S. posteriormente este valor cambiará con respecto a un nuevo parámetro llamado Parámetro de Saturación (PS), el cual indica el número de improvisaciones consecutivas las cuales no hayan encontrado una mejor melodía (vector solución), este parámetro junto a la exactitud constituyen los criterios de parada del algoritmo ABHS.

La V1 modifica el valor constante del BW cuando PS supere un umbral, la modificación de BW es de forma exponencial decreciente de la misma forma que la ecuación (6), partiendo desde un valor N veces BW inicial, hasta encontrar una mejor solución en ese instante BW toma el valor de retorno o valor inicial como se aprecia en las **¡Error! No se encuentra el origen de la referencia., ¡Error! No se encuentra el origen de la referencia.** y Figura 6, las tres figuras corresponden a diferentes instantes donde se encontró una mejor armonía

Figura 4. Versión 1 de la variación del BW con respecto al número de iteraciones.

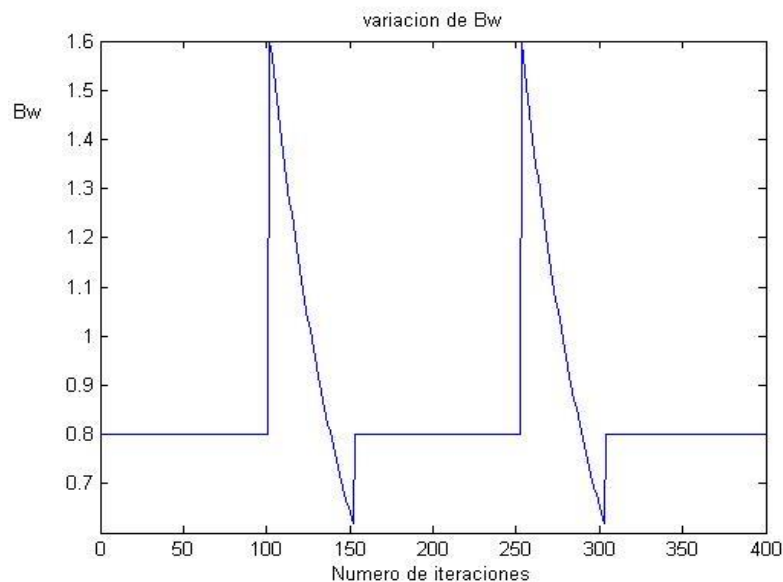


Figura 5. Versión 2 de la variación del parámetro ancho de banda BW.

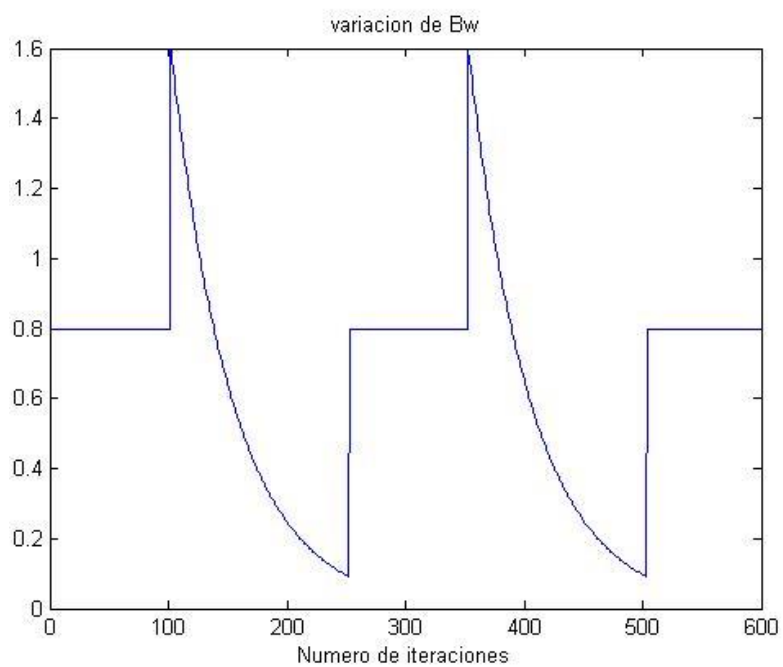
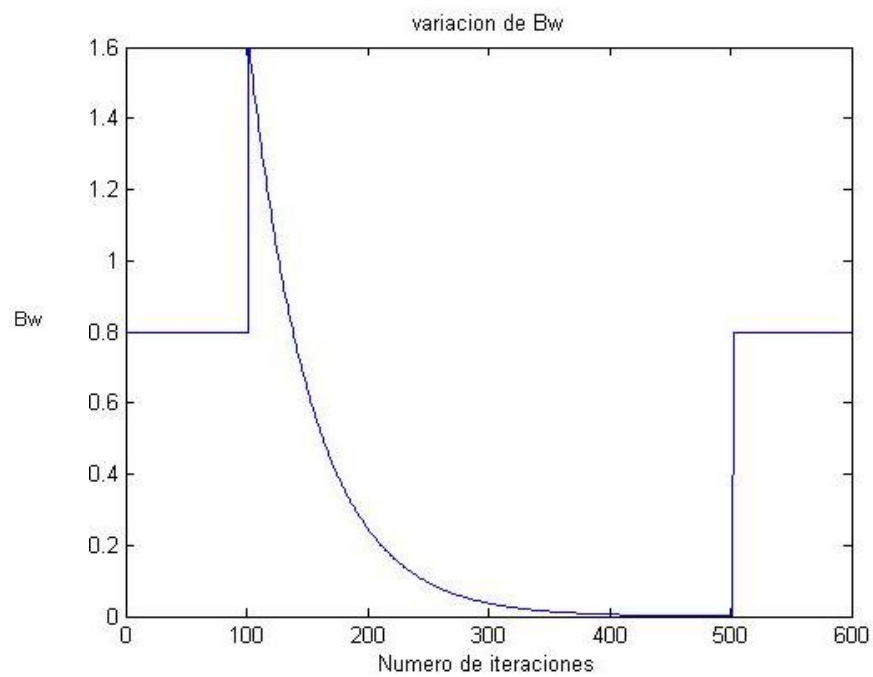


Figura 6. Versión 3 de la variación del ancho de banda BW



La V2 modifica el valor constante del BW cuando PS supere un umbral, Esta actualización se hace de forma exponencial decreciente, de la misma manera que en la ecuación (3) partiendo desde un valor N veces BW inicial, hasta encontrar una mejor solución, instante a partir del cual BW permanece constante hasta que PS vuelva a superar el valor umbral. En ese caso, BW volverá a modificarse, repitiendo el proceso las veces que sean necesarias, como se aprecia en las Figura 7 y **¡Error! No se encuentra el origen de la referencia..**

La necesidad de conocer el número de iteraciones de cada función en particular, fue la razón por la cual todas las pruebas fueron realizadas con el algoritmo ABHS; Se declaró una variable de observación (VO), la cual será un contador de las veces que el algoritmo pudo escapar de una inminente saturación.

Figura 7. Variación 1 del ancho de banda BW con respecto al número de iteraciones

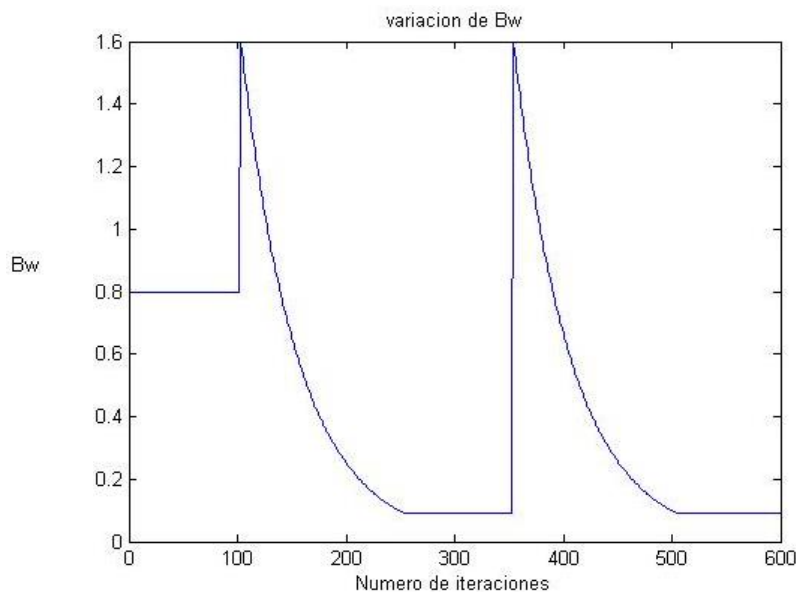
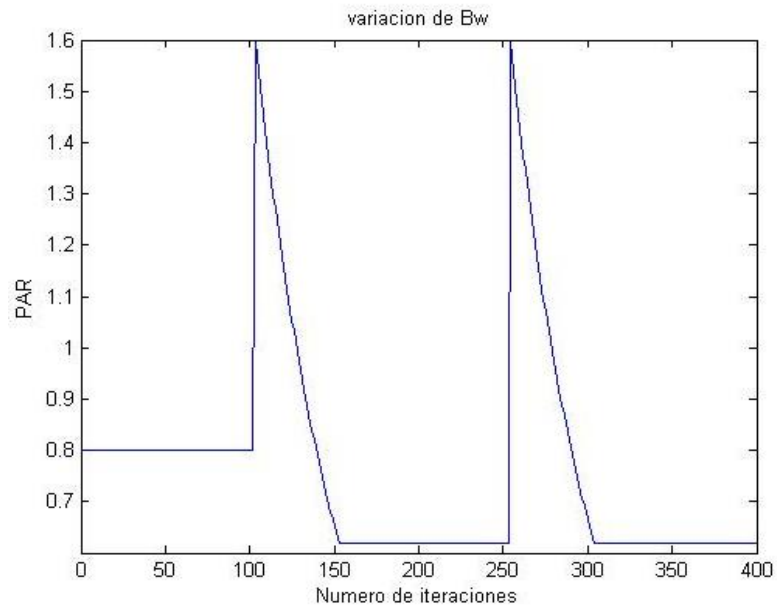


Figura 8. Variación 2 del parámetro ancho de banda con respecto al número de iteraciones



3 COMPARACION HS, IHS Y ABHS (V1 Y V2)

Para realizar estas pruebas se usó el equipo de cómputo con las especificaciones que se pueden observar en la Tabla 1

Tabla 1 Características de equipo de cómputo donde se realiza las pruebas.

Equipo de computo	
Procesador	Intel Core i3 @ 2.1GHz
Alimentación	Conectado a red eléctrica
Plan de energía	Alto rendimiento
RAM	3 Gb
SO	Windows 7 64 bits

Para comparar el rendimiento de las variantes de HS se realizaron dos pruebas con dos funciones no lineales. La primera prueba se realizó con una función de orden dos (2), mientras que la segunda se hizo con una de orden seis (6). En

ambas pruebas se tiene en cuenta el tiempo de cómputo, el número de iteraciones y el valor de la función evaluada, datos que se resumen de la Tabla 2 a la Tabla 5. Adicionalmente, se realizó una prueba de orden 10 entre IHS y ABHS V2.

3.1 PRUEBA FUNCION ORDEN 2

La ecuación 8 representa la función objetivo, en ella se puede ver que hay dos variables y que la función es altamente no lineal.

$$f(x, y) = x^2 + 2y^2 - 0.3 \cos(3\pi x) - 0.4 \cos(4\pi y) + 0.7 \quad [8]$$

- **HS original:**

Los parámetros utilizados en el *Harmony Search* original son HMCR=0.9; PAR=0.5; BW=0.001; HMS=20; NI=1000000; los resultados de estas pruebas se aprecian en la Tabla 2

Tabla 2. Datos para el HS original con función de prueba 1

Función de prueba uno		HS original	
Prueba	Tiempo de computo (s)	PS	F(x,y)
1	19,658409	134672	0,46988242
2	19,258351	250724	3,20E-10
3	19,618282	674095	1,43E-10
4	19,765609	6943	4,67E-10
5	19,606052	431130	1,13E-10
6	19,90971	812083	0,46988242
7	20,060832	504577	0,41292683
8	19,611443	607191	8,60E-11
9	19,752435	476613	0,46988242
10	19,782192	316516	0,41292683

- **IHS:**

Los parámetros utilizados en IHS fueron HMCRmin=0.8; HMCRmax=0.9; PARmin=0.7; PARmax=0.9; BWmax=0.8; BWmin=10⁻⁸; HMS=20; NI=1000000; los resultados de esta prueba se aprecian en la Tabla 3.

Tabla 3. Datos para HS mejorado con función de prueba 1

Función de prueba uno		Improved Harmony Search	
Prueba	Tiempo de computo (s)	Iteraciones	f(x,y)
1	9,088517	413384	-0,006628
2	9,775231	391901	2,70E-10
3	8,754137	375076	7,00E-10
4	9,140724	410672	2,03E-10
5	9,446393	394850	4,50E-10
6	8,569321	381129	2,96E-10
7	8,570365	378363	3,87E-10
8	7,587547	333229	4,91E-10
9	9,206457	410049	2,89E-10
10	8,931426	398396	5,98E-10

- **ABHS:**

Los parámetros utilizados por el ABHS V1 y V2 fueron HMCRmax=0.9; HMCRmin=0.8; PARmin=0.7; PARmax=0.9; BWmax=0.8; BWmin=10⁻¹²; HMS=20; los resultados de esta prueba se aprecian en la Tabla 4 y Tabla 5 respectivamente.

Tabla 4. Datos para versión 1 del ABHS con función de prueba 1

Función de prueba uno		ABHS V1		
Prueba	Tiempo de computo (s)	Iteraciones	VO	f(x,y)
1	0,594012	25143	21	2,72E-11
2	0,742662	32071	20	8,13E-11
3	0,422578	18340	8	9,82E-11
4	0,280304	12453	4	2,73E-11
5	0,363771	15875	7	6,55E-11
6	0,450264	19604	9	9,06E-11
7	0,338277	14768	6	2,04E-11
8	0,664697	28524	16	3,37E-11
9	0,672349	29047	23	8,03E-11
10	0,568938	24636	16	6,69E-11

Tabla 5. Datos para la versión 2 del ABHS con función de prueba 1

Función de prueba uno		ABHS V2		
Prueba	Tiempo de computo (s)	Iteraciones	VO	f(x,y)
1	0,699556	29882	23	9,23E-12
2	0,978694	41398	33	2,06E-11
3	0,753347	32311	23	5,69E-11
4	0,620138	26637	20	1,37E-11
5	0,85889	36619	22	6,18E-11
6	0,591277	24923	15	1,30E-11
7	0,803582	34324	28	9,78E-11
8	0,691143	29655	16	4,98E-11
9	0,774208	33080	32	9,95E-11
10	1,008119	42829	36	1,25E-11

3.2 PRUEBA FUNCION ORDEN 6

La ecuación (9) describe la función de JONG de orden 6, es una función no lineal con mínimo en $F(0,0,0,0,0,0)=0$, se realizaron pruebas y se tabularon los resultados

$$f(X) = \sum_{i=1}^n x_i^2 ; X = \{x_1, x_2, \dots, x_n\} \quad (9)$$

- **HS original:**

Los parámetros utilizados en el Harmony Search original son HMCR=0.9; PAR=0.5; BW=0.001; HMS=20; NI=1000000; los resultados de estas pruebas se aprecian en la Tabla 6.

Tabla 6. Datos para HS original con función de Jong

función Jong orden 6		HS original	
Prueba	Tiempo de computo (s)	saturación	f(x,y)
1	22.895.642	0	0,897296194
2	23.422.249	1	2,83E+01
3	23.101.039	153054	2,08E-08
4	22.783.326	205310	2,38E-08
5	22.802.469	89612	7,54E-08
6	22.965.822	364717	2,48E-08
7	22.716.112	313373	1,56E-08
8	23.151.449	417949	1,76E-08

- **IHS:**

Los parámetros utilizados en IHS fueron HMCRmin=0.8; HMCRmax=0.9; PARmin=0.7; PARmax=0.9; BWmax=0.8; BWmin=10⁻⁸; HMS=20; NI=1000000; los resultados de esta prueba se aprecian en la Tabla 7.

Tabla 7. Datos para HS mejorado con función de Jong

función Jong orden 6		IHS	
Prueba	Tiempo de computo (s)	Iteraciones	f(x,y)
1	12,671959	475555	5,75761E-10
2	12,456869	467011	9,9265E-10
3	12,795696	467227	8,17536E-10
4	13,403685	480047	6,79766E-10
5	13,637646	482834	9,76926E-10
6	13,216119	476463	6,45088E-10
7	13,600148	475461	8,77463E-10

- **ABHS:**

Los parámetros utilizados por el ABHS V1 y V2 fueron HMCRmax=0.9; HMCRmin=0.8; PARmin=0.7; PARmax=0.9; BWmax=0.8; BWmin=10⁻¹²; HMS=20; los resultados de esta prueba se aprecian en la

Tabla 8 y Tabla 9 respectivamente

Tabla 8. Datos para la versión 1 del HS con ancho de banda adaptativo con función de Jong

Función Jong orden 6				ABHS V1
Prueba	Tiempo de computo (s)	Iteraciones	VO	f(x,y)
1	1,572838	52828	31	7,8152E-11
2	1,66188	56125	42	8,1202E-11
3	1,787422	59717	36	9,8151E-11
4	2,120124	71329	42	8,2862E-11
5	1,994483	65946	35	7,2338E-11
6	1,886639	63195	38	5,8956E-11
7	1,889243	61703	42	8,9437E-11
8	1,733343	58325	35	8,173E-11

Tabla 9. Datos para la versión 2 del HS con ancho de banda adaptativo con función de Jong

función Jong grado 6				ABHS V2
Prueba	Tiempo de computo (s)	Iteraciones	VO	f(x,y)
1	0,831128	28060	16	8,7413E-11
2	1,153275	39166	12	7,7771E-11
3	0,922917	31406	15	7,5296E-11
4	1,152836	39176	14	9,4063E-11

5	1,083012	36794	19	8,2354E-11
6	1,028455	35178	13	9,193E-11
7	1,006702	34577	14	8,6223E-11
8	0,879768	30072	11	7,5529E-11

3.3 PRUEBA FUNCION ORDEN 10

Se realizó la prueba de la función Rastrigin la cual se observa en la ecuación (10), la prueba se realizó con esta función orden 10 con los dos métodos que presentaron mejores resultados hasta el momento IHS y ABHS V2, los resultados se observan en las Tabla 10,

Tabla 11,

Tabla 12.

$$f(X) = \sum_{i=1}^{10} x_i^2 - 10 \cos(2\pi x_i) + 10 ; X = \{x_1, x_2, \dots, x_n\} \quad (10)$$

Tabla 10 Datos para HS mejorado con función Rastrigin

Función Rastrigin		IHS	
Prueba	Tiempo	Iteraciones	Exactitud 10 ⁻¹⁰
1	25,355154	628387	9,427871817
2	26,037121	646012	8,949321284
3	26,441581	660270	9,732072925
4	26,098377	655324	8,308163046
5	26,348442	658140	9,279919055

Tabla 11 Datos ABHS con función Rastrigin

Función Rastrigin		ABHS	
Prueba	Tiempo	Iteraciones	Exactitud 10 ⁻¹⁰
1	5,031199	145281	6,191136492
2	5,202211	150852	9,04801567
3	6,68836	188599	9,637002307
4	5,944211	171686	9,095622033

5	4,994423	143705	6,86405599
---	----------	--------	------------

Tabla 12 Datos para HS mejorado con función Rastrigin mejor resultado encontrado

Función Rastrigin		IHS	
Prueba	Tiempo	Iteraciones	Exactitud 10^{-10}
1	2,564015	70571	5,389004798
2	2,565373	70924	8,989857747
3	2,649114	69909	9,964065129
4	2,550446	70487	9,010676649
5	2,519584	69721	8,251834771

Las Tabla 10 y 11 representan los valores hallados utilizando los parámetros generales, en la tabla 12 se observan los mejores resultados encontrados para IHS variando el número de iteraciones hasta encontrar el valor adecuado en este caso 100.000 iteraciones. Valores por encima hacen que el algoritmo converja en más tiempo, iteraciones por debajo no llegan a una respuesta.

3.4 ANALISIS DE RESULTADOS

Al realizar el estudio comparativo entre las diferentes versiones del algoritmo generadas a lo largo de la investigación y la versión original, se observó que el algoritmo IHS presenta mejores resultados que el algoritmo original como era de esperar, pero no mejores que la modificación propuesta ABHS en sus dos versiones, alcanzando una mayor precisión y exactitud, en un menor tiempo de

cómputo, estos datos fueron obtenidos sin adecuar el número de iteraciones al problema en específico.

Se eligió el algoritmo ABHS V2, para realizar las pruebas en los circuitos electrónicos no lineales basándonos en el parámetro VO del sistema de grado seis, que nos indica las veces que el algoritmo entró y logró salir de un estado de saturación.

El algoritmo IHS utilizado en forma general es decir con todos los parámetros constantes incluido el número de iteraciones fijo para las diversas funciones de prueba no presenta mejores resultados que ABHS, pero al variar sus parámetros en especial el NI a un valor adecuado dependiendo el problema presenta resultados similares e incluso mejores que ABHS.

4 EVALUACIÓN DEL ALGORITMO A TRAVES DE EJEMPLOS

Es esta sección se comprobó el algoritmo mediante el uso de funciones estándar, de la cual ya se conocen sus mínimos. Para realizar estas pruebas se usó el equipo de cómputo con las especificaciones que se pueden observar en la Tabla 13

Tabla 13. Características de equipo de cómputo donde se realiza las pruebas.

Equipo de computo	
Procesador	Intel Core i3 @ 2.1GHz
Alimentación	Conectado a red eléctrica
Plan de energía	Alto rendimiento
RAM	3 Gb
SO	Windows 7 64bits

Los parámetros por defecto trabajados en esta sección son los que se observa en la Tabla 14. El parámetro HCMR y el PAR son dos parámetros los cuales varían según el número de iteraciones; El PS, los límites de búsqueda y el HMS son constantes en todas las pruebas y el BW es un parámetro que varía según las condiciones del algoritmo ABHS V2

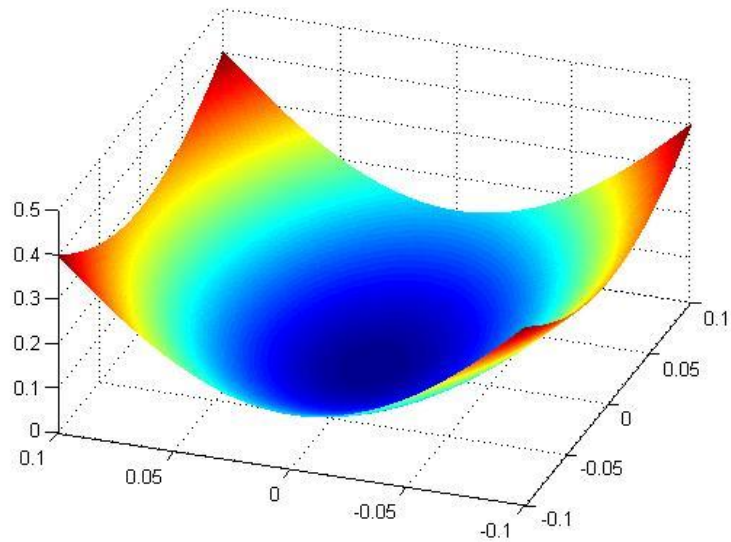
Tabla 14. Parámetros por defecto.

Parámetros por defecto	
HCMR	0,8<HCMR<0,9
BW	Dinámico: 1,6<BW<10 ⁻¹²
	Estático: 0,8
PAR	0,7<PAR<0,9
PS	1000
HMS	20
Limites búsqueda inicialmente	0<Lim<100

4.1 FUNCIÓN DE PRUEBA UNO

Para esta prueba se seleccionó una función polinomio la cual tiene componentes trigonométricas, esto hace que la función sea no lineal, esta función es continua con todo valor real. Para realizar esta prueba se toma los parámetros por defecto la función tiene un mínimo el cual es $f(0,0) = 0$, como se puede observar en la Figura 9.

Figura 9. Gráfico 3D de función de prueba 1



La función a trabajar es la observada en la ecuación (11). En ella se puede ver que hay dos variables de segundo orden.

$$f(x, y) = x^2 + 2y^2 - 0.3 \cos(3\pi x) - 0.4 \cos(4\pi y) + 0.7 \quad (11)$$

Solución del algoritmo:

En la Tabla 15 se observa los datos tabulados al correr el algoritmo, este arroja un tiempo de computo de máximos no mayores a 1 [s] en la mayoría de los casos. Para el desarrollo de esta prueba se toma como parámetro HCMR por defecto, al igual que el espacio de búsqueda inicial. Los resultados observados son bastante aceptables según al grado de exactitud y al tiempo de cómputo, en la Tabla 15 se resaltan las respuesta más rápida y la que arrojó un valor más cercano al teórico de color verde y amarillo respectivamente.

Tabla 15. Datos correspondientes a la función de prueba 1.

Función de prueba uno			ABHS V2			
Prueba	Tiempo de computo (s)	Iteraciones	VO	X	Y	f(x,y)
1	0,699556	29882	23	2,67E-07	4,95E-07	9,23E-12
2	0,978694	41398	33	1,20E-06	2,38E-08	2,06E-11
3	0,753347	32311	23	9,10E-07	-1,16E-06	5,69E-11
4	0,620138	26637	20	9,74E-07	-6,31E-08	1,37E-11
5	0,85889	36619	22	-6,23E-07	1,29E-06	6,18E-11
6	0,591277	24923	15	-9,02E-07	-2,01E-07	1,30E-11
7	0,803582	34324	28	1,91E-06	1,17E-06	9,78E-11
8	0,691143	29655	16	0,11782166	0,09441349	4,98E-11
9	0,774208	33080	32	2,45E-06	6,27E-07	9,95E-11
10	1,008119	42829	36	6,07E-07	-4,62E-07	1,25E-11

4.2 FUNCIÓN DE JONG

Es una función continua, la cual es conocida como primera función de Jong [6], esta presenta un comportamiento suave y simétrico, es usada comúnmente para evaluar la eficiencia de algoritmos genéticos, matemáticamente se representa así:

$$f(X) = \sum_{i=1}^n x_i^2 ; X = \{x_1, x_2, \dots, x_n\} \quad (12)$$

Para esta prueba se va a implementar la función con n=2, como se observa en la ecuación 13 la función de esta forma tiene dos variables.

$$f(x, y) = x^2 + y^2 \quad (13)$$

Los parámetros utilizados en esta prueba son los establecidos por defecto. Se realizará 5 pruebas y se resaltará la de mejor resultado y la de menor tiempo de respuesta.

Solución del algoritmo:

Se han realizado cinco pruebas en los cuales los resultados son coherentes con el valor teórico el cual es $f(0,0)=0$, como se puede observar por inspección en la Figura 10. Los resultados se tabularon en la Tabla 16, en dicha tabla se observa que los valores arrojados por el algoritmo son del orden de 10^{-12} y que el tiempo de respuesta aceptable para nuestro fin. En la Figura 11 se puede observar como el algoritmo se va a aproximando al punto de convergencia.

Tabla 16. Datos correspondientes a la función de Jong.

Función de Jong	ABHS V2				
Prueba	Tiempo (s)	Iteraciones	x	y	f(x,y)
1	0,2015	8898	-6,48E-06	3,10E-06	5,17E-11
2	0,1846	7400	-5,60E-07	-4,21E-06	1,80E-11
3	0,1883	7369	-1,18E-06	2,42E-06	7,25E-12
4	0,1978	7549	3,22E-06	-7,36E-06	6,45E-11
5	0,1696	5332	-2,62E-06	0-1995e-5	1,08E-11

Figura 10. Gráfico 3D de función de Jong

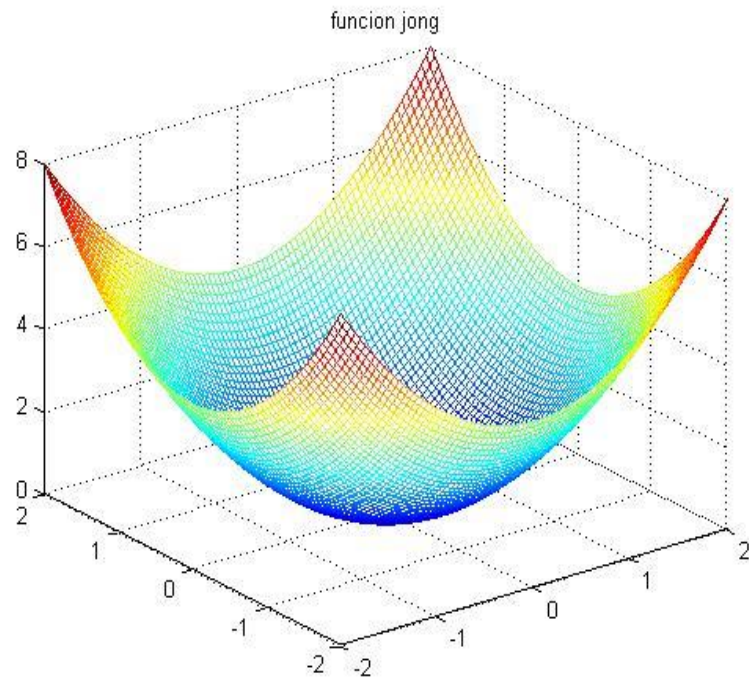
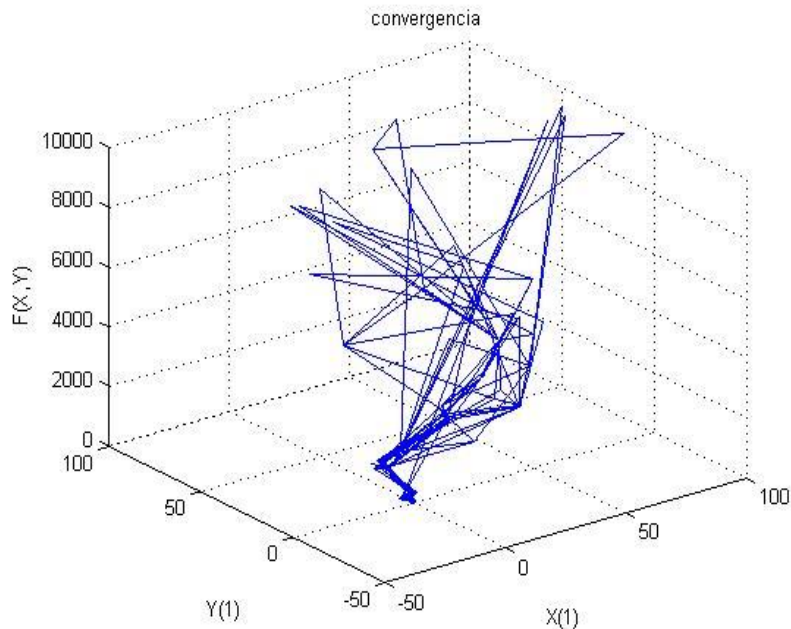


Figura 11. Búsqueda de la convergencia por parte del algoritmo
Búsqueda de la convergencia



4.3 FUNCIÓN DEL VALLE ROSENBROCK

La función de del valle de Rosenbrock es una función no convexa la cual se usa habitualmente para realizar pruebas en problemas de optimización [8] , debido a que es difícil para que los algoritmos converjan en el mínimo, por lo que su mínimo se encuentra ubicado en un valle estrecho.

$$f(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right] \quad (14)$$

Solución de algoritmo

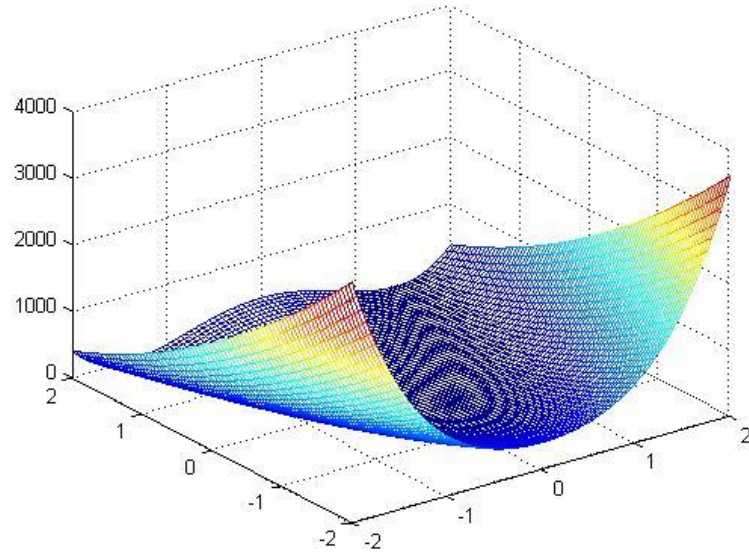
Esta función es un poco más compleja que las anteriores y esto se refleja en el tiempo cómputo que tardó; Las pruebas están cerca de 1 segundo. Aunque los resultados arrojados por el algoritmo no dejaron de ser exactos, siguen siendo del orden de 10^{-12} , y el error de las respuesta teórica puede ser considerado nulo para los resultados arrojado por el algoritmo. El espacio inicial de búsqueda del

algoritmo es el valor por defecto, es decir de 100; el HMCR=0.7 también es el valor por defecto.

Tabla 17. Datos correspondientes a la función de Rosenbrock

Función Rosenbrock	ABHS V2				
Prueba	Tiempo de computo (s)	Iteraciones	x	y	f(x,y)
1	0,8326	38364	1,00E+00	1,00E+00	4,83E-11
2	0,8393	37674	1,00E+00	1,00E+00	7,02E-12
3	1,1034	49711	1,00E+00	1,00E+00	7,50E-11
4	0,8331	37602	1,00E+00	1,00E+00	9,85E-11
5	1,447	65,168	1,00E+00	1,00E+00	6,55E-11
6	2,229	99626	1,00E+00	1,00E+00	9,07E-11
7	1,243	54887	1,00E+00	1,00E+00	3,96E-11
8	0,8867	40126	1,00E+00	1,00E+00	9,83E-11
9	1,3267	60155	1,00E+00	1,00E+00	5,92E-11
10	1,222	54638	1,00E+00	1,00E+00	1,48E-13

Figura 12. Grafica 3D de función de valle de Rosenbrock



4.4 FUNCIÓN DE WOOD

Con ánimo de exigir un poco más el algoritmo se toma la ecuación (15), la cual presenta continuidad en el espacio, esta función polinómica no lineal de cuatro variables.

$$\begin{aligned}
 f(x_1, x_2, x_3, x_4) = & \left(10(x_2 - x_1^2)\right)^2 + (1 - x_1)^2 + \left(\left(90\right)^{\frac{1}{2}}(x_4 - x_3^2)\right)^2 + (1 - x_3)^2 \\
 & + \left(\left(10\right)^{\frac{1}{2}}(x_2 + x_4 - 2)\right)^2 + \left(\left(10\right)^{-\frac{1}{2}}(x_2 - x_4)\right)^2 \quad (15)
 \end{aligned}$$

Solución del algoritmo:

Se observa que los tiempos de cómputos para esta prueba son mucho más altos con respecto a los que se venían observando en las pruebas anteriores y la respuesta pasó de ser del orden de 10^{-12} a 10^{-6} , menos exacta al valor verdadero pero aun así aceptable para nuestro criterio. Observando la Tabla 18 podemos resaltar la prueba 2 por ser la prueba en la cual al algoritmo tardo menos en

converger a una respuesta y la prueba 8 por ser la cual el algoritmo presentó mejor solución. Se pueden encontrar mejores resultados a los obtenidos variando algunos parámetros, pero el fin de las pruebas es utilizar los parámetros iguales en todos los casos.

Tabla 18. Datos correspondientes a la función de Wood.

Función de Wood		ABHS V2					
Prueba	Tiempo de computo (s)	Iteraciones	x1	x2	x3	x4	f(x,y)
1	19,981	625093	0,9986	0,9971	1,0012	1,0026	8,26E-06
2	15,768	492133	0,9985	0,9968	1,0015	1,0031	9,58E-06
3	19,471	604933	1,0012	1,0022	0,9987	0,9976	7,97E-06
4	18,639	581476	1,0009	1,0017	0,9992	0,9983	2,72E-06
5	21,171	645609	1,0014	1,0028	0,9985	0,9971	7,79E-06
6	17,7	551841	0,9985	0,9969	1,0014	1,0028	8,65E-06
7	19,536	592062	0,9984	0,9968	1,0015	1,0032	9,93E-06
8	17,958	559988	0,9995	0,99	1,0006	1,0012	2,19E-06
9	19,659	616513	1,0013	1,0028	0,9985	0,9969	9,83e3-6
10	17,778	557746	0,9985	0,9969	1,0015	1,003	9,15E-06

4.5 FUNCIÓN DE BOOTH

Esta función no presenta discontinuidad, tiene dos variables las cuales son cuadráticas, lo que la hace no lineal. Su valor mínimo es $f(1,3)=0$ teóricamente. Para generar los resultados de esta simulación los parámetros utilizados fueron los estándares [10].

$$f(x_1, x_2) = (x_1 + 2 * x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \quad (16)$$

Solución del algoritmo:

En la Tabla 19 se muestra la tabulación de valores arrojados por el algoritmo, así como el tiempo que demoró en mostrar las respuestas y el número de iteraciones necesaria para ello. De la Tabla 19 cabe resaltar la prueba 2 y la prueba 9 debido que son las que presentaron mejor tiempo de respuesta y las respuestas más cercanas a la teórica, respectivamente. En la Figura 14 se observa como el algoritmo llega a su punto de convergencia.

Tabla 19. Datos correspondientes a la función de Booth.

Función de Booth	$f(x,y)=(x1+2*x2-7)^2+(2x1+x2-5)^2$				
Prueba	Tiempo de computo (s)	Iteraciones	x1	x2	f(x,y)
1	0,5827	24381	1	3	5,63E-11
2	0,2424	10714	1	3	3,71E-11
3	0,8401	34471	1	3	6,61E-11
4	0,9479	38611	1	3	6,69E-11
5	0,7778	31947	1	3	9,94E-11
6	0,7293	30030	1	3	6,99E-11
7	0,4925	20730	1	3	7,25E-11
8	0,4147	17801	1	3	5,82E-11
9	0,3009	13262	1	3	1,50E-11
10	0,2633	11627	1	3	3,84E-11

Figura 13. Gráfica 3D con de la función de Booth.

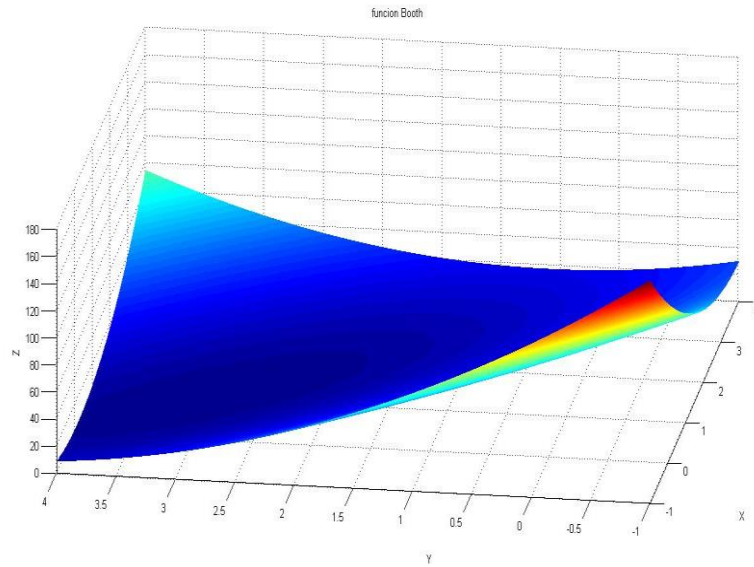
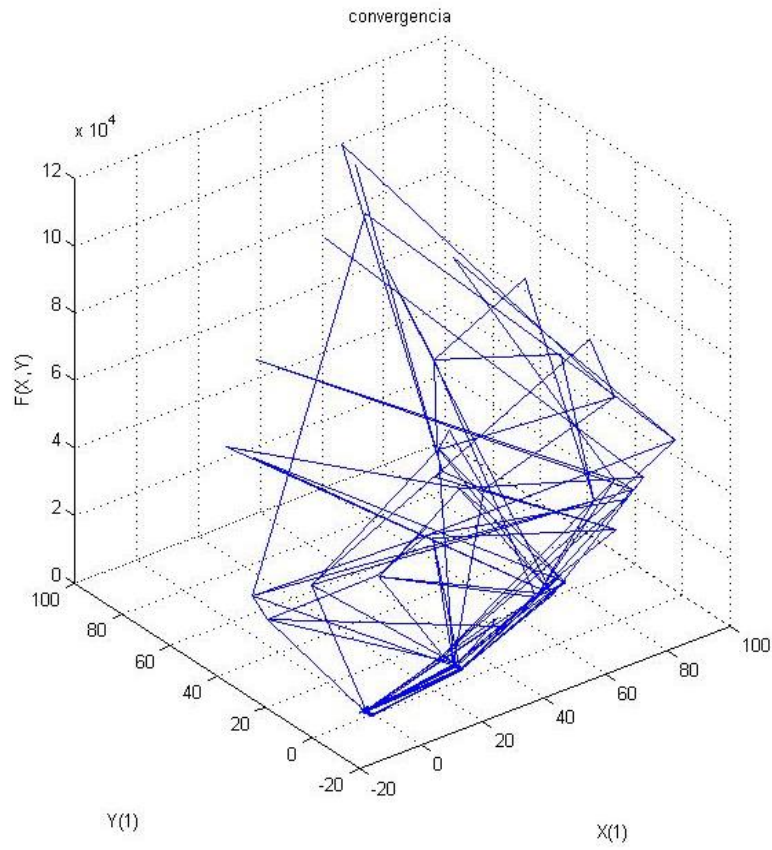


Figura 14. Búsqueda del punto de convergencia por parte del algoritmo.



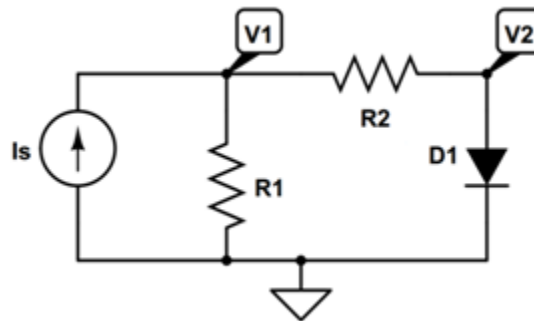
5 EXPERIMENTOS CON CIRCUITOS CON COMPONENTES NO LINEALES

Los resultados reportados en esta sección se obtuvieron utilizando el mismo equipo de cómputo y las especificadores que la sección anterior. Para las pruebas en ocasiones fue necesario variar los valores por defecto para que el algoritmo presentara mejor respuesta.

5.1 CIRCUITO DE PRUEBA UNO

La primera prueba resuelve el sistema de ecuaciones que describe el circuito observado en la Figura 15, (es tomado de [7]), el cual tiene un componente no lineal. Durante los ensayos con este circuito se variaron los valores de la fuente de corriente y de V_t , el cual es un parámetro que ayuda a describir el comportamiento no lineal del diodo y se observa cómo influye el cambio de esos valores en los resultados.

Figura 15. Circuito de prueba 1.



La ecuación (17) y ecuación (18) describen el comportamiento eléctrico del sistema, se utilizó el modelo del diodo Shockley (en honor a William Bradford Shockley). Se buscó hallar la tensión marcada en los nodos 1 y 2.

$$0 = \left(\frac{1}{R_1} + \frac{1}{R_2} \right) * V_1 - \frac{1}{R_2} * V_2 - I_s \quad (17)$$

$$0 = -\frac{1}{R_2} * V_1 + \left(\frac{1}{R_2}\right) * V_2 + I_{sat} * \left(e^{\frac{V_2}{V_t}} - 1\right) \quad (18)$$

5.1.1 Ensayo 1:

En este primer ensayo se usó como valores de los componentes del circuito los observados en la Tabla 20.

Tabla 20. Valores de los componentes del circuito para el ensayo 1

Parámetros Circuito de prueba 1	
R1(ohm)	100
R2(ohm)	10000
Is(A)	0,1
Vt(V)	0.025875
Isat(A)	10 ⁻¹⁵

Solución del algoritmo:

Como se puede observar en la Tabla 21, los resultados entregados por el algoritmo son bastante buenos en términos de tiempo de cómputo, además también se observa que el aumento en la precisión aumenta el tiempo de cómputo y el número de iteraciones como es de esperarse.

Tabla 21. Datos para el circuito de prueba 1.

Circuito de Prueba 1			ABHS V2			
precisión e-016						
Prueba	Tiempo (s)	Iteraciones	PO	V1	V2	F(V1,V2)
1	0,556227	26407	17	9,90804556	0,71278177	3,31647963
2	0,611651	29201	18	9,90804831	0,71278171	1,20775458
3	0,572932	27073	22	9,90804561	0,71278129	6,96957795
4	0,589759	28084	20	9,90804719	0,71278229	2,49865594
5	0,488595	23389	12	9,90804703	0,71278273	9,79993777
6	0,502714	23958	12	9,90804989	0,71278162	1,65599978
precisión e-13						
7	0,277185	12999	7	9,90808171	0,71278266	1,21E-13
8	0,22109	10138	4	9,90807853	0,71275867	7,88E-13
9	0,166181	8068	5	9,90804634	0,7127923	1,39E-13
10	0,160809	7881	5	9,90797147	0,71276634	8,82E-13
12	0,179043	8685	3	9,90807445	0,71277725	1,03E-13

En este mismo circuito los datos fueron obtenidos de [9] y [13] aunque existen diferencias en los equipos de cómputo utilizados. No fue posible obtener los códigos para hacer las pruebas en un mismo equipo.

El algoritmo que presento mejores resultados en cuanto a tiempo de computo fue UPSO implementado en un equipo con procesador Intel i7, 6 GB RAM, con un resultado muy similar se encuentra el algoritmo ABHS V2 objeto de estudio en este trabajo de grado implementado en un equipo con procesador Intel i3, 3 GB RAM.

Tabla 22. Tabla comparativa para los siguientes métodos

	ABHS V2	CFOFORM	PSO	UPSO	NRM	PSPICE
V1	9,90804703	99.135	10	9,908	9,908	9,908
V2	0,71278273	0.70610	0	0,7128	0,713	0,7125
PRECISIÓN	9,73E-16	-4,72E-05	2,00E-06	7,61E-22	0	-----
TIEMPO	0,488595	68,32	0,28	0,06	0,16	0
ITERACIONES	23389	-----	433	98	369	-----

Respecto al número de iteraciones utilizadas se evidencia que ABHS V2 tiene una cantidad de iteraciones grande con respecto a los demás métodos esto quiere decir puede realizar más iteraciones en el mismo rango de tiempo que los otros algoritmos mostrando que el costo computacional es menor, se pueden obtener mejores resultados para HS variando algunos parámetros.

5.1.2 Ensayo 2:

El segundo ensayo se realizó con los parámetros de la Tabla 23 estos se variaron con respecto a los anteriores valores de V_t y corriente de la fuente, *se aclara que algunos valores del voltaje térmico V_t utilizados no son plausibles, son usados para demostrar el efecto de la modificación del componente no lineal del circuito.*

Tabla 23. Valores de los componentes del circuito de prueba 1 para el ensayo 2

Parámetros Circuito de prueba 1	
R1(ohm)	100
R2(ohm)	10000
Is(A)	0,5
Vt(V)	0.5
Isat(A)	10^{-15}

Solución de algoritmo:

Comparando los tiempos de cómputo de esta tabla y la anterior se observa que no existe un aumento significativo al variar los parámetros V_t e I_s . Los valores de las incógnitas hallados siguen siendo igual de precisos que en el ensayo anterior.

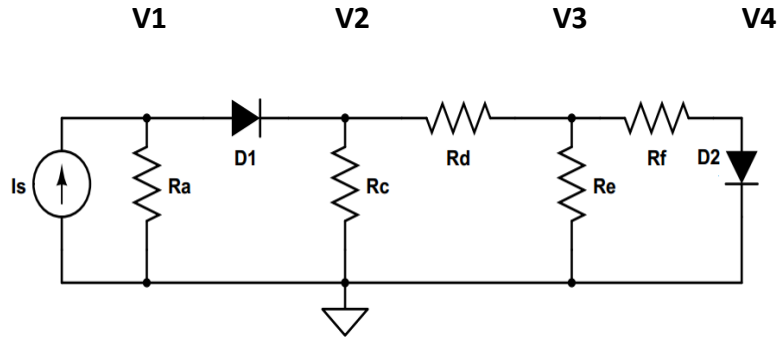
Tabla 24. Datos para el circuito de prueba 1 variando V_t e I_s

Prueba	Tiempo de computo (s)	Iteraciones	PO	V1	V2	precisión 10^{-16}
1	0,485516	22854	15	49,64796882	14,44478289	6,230255092
2	0,251874	12162	5	49,64796841	14,44478456	1,522283377
3	0,357559	16748	11	49,64796842	14,44478922	4,54944536
4	0,510481	23744	18	49,64796687	14,4447868	1,932945461
5	0,337526	16006	12	49,64796537	14,44478534	8,3679762
6	0,373947	17701	9	49,64796798	14,44478694	2,891349653

5.2 CIRCUITO DE PRUEBA DOS

Para el circuito mostrado en la Figura 16 se realizaron dos pruebas, una variando el HMS y otra variando el parámetro del diodo V_t .

Figura 16. Circuito de prueba 2



Las ecuaciones que describen este sistema están numeradas de la ecuación (19) a la ecuación (22), como se puede ver estas ecuaciones tienen componentes no lineales.

$$f_1 = \frac{1}{R_a} * v_1 + I_{sat} \left(e^{\frac{q * V_{V1} - V_2}{n * K_B * T}} - 1 \right) - I_S = 0 \quad (19)$$

$$f_2 = -I_{sat} \left(e^{\frac{q * V_B}{n * K_B * T}} - 1 \right) + \left(\frac{1}{R_c} + \frac{1}{R_d} \right) * v_2 + \left(-\frac{1}{R_d} \right) * v_3 = 0 \quad (20)$$

$$f_3 = \left(-\frac{1}{R_d}\right) * v_2 + \left(\frac{1}{R_d} + \frac{1}{R_e} + \frac{1}{R_f}\right) * v_3 + \left(-\frac{1}{R_f}\right) * v_4 = 0 \quad (21)$$

$$f_4 = \left(-\frac{1}{R_f}\right) * v_3 + \left(\frac{1}{R_f}\right) * v_4 + I_{sat} \left(e^{\frac{q * V_4}{n * K_B * T}} - 1 \right) = 0 \quad (22)$$

5.2.1 Ensayo 1: Se realizó la prueba modificando el parámetro HMS para verificar como responde según el tiempo de cómputo. se observa que la respuesta está en el orden de 10^{-15} .

Tabla 25. Valores de los componentes del circuito de prueba 2 para el ensayo 1

Parámetros	
Vt	0,025
ra, rc, rd, re, rf (Ω)	1
Isat(A)	10^{-6}
I1(A)	0.05

Solución del algoritmo:

Al variar el parámetro HMS se observó que la respuesta en este caso converge más rápido, aunque no es recomendable asignarle valores pequeños.

Si HMS fuese muy pequeño el algoritmo podría saturarse o podría caer en un minino local.

Tabla 26. Datos para el circuito de prueba 2 variando el valor del parámetro HMS

Circuito de Prueba 2						
Precisión 10^{-15}						
HMS=20						
Prueba	Tiempo de computo (s)	Iteraciones	V1	V2	V3	V4
1	14,0837	377514	4,99936E-02	4,27332E-06	2,11005E-06	2,11176E-06
2	13,2358	396667	4,99936E-02	4,24200E-06	2,10358E-06	2,02941E-06
3	14,5113	424375	4,99937E-02	4,27150E-06	2,14512E-06	2,12728E-06
4	14,1835	415081	4,99937E-02	4,25772E-06	2,16324E-06	2,17053E-06
5	15,1728	450914	4,99936E-02	4,24862E-06	2,13959E-06	2,10744E-06
Promedio	14,2374	412910.2	4,99936E-02	4,25863E-06	2,13232E-06	2,10928E-06
HMS=10						
1	8,08660	242920	4,99936E-02	4,27775E-06	2,13081E-06	2,16158E-06
2	9,23989	280197	4,99937E-02	4,23254E-06	2,13360E-06	2,14916E-06
3	7,94315	237124	4,99936E-02	4,22250E-06	2,17556E-06	2,17277E-06
4	7,48138	227121	4,99936E-02	4,21652E-06	2,09844E-06	2,09122E-06
5	7,48138	259612	4,99936E-02	4,26009E-06	2,11722E-06	2,07794E-06
Promedio	8,04648	249394.8	4,99936E-02	4,24188E-06	2,13113E-06	2,13053E-06

5.2.2 Ensayo 2:

Se varió el valor de la constante V_t y se toman solo las iteraciones y el tiempo de cómputo debido a que las respuestas del sistema son lo suficientemente exactas y estables.

Tabla 27. Valores de los componentes del circuito de prueba 2 para el ensayo 2.

Parámetros	
ra, rc, rd, re, rf (Ω)	1
Isat(A)	10^{-6}
I1(A)	0.05

Como se puede ver si se varía el parámetro V_t el cual hace parte de la componente no lineal, aumenta el tiempo de convergencia y el número de iteraciones, pero no de manera brusca.

Tabla 28. Datos para el circuito de prueba 2 variando $q/(n \cdot k_b \cdot t)$

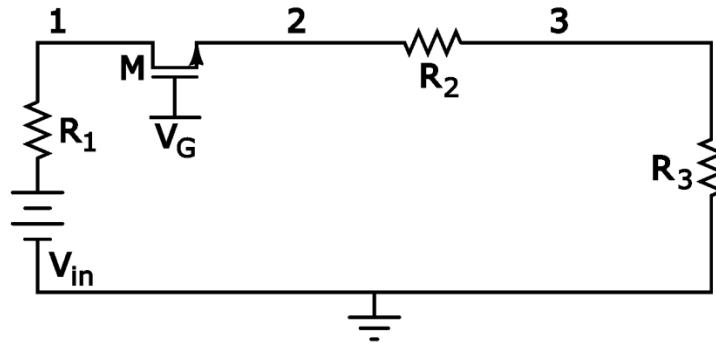
Parámetros 1				
HMS=10				
Vt=0.033			Vt=0.066	
Prueba	Tiempo de computo (s)	Iteraciones	Tiempo de computo (s)	Iteraciones
1	6,647876	200686	9,203647	217715
2	7,584575	222130	10,338699	303170
3	7,365577	221233	7,363182	223143
4	7,690246	230839	7,432563	225510
5	7,239098	217715	8,228142	248735
Promedio	7,3054744	218520	8,5132466	243654

5.3 CIRCUITO DE PRUEBA TRES

Se resolvió el circuito DC-DC de la

Figura 17, este circuito es utilizado para obtener en su salida una tensión de DC más reducida que la de su entrada, la corriente que circula por el circuito es controlada por la tensión aplicada entre V_G y V_2 , siendo la tensión de salida V_3 directamente proporcional tanto a la tensión V_{GS} del transistor como a la resistencia R_3 .

Figura 17. Circuito de prueba conversor DC-DC



Los parámetros utilizados para evaluar son los de la Tabla 29

Tabla 29. Valores de los componentes del circuito conversor DC-DC

Fuente: elaboración propia

V_{in}	5[V]
V_G	5[V]
R_1	100[Ω]
R_2	10[Ω]
R_3	500[Ω]
$\left(\frac{k_n}{w * l}\right)$	$450 \times 10^{-6} [A/V^2]$

El sistema de ecuaciones utilizado para describir este sistema está dado por las ecuaciones (23) a la (26) donde I_M representa la corriente que fluye entre drenador y surtidor del transistor MOSFET.

$$I_M = 0.5 * \left(\frac{k_n}{W * l} \right) * (V_{GS} - V_T)^2 \quad (23)$$

$$f1 = \frac{V_{in}}{R_1} - \frac{V(1)}{R_1} - I_M = 0 \quad (24)$$

$$f2 = I_M - \frac{V(2)}{R_2} + \frac{V(3)}{R_2} = 0 \quad (25)$$

$$f3 = \left(\frac{1}{R_2} + \frac{1}{R_3} \right) * V(3) - \frac{V(2)}{R_2} = 0 \quad (26)$$

Solución del algoritmo:

Se observó tiempos de cómputos polarizados debidos a la naturaleza del método, la precisión obtenida en todas las pruebas fue en el orden 10^{-13} , se observan dos posibles soluciones esto es debido a la ecuación cuadrática (23), es posible encontrar dos tensiones V_{GS} para un mismo valor de corriente del transistor. La región que se presume está el transistor trabajando es en saturación para estar en esta región se deben satisfacer dos condiciones.

$$V_{GS} > V_T \quad (27)$$

$$V_{DS} > V_{GS} - V_T \quad (28)$$

Donde V_{GS} es igual a

$$V_{GS} = V_G - V_2 \quad (29)$$

Las tensiones resaltadas en verde satisfacen esta condición presentando una tensión promedio $V_{GS} = 3,8568161988442$. Como resultado de esto ambas respuesta son válidas matemáticamente pero solo la resaltada en verde tiene fundamentación física.

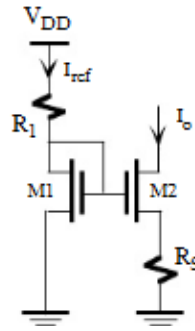
Tabla 30 Datos tomados por parte del algoritmo ABHS V2 para el circuito de prueba 3

Precisión 10^{-13}					
Prueba	Tiempo de computo (s)	Iteraciones	V1	V2	V3
1	16.737053	165627	4.7758253681976	1,1431838011558	1.1207624065775
2	19.048947	185482	4.7758458673265	1.1435221495196	1.1210974069106
3	4.890430	144409	4.7758672735953	1.1437902283104	1.1213699517506
4	17.946789	165252	4.7758670620975	1.1438035857462	1.1213830536284
5	13.546793	123548	1.8289611566820	16.1716124901173	15.8545146864181
6	19.982624	185330	1.8289632464956	16.1715510109537	15.8544554776147
7	24.243006	232326	1.8294032166230	16.1707948500465	15.8537268184704
8	17.084787	161398	1.8289662964489	16.1716123792066	15.8545147654937

5.4 CIRCUITO ESPEJO DE CORRIENTE WIDLAR

Este circuito, Figura 18 es un espejo de corriente Widlar sencillo, tiene dos transistores M1 y M2. Este circuito no es simétrico debido a que la diferencia de tensión de puerta surtidor es diferente para cada transistor.

Figura 18. Espejo de corriente Widlar



Los parámetros del sistema son los que se observan en la Tabla 31

Tabla 31. Parámetros constantes para el espejo de corriente

V_t	0.7 [V]
$K_n * \left(\frac{W}{L}\right)$	19.4×10^{-3} [A/V ²]
V_{GS1}	0.8015[V]

El sistema de ecuaciones (30) y (31) que rigen el sistema está dado por

$$0 = -I_0 + 0.5 * \left(\frac{k_n}{W * l}\right) * (V_{GS2} - V_T)^2 \quad (30)$$

$$0 = -V_{GS1} + V_{GS2} + I_0 * R_S \quad (31)$$

Se tomó como criterio de parada 1000 iteraciones en saturación o si la respuesta estaba en el orden de 10^{-11} . Para los siguientes ensayos se varió la resistencia y se observó cómo afectó el tiempo de respuesta y la precisión.

Solución del algoritmo:

Ensayo 1:

Para este ensayo el promedio de tiempo de cómputo fue mayor que para otros circuitos, por otra parte los valores hallados mostraron una buena exactitud.

Tabla 32. Datos para prueba 1 del espejo de corriente Widlar

Fuente de corriente Widlar $R_s=1000 \Omega$					
Prueba	Tiempo de computo (s)	Iteraciones	V1	V2	Solución
1	366,86339	13919144	0,761241162	4,02651E-05	5,3752E-11
2	104,374589	3632525	0,76177532	3,97158E-05	8,7007E-11
3	832,229586	28071010	0,76207666	3,94143E-05	8,5176E-11
4	650,697601	23010258	0,763425318	3,80774E-05	8,127E-12
5	274,507322	9783209	0,75887491	4,26288E-05	9,4908E-11
Promedio	445,7344976	15683229	0,761478674	4,00203E-05	6,5794E-11

Ensayo 2:

Se observó que al reducir la resistencia de 1000Ω a 100Ω los tiempos de cómputo se reducen drásticamente.

Tabla 33. Datos para ensayo 2 del espejo de corriente Widlar

fuente de corriente Widlar $R_s=100 \Omega$					
Prueba	Tiempo de computo (s)	Iteraciones	V1	V2	solución
1	5,17	1,77E+05	0,793541981142632	7,96E-05	3,17E-11
2	5,64	1,93E+05	0,793731439291911	7,77E-05	5,77E-11
3	23,5	8,23E+05	0,792286941956214	9,21E-05	9,81E-11
4	22,1	7,73E+05	0,792259595416244	9,24E-05	9,93E-11
5	13,1	4,54E+05	0,79386738020569	7,63E-05	8,43E-11
Promedio	13,9	4,84E+05	7,93E-01	8,36E-05	7,42E-11

Ensayo 3:

Por último se realizó una prueba en la cual se aumentó el grado de exactitud es decir en el orden de 10^{-16} , se evaluó con el mismo valor de resistencia que el ensayo 2, se observa que en promedio el tiempo de cómputo aumentó 5 veces al hallar respuestas más exactas y precisas.

Tabla 34. Datos para ensayo 3 del espejo de corriente Widlar

Fuente de corriente Widlar R=100 exactitud 10-16					
Prueba	Tiempo de computo (s)	Iteraciones	V1	V2	solución
1	50,609513	1.926.002,	7,93E-01	8,41E-05	1,00E-15
2	89,038764	2.952.852,	7,93E-01	8,41E-05	1,00E-15
3	52,352246	1.649.103,	7,93E-01	8,41E-05	4,4x-16
4	65,638383	2.115.995,	7,93E-01	8,40E-05	9.9314x-16
5	46,578305	1.504.519,	7,93E-01	8,41E-05	2.36745x-16
Promedio	60,8434422	2.029.694,2	7,93E-01	8,41E-05	1,00E-15

6 CONCLUSIONES

1. Se cumplieron todos los objetivos planteados en el plan satisfactoriamente
2. Se implementó el algoritmo metaheurístico llamado Harmony Search siendo capaz de encontrar el mínimo de una función objetivo, y de solucionar un sistema de ecuaciones no lineales. Se comprobó en un sistema de diez y hasta 100 variables encontrando la respuesta con una exactitud de 10^{-10} (Ver en capítulo 6 y anexo B).
3. Al aplicar el algoritmo ABHS a un sistema de ecuaciones que describen el modelo matemático de un circuito electrónico de componentes no lineales en DC, no se encontraron inconvenientes para resolver el circuito, salvo en el aumento del tiempo de cómputo al variar ciertos parámetros, como el valor de algunas resistencias (Ver en capítulo VII).
4. Este método resultó ser una alternativa viable en la solución de circuitos electrónicos no lineales en DC, en cuanto se garantice el teorema de las raíces reales, haciendo posible encontrar la solución sin tener que estimar un valor cercano a la respuesta (Ver en capítulo V).

5. La variación propuesta ABHS del algoritmo original HS, presenta mejores resultados tanto en tiempo de cómputo como en precisión frente al algoritmo original como en su modificación más popular IHS.
6. Al contrastarse en un mismo circuito HS, CFO, PSO y NRM se pudo observar que el algoritmo HS presentó mejores resultados frente al algoritmo CFO, los resultados son similares frente a PSO, NRM respecto al tiempo de cómputo y exactitud (Ver en capítulo VII).

7 RECOMENDACIONES

Se recomienda en investigaciones futuras, analizar el comportamiento y la viabilidad de un algoritmo basado en Harmony Search con parámetros HMCR y PAR ajustables.

8 REFERENCIAS BIBLIOGRÁFICAS

- [1] G. V. Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search," *Simulation*, vol. 76, no. 2, pp. 60–68, Feb. 2001.
- [2] I. Amaya, J. Cruz, and R. Correa, "Real Roots of Nonlinear Systems of Equations Through a Metaheuristic Algorithm," *Revista Dyna*, vol. 78, no. 170, pp. 15–23, 2011
- [3] I. Kougiyas and N. Theodosiou, "A New Music-Inspired Harmony Based Optimization Algorithm. Theory And Applications." Division of Hydraulics and Environmental Engineering Department of civil Engineering. pp. 1–8, 2010.
- [4] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, May 2007.
- [5] X. Yang, "Harmony search as a metaheuristic algorithm," *Music-inspired harmony search algorithm*, Theory and Applications (Editor Z,W. Geem) Studies in Computational Intelligence, Springer Berlin, vol 191, pp 1-14 (2009).
- [6] (2011). [ONLINE] Available at: <http://commonhacks.blogspot.com/2012/05/suite-de-funciones-de-de-jong-en-1975.html>. [Last Accessed Enero-10- 2012].
- [7] I. Amaya, J. Cruz, and R. Correa, "Solution Of The Mathematical Model Of A Nonlinear Direct Current Circuit Using Particle Swarm Optimization," *Dyna*, 2012.
- [8] Felipe Andrés Besoain Pino (2008). Optimización de funciones mediante método de Rosenbrock. [ONLINE] Available at: <http://fbesoain.wordpress.com/2008/11/20/optimizacion-de-funciones-mediante-metodo-de-rosenbrock/>. [Ultimo acceso Enero-10-2012].
- [9] O. Roa and F. Ramírez, "Solución De Las Ecuaciones Que Describen El Modelo Matemático De Un Circuito Electrónico Compuesto De Elementos No Lineales Mediante El Optimizador De Fuerza Central," Trabajo de grado Ingeniería Electronica UIS pp. 1–69, 2012.
- [10] J. Fourie, S. Mills, and R. Green, "Visual Tracking Using Harmony Search," *Recent Advances In Harmony Search*, Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand, pp 1-6 2010.

- [11] Ghosh, S. Kundu, D. Suresh, K. Das, S. Abraham, "A Design of optimal digital IIR filters by using a Bandwidth Adaptive Harmony Search algorithm", *Nature & Biologically Inspired Computing*, 2009. NaBIC 2009. World Congress.
- [12] Milan Cisty, "Hybrid Model for Water Distribution Design," *Evolutionary Computation (CEC)*, 2010 IEEE Congress on.
- [13] Jinxue Sui, Li Yang, Hui Fan, Zhen Hua, Jinxue Sui, Li Yang, Mine. "Airflow Optimizing Control Based on Harmony Annealing Search". 2010 International Conference of Information. *Journal of Computers*, Vol 6,N 6, june 2011
- [14] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36–38, pp. 3902–3933, Sep. 2005.

9 ANEXOS

9.1 ANEXO A

Tablas comparativas circuito de prueba 1 tomado de [2]

Circuito de prueba uno CFO1

Tabla 35. Circuito de prueba uno CFO1

Prueba	Tiempo de computo (s)	BF	Salida	Posición		Np	300
				X	Y	Nt	10000
1	1949.498766	-9.2216E-07	C	9.8934	0.3563	S	10000
2	2274.904957	-9.0223E-07	C	9.9040	0.4059	C	1.00E-20
3	2248.424497	-3.5623E-07	C	9.8490	0.7126	Mapeo	2000000
Promedio	2157.609407	Promedio		9.8821	0.4916		
		Distancia BF		0.0590			
		Distancia Promedio		0.2227			

Circuito de prueba uno CFO2

Tabla 36. Circuito de prueba uno CFO2

Prueba	Tiempo de computo (s)	BF	Salida	Posición	
				X	Y
1	389.888657	-2.8972E-22	C	9.9080	0.7128
2	677.528430	-1.0863E-14	S	9.9080	0.7128
3	406.535890	-4.1489E-22	C	9.9080	0.7128
Promedio	491.317659	Promedio		9.9080	0.7128
		Distancia BF		0.0000	
		Distancia Promedio		0.0000	

Circuito de prueba uno CFOFORM

Tabla 37. Circuito de prueba uno CFOFORM

Prueba	Tiempo de computo (s)	BF	Salida	Posición		
				X	Y	
1	68.328.700	-4,72E-04	S	99.135	0.70610	
2	72.092.300	-4,72E+00	S	99.135	0.71610	
3	71.313.800	-4,72E-04	S	99.135	0.70610	
Promedio	70.578.267	Promedio		99.135	0.70943	
				Distancia BF		0.0087
				Distancia Promedio		0.0064

9.2 ANEXO B

Se implementó el algoritmo para solucionar la función Griewank de orden 100 ecuación 26

$$f(x) = \sum_{j=1}^{100} \frac{x_j^2}{4000} - \prod_{i=1}^{100} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad [26]$$

Los resultados se pueden observar en la tabla 37, se utilizó una HMS de 20.

Función Griewank grado 100		ABHS	
Prueba	Tiempo	Iteraciones	Exactitud
1	23,508467	88188	9,8983E-10
2	21,841839	86009	9,64239E-10
3	102,817733	384515	0,034384062
4	21,775784	82914	9,88029E-10
5	22,01022	81461	9,52576E-10
6	26,719696	99248	9,48716E-10
7	22,789469	83708	9,95234E-10
8	22,687504	84968	9,92024E-10

Tabla 38 resultados función Griewank orden 100 ABHS V2

Se utilizaron los siguientes parámetros

HMCRmin=0.8; HMCRmax=0.9; PARmin=0.7; PARmax=0.9; BWmax=0.8; BWmin=10⁻⁸;

HMS=1000;

Se observa que de la pruebas solo una no pudo llegar a una respuesta con alto grado de exactitud.