

**ANÁLISIS DE LA EFICIENCIA DEL *ALGORITMO HARMONY SEARCH* EN LA  
SOLUCIÓN DE UN CIRCUITO ELECTRÓNICO NO LINEAL**

**AUTORES:**

**SAUL ENRIQUE GONZALEZ AGUINAGA**

**OSCAR IVAN TRASLADINO SALAZAR**

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

**2013**

**ANÁLISIS DE LA EFICIENCIA DEL *ALGORITMO HARMONY SEARCH* EN LA  
SOLUCIÓN DE UN CIRCUITO ELECTRÓNICO NO LINEAL**

**SAUL ENRIQUE GONZALEZ AGUINAGA**

**OSCAR IVAN TRASLADINO SALAZAR**

**Trabajo de Grado para optar al título de  
INGENIERO ELECTRÓNICO**

**Director**

**PhD. CARLOS RODRIGO CORREA CELY**

**Co-director**

**PhD. (c) IVÁN MAURICIO AMAYA CONTRERAS**

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

**2013**

## **AGRADECIMIENTOS**

Les damos gracias a nuestros padres por apoyarnos en todo momento, por los valores que nos inculcaron, y por habernos dado la oportunidad de tener una excelente educación.

A dos grandes seres humanos *Ph.D. Carlos Rodrigo Correa* y *Ph.D.(c) Iván Mauricio Amaya*, por habernos brindado la oportunidad de desarrollar nuestro trabajo de grado y por llevar a cabo la dirección de esta tesis.

## DEDICATORIAS

A mis padres, *Antonio Trasladino Ramírez* y *Beatriz Salazar Vanegas*, por darme la vida, y por darme un excelente ejemplo de vida a seguir.

A mi madre, *Lida T. Aguinaga Rivera* por haberme apoyado y guiado durante este tiempo de vida, además de ser siempre mi apoyo incondicional, e igualmente a mi padre *Saul Gonzalez Castillo* por su apoyo y colaboración.

## TABLA DE CONTENIDO

INTRODUCCIÓN .....	18
1. EL TEMA QUE SE ABORDA .....	20
1.1 Objetivo General <sup>1</sup> .....	20
1.2 Objetivos Específicos <sup>1</sup> .....	20
2. FUNDAMENTACIÓN TEÓRICA.....	21
2.1 Estrategia de solución.....	21
2.2 Método de optimización de Búsqueda Armónica (Harmony Search).....	22
2.3 Descripción del algoritmo búsqueda armónica (Harmony Search): .....	23
2.4 Modelo matemático de los dispositivos no-lineales utilizados .....	29
2.4.1 Diodo .....	29
2.4.2 Transistor MOSFET.....	29
3. EXPERIMENTACIÓN.....	31
3.1 Convenciones para parámetros y resultados.....	31
3.2 Funciones de prueba .....	32
3.2.1 Función Himmelblau .....	33
3.2.2 Función Ackley .....	36
3.2.3 Función Rosenbrock.....	41
3.3 SOLUCIÓN DE CIRCUITOS ELECTRONICOS NO LINEALES DE C.C.....	49
3.3.1 Circuito de corriente directa con un diodo .....	49
3.3.2 Circuito de corriente directa con un MOSFET .....	52
3.3.3 Circuito convertidor de corriente a voltaje de corriente directa con cinco MOSFETS .....	55

3.4	Circuito de corriente directa con lazos adicionales en la carga. ....	60
3.4.1	Circuito de corriente directa sin lazos adicionales .....	61
3.4.2	Circuito de corriente directa con un lazo adicional .....	63
3.4.3	Circuito de corriente directa con ocho lazos adicionales .....	64
3.4.4	Circuito de corriente directa con trece lazos adicionales .....	66
4.	OBSERVACIONES Y CONCLUSIONES .....	70
5.	RECOMENDACIONES .....	72
	BIBLIOGRAFIA.....	73
	ANEXOS.....	75
	ANEXO A. Tutorial para la ejecución del algoritmo de búsqueda armónica .....	75
	ANEXO B. Función Rosenbrock para diez, veinte y treinta variables .....	78

## LISTA DE FIGURAS

Figura 1. Analogía entre la improvisación musical y la optimización en ingeniería. ....	22
Figura 2. Inicialización de la memoria armónica. ....	26
Figura 3. Memoria armónica actualizada. ....	28
Figura 4. Función Himmelblau 2D. ....	33
Figura 5. Solución de la función de Himmelblau 2D. ....	34
Figura 6. Variación del parámetro HMCR en la función Himmelblau. ....	35
Figura 7. Variación del parámetro PAR en la función Himmelblau. ....	35
Figura 8. Función Ackley 2D. ....	37
Figura 9. Solución de la función de Ackley 2D. ....	38
Figura 10. Variación del parámetro HMCR en la función Ackley. ....	39
Figura 11. Variación del parámetro PAR en la función Ackley. ....	39
Figura 12. Función Rosenbrock 2D. ....	41
Figura 13. Solución de la función de Rosenbrock 2D. ....	42
Figura 14. Variación del parámetro HMCR en la función Rosenbrock 2D. ....	43
Figura 15. Variación del parámetro PAR en la función Rosenbrock 2D. ....	43
Figura 16. Variación del parámetro HMCR en la función Rosenbrock 3D. ....	45
Figura 17. Variación del parámetro PAR en la función Rosenbrock 3D. ....	45
Figura 18. Variación del parámetro HMCR en la función Rosenbrock 4D. ....	47
Figura 19. Variación del parámetro PAR en la función Rosenbrock 4D. ....	47
Figura 20. Circuito de c.c con un diodo. ....	49
Figura 21. Circuito de amplificador Single con un MOSFET. ....	52
Figura 22. Circuito convertidor de corriente a voltaje de c.c con 5 MOSFET. ....	55
Figura 23. Circuito de c.c con lazos adicionales en la carga. ....	60
Figura 24. Script OF.m. ....	75
Figura 25. Limite superior e inferior del script BA.m. ....	76
Figura 26. Ejecución del archivo BA.m. ....	76

Figura 27. Resultado final de la ejecución del algoritmo de búsqueda armónica...77

## LISTA DE TABLAS

Tabla 1. Parámetros de inicialización de a memoria armónica.....	25
Tabla 2. Limites inferior y superior de cada variable de decisión.....	25
Tabla 3. Valores aleatorios de la primera fila de la memoria armónica.....	25
Tabla 4. Valores generados aleatoriamente y parámetros anteriores.....	27
Tabla 5. Parámetros del diodo.....	29
Tabla 6. Ecuaciones para las tres regiones de operación del transistor NMOS ....	30
Tabla 7. . Ecuaciones para las tres regiones de operación del transistor PMOS ..	30
Tabla 8. Parámetros del transistor .....	30
Tabla 9. Especificaciones del hardware usado.....	31
Tabla 10. Parámetros y resultados del algoritmo con la función Himmelblau 2D. .	36
Tabla 11. Parámetros y resultados del algoritmo para la función Ackley 2D. ....	40
Tabla 12. Parámetros y resultados del algoritmo con la función Rosenbrock 2D. .	44
Tabla 13. Parámetros y resultados del algoritmo con la función Rosenbrock 3D. .	46
Tabla 14. Parámetros y resultados del algoritmo con la función Rosenbrock 4D. .	48
Tabla 14. Comparación de resultados entre el algoritmo de Búsqueda Armónica y la simulación para un circuito de c.c con un diodo.....	50
Tabla 15. Parámetro de inicialización para la ejecución del algoritmo de Búsqueda Armónica para la solución de un circuito c.c con un diodo. ....	51
Tabla 16. Comparación de resultados entre los métodos de optimización de Búsqueda Armónica y la Espiral de un circuito de c.c con un diodo.....	51
Tabla 17. Comparación de resultados entre el algoritmo de Búsqueda Armónica y la simulación de un amplificador Single con un MOSFET. ....	54
Tabla 18. Parámetro de inicialización para la ejecución del algoritmo de Búsqueda Armónica de un amplificador Single con un MOSFET.....	54
Tabla 19. Comparación de resultados entre el método de optimización de Búsqueda Armónica y el método de la Espiral de un amplificador Single con un MOSFET.....	54

Tabla 20. Comparación de resultados entre el algoritmo de Búsqueda Armónica y la simulación de un convertidor de corriente a voltaje de c.c. ....	58
Tabla 21. Parámetro de inicialización para la ejecución del algoritmo de Búsqueda Armónica de un convertidor de corriente a voltaje de c.c. ....	58
Tabla 22. Comparación de resultados entre los métodos de optimización de Búsqueda Armónica y la Espiral de un circuito convertidor de corriente a voltaje. ....	59
Tabla 23. Parámetros del circuito (n=0). ....	62
Tabla 24. Resultados y parámetros del algoritmo (n=0). ....	62
Tabla 25. Comparación de resultados entre el algoritmo (HS) y OrCAD. ....	63
Tabla 26. Parámetros del circuito (n=1). ....	63
Tabla 27. Resultados y parámetros del algoritmo (n=1). ....	63
Tabla 28. Comparación de resultados entre el algoritmo (HS) y OrCAD. ....	64
Tabla 29. Parámetros del circuito (n=8). ....	64
Tabla 30. Resultados y parámetros del algoritmo (n=8). ....	65
Tabla 31. Comparación de resultados entre el algoritmo (HS) y OrCAD. ....	66
Tabla 32. Parámetros del circuito (n=13). ....	67
Tabla 33. Resultados y parámetros del algoritmo (n=13). ....	67
Tabla 34. Comparación de resultados entre el algoritmo (HS) y OrCAD. ....	69

## ANEXOS

ANEXO A. Tutorial para la ejecución del algoritmo de búsqueda armónica .....	75
ANEXO B. Función Rosenbrock para diez, veinte y treinta variables .....	78

## RESUMEN

### TÍTULO

**ANÁLISIS DE LA EFICIENCIA DEL ALGORITMO HARMONY SEARCH EN LA SOLUCIÓN DE UN CIRCUITO ELECTRÓNICO NO LINEAL\***

### AUTORES

Saul Enrique González Aguinaga<sup>+</sup>

Oscar Iván Trasladino Salazar<sup>+</sup>

### PALABRAS CLAVE

Búsqueda armónica, Metaheurística, Circuito de corriente continua, Sistemas de ecuaciones no-lineales, Solución del modelo.

### DESCRIPCIÓN

Basado en el algoritmo de optimización *HARMONY SEARCH* desarrollado por *Kang Seok Lee* y *Zong Woo Geem* [1] que está inspirado en la improvisación musical, el presente informe de investigación expone los principales resultados de su aplicación para resolver las ecuaciones de los modelos matemáticos que describen circuitos de c.c. *no-lineales*. Se implementó el método de Búsqueda Armónica en el software *Matlab®*, y se utilizó para resolver de forma satisfactoria las ecuaciones correspondientes a funciones típicas de prueba, como la de *Himmelblau*, *Rosenbrock* y *Ackley*. Además se modelaron y resolvieron algunos circuitos de c.c. con elementos no-lineales, como diodos, y *MOSFETs*. Entre los circuitos modelados se encuentra, un circuito amplificador *SINGLE* con un *MOSFETs*, un circuito convertidor de tensión a corriente con *MOSFETs* y uno de corriente continua con diodos. Para este último circuito en particular, se obtienen de dos a quince variables según el número de lazos de carga que se adicionen al circuito base. Con el algoritmo implementado en *Matlab®* se resuelven dichas ecuaciones, obteniéndose resultados satisfactorios. Este hecho se refleja en los valores de los errores obtenidos al ejecutar el programa y evaluar las soluciones en las mismas ecuaciones que se utilizaron dentro del programa; también al compararlos con los generados con *OrCAD CAPTURE®*.

---

\* Trabajo de grado en la modalidad investigación creado y desarrollado en coparticipación con el profesor Rodrigo Correa y el estudiante de doctorado Iván Amaya.

<sup>+</sup> Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Carlos Rodrigo Correa. Co-director: Iván Mauricio Amaya.

## ABSTRACT

### TITLE

**ANALYSIS OF THE EFFICIENCY OF HARMONY SEARCH ALGORITHM IN SOLVING A NONLINEAR ELECTRONIC CIRCUIT \***

### AUTHORS

Saul Enrique González Aguinaga <sup>+</sup>

Oscar Ivan Trasladino Salazar <sup>+</sup>

### KEYWORDS

Harmony Search, Metaheuristic, DC Circuits, Non-linear equations systems, Model solution.

### DESCRIPTION

Based on an algorithm of optimization *HARMONY SEARCH* developed by *Kang Seok Lee y Zong Woo Geem* [1], which is inspired in musical improvisation, the present document expound the principal results of the application of such algorithm to solve the mathematical equations which describe c.c. circuit non- lineal. Was implemented the harmony search method program in *Matlab®*, based on the before mentioned algorithm and it is used to solve in a satisfactory way, the equations that correspond to typical proof functions, such the ones of *Himmelblau, Ackley and Rosenbrock*. Too, was modeled and solved some cc circuits with non-lineal elements such as diodes y *MOSFETs*. Among the modeled circuits it is found an amplifier SINGLE with *MOSFETs*, a converter circuit tension-current with *MOSFETs* and a current continuos circuit with diodes. For this last circuit in particular, there are obtained from two to fifteen variables depending on the number of charge ties added, and with the algorithm implemented in *Matlab®* we solved the before mentioned equations getting satisfactory results. This fact, that is reflected in the values of the obtained mistakes when it is executed the program and evaluated the solutions in the same equations that were used inside the program, and also when we compare them with the results of *OrCAD Capture®* for the same circuits.

---

\* Degree work in research mode created and developed in partnership with Professor Rodrigo Correa and doctoral student Ivan Amaya.

<sup>+</sup> Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Advisor: Carlos Rodrigo Correa. Co-Advisor: Iván Mauricio Amaya.

## INTRODUCCIÓN

En ingeniería, la solución de sistemas de ecuaciones es un problema muy común y de tal importancia, que a través del tiempo se han desarrollado varios métodos fundamentados en diferentes axiomas. En la electrónica los circuitos involucran elementos no lineales que para su correspondiente análisis se acude a un modelo matemático existente. Estos circuitos pueden ser resueltos por métodos analíticos tradicionalmente conocidos, pero cuando estos modelos son muy complejos o el circuito presenta muchas variables ya no es tan sencillo y eficiente resolverlos por tales métodos. El presente trabajo de grado ofrece otra opción en la solución de circuitos electrónicos no lineales empleando un método de optimización. El algoritmo seleccionado es *Búsqueda Armónica* (Harmony Search), denominado así por su similitud con la improvisación musical que según los creadores de éste, constituye su referente.

Este trabajo de grado inicia con la implementación del algoritmo en el software MATLAB®. Una vez implementado, el siguiente paso fue verificar tanto su viabilidad como precisión y eficiencia frente a funciones tradicionalmente usadas en la prueba de nuevos algoritmos de optimización. Las funciones de pruebas seleccionadas fueron las de *Himmelblau*, *Ackley* y *Rosenbrock*. Se evaluó su eficiencia ejecutándolo con cambios en los parámetros propios del método.

Posteriormente se procede a resolver los diferentes modelos de circuitos electrónicos no lineales, los cuales varían en características como: el número de variables, el modelo matemático de los elementos no lineales y su topología, entre otras. Cada una de las respuesta dadas por el método de búsqueda armónica se valida con base en la simulación de dichos circuitos con la herramienta *OrCAD Capture*®. Los análisis y comparaciones mencionadas anteriormente se muestran en tablas y gráficas.

Con el objetivo de tener un mejor análisis de las ventajas y desventajas de este algoritmo de optimización se realiza una comparación entre los métodos de búsqueda armónica y el método de la espiral. Se observan los parámetros más importantes al finalizar la ejecución, siendo estos: precisión de la respuesta, error y tiempo de ejecución principalmente.

Se propone un circuito escalable que es implementado en la plataforma MATLAB®, es decir, un circuito que mantenga su topología general pero se pueda aumentar su número de lazos con el simple ajuste en un parámetro. Esto se

realiza con el fin de poner a prueba el método y conocer cómo cambia su rendimiento conforme se aumenta el número de variables. Los resultados de estas variaciones son comparados con simulaciones hechas en *OrCAD Capture®*.

De los resultados obtenidos se resalta la disponibilidad de este software acorde con el objetivo central del presente trabajo de grado que se probó dando excelentes resultados frente a funciones objetivos de óptimos conocidos. Posteriormente se utilizó con una gran variedad de circuitos eléctricos lográndose resultados comparables con los obtenidos frente al software comercial.

## 2. EL TEMA QUE SE ABORDA

Un circuito es una red electrónica que contiene al menos una trayectoria cerrada. Los circuitos que contienen solo componentes lineales pueden analizarse por métodos algebraicos sencillos para determinar su comportamiento en corriente directa o en corriente alterna. Las redes electrónicas son generalmente no lineales y requieren diseños y métodos de análisis más complejos. El presente proyecto de grado aborda la evaluación de la efectividad que tiene el algoritmo Harmony Search para resolver un circuito electrónico no lineal.

A continuación se transcriben los objetivos aprobados para el presente trabajo de grado en la modalidad investigación.

### 2.1 Objetivo General <sup>1</sup>

Evaluar la eficiencia del algoritmo metaheurístico Harmony Search en la solución del modelo matemático de circuitos electrónicos no lineales. (*Ver páginas 49 -70*)

### 2.2 Objetivos Específicos <sup>1</sup>

1. Selección de un circuito con una estructura adecuada. (*Ver páginas 49 - 66*).
2. Implementar el algoritmo Harmony Search para la solución de un circuito electrónico no lineal usando la herramienta software Matlab (*Ver páginas 23 - 29*).
3. Analizar la respuesta dada por el algoritmo frente a una solución analítica y/o numérica disponible (*Ver páginas 49 - 66*).
4. Cuantificar el desempeño del algoritmo desarrollado en términos de tiempo de cómputo y precisión de la respuesta (*Ver páginas 49 - 66*).

<sup>1</sup> *Esto constituyó un aporte intelectual colaborativo de los estudiantes González y Trasladino junto con los directores Rodrigo Correa e Iván Amaya.*

### 3. FUNDAMENTACIÓN TEÓRICA

Los fundamentos teóricos que se presentan en este trabajo de grado se hacen con el objetivo de facilitar al lector la interpretación de los métodos usados. En esta sección se recopila el trabajo realizado con anterioridad en nuestro grupo de investigación y que constituye una creación intelectual que ha evolucionado con varios trabajos de grado, creados y dirigidos por el profesor Rodrigo Correa y el estudiante de doctorado Iván Amaya. Para las referencias concretas se invita al lector a consultar [2], [3], [4], [5], [6], [7].

A continuación se describe el algoritmo Harmony Search (“*Búsqueda Armónica*”) junto con la estrategia de solución utilizada.

#### 3.1 Estrategia de solución

El procedimiento empleado para la solución de un sistema de ecuaciones por medio de un método de optimización de Búsqueda Armónica se describe a continuación:

1. Se define un sistema de  $n$  ecuaciones y  $n$  incógnitas (1), como la representación del modelo matemático de un circuito eléctrico no lineal.

$$\begin{aligned} f_1 &= (x_1, x_2, \dots, x_n) = 0 \\ f_2 &= (x_1, x_2, \dots, x_n) = 0 \\ &\vdots \\ f_m &= (x_1, x_2, \dots, x_n) = 0 \end{aligned} \quad (1)$$

2. Para la solución del sistema de ecuaciones (1), se recurre a un método de optimización. La estrategia numérica se basa en la búsqueda del valor mínimo de la función objetivo descrita de la siguiente forma:

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^m (f_i(x_1, x_2, \dots, x_n))^2 \quad (2)$$

3. Tomando como base lo anterior, se puede utilizar el algoritmo de búsqueda armónica, para obtener la solución de la función objetivo.

Se describe la estructura básica y funcionamiento del algoritmo de búsqueda armónica (HARMONY SEARCH), a continuación.

### 3.2 Método de optimización de Búsqueda Armónica (Harmony Search)

El algoritmo de búsqueda armónica, conocido como HS por sus siglas en inglés (Harmony Search), es un algoritmo metaheurístico que basa su funcionamiento en el proceso de improvisación musical. La improvisación musical es una característica musical que no todo músico posee. Es un proceso en el cual la experiencia y el conocimiento previo de las armonías aportan a la calidad de la pieza que se está tocando, [8].

La primera versión de este algoritmo fue creada por Zong Woo Geem y Kang Seo Lee [9] en el 2001 para problemas de sistemas discretos. Posteriormente, en 2005 se propuso la versión para problemas de sistemas continuos.

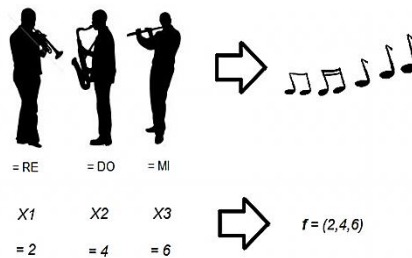
Búsqueda armónica simula el proceso de improvisación musical, en el cual los músicos buscan producir una armonía agradable para el oído humano mediante la acción de la improvisación, donde el músico realiza cada uno de los siguientes pasos, [8]:

1. Toca alguna melodía conocida que ha aprendido anteriormente.
2. Toca algo similar a la melodía anteriormente mencionada, para luego ajustarla poco a poco al tono deseado.
3. Compone una melodía basándose en sus conocimientos para seleccionar nuevas notas aleatorias.

Estos tres pasos corresponden a los elementos del algoritmo: Uso de la memoria armónica, ajuste del tono y aleatoriedad, respectivamente. La analogía entre la improvisación musical y la optimización en ingeniería se encuentra en la Figura 1.

**Figura 1. Analogía entre la improvisación musical y la optimización en ingeniería.**

*(Fuente: autores)*



De acuerdo como lo expresan sus autores en [1], “en la improvisación musical, cada músico toca cualquier nota de un posible rango, juntos hacen un vector armonía. Si todas las notas hacen una buena armonía, esta experiencia es almacenada en la memoria de cada músico, aumentando la posibilidad de hacer una nueva armonía la próxima vez. Similarmente en ingeniería cada variable decisiva inicia eligiendo cualquier valor dentro de su posible rango, juntas conforman un vector solución. Si todos los valores de las variables decisivas producen una buena solución, esta experiencia es almacenada en la memoria de cada variable, incrementando la posibilidad de crear una buena solución la próxima vez”.

### 3.3 Descripción del algoritmo búsqueda armónica (Harmony Search):

El proceso para crear la versión continua del algoritmo búsqueda armónica es el siguiente, [8]:

1. **Inicializar los parámetros del problema y los parámetros de búsqueda armónica:** El problema de optimización se define como minimizar (o maximizar)  $f(x)$  tal que  $LB_i < x_i < UB_i$  donde,  $f(x)$  es la función objetivo,  $x$  es una solución candidata que consiste de  $N$  variables de decisión ( $x_i$ ), y  $LB_i$  y  $UB_i$  son el limite de decisión mas bajo y el mas alto de cada variable, respectivamente. Los parámetros de búsqueda armónica se especifican en este paso. Estos parámetros son el tamaño de la memoria armónica (HMS), la tasa de consideración de la memoria armónica (HMCR), la tasa de ajuste del tono (PAR), el ancho de banda de ajuste del tono (BW) y el número de improvisaciones (NI).
2. **Inicializar la memoria armónica:** La memoria armónica inicial es generada desde una distribución uniforme en los rangos  $[LB_i, UB_i]$ , donde  $1 < i < N$ . Esto se realiza de la siguiente manera  $x_i^j = LB_i + r \times (UB_i - LB_i)$ , donde  $j = 1, 2, \dots, HMS$  y  $r \sim U(0,1)$ . La variable  $r$  hace referencia a un número aleatorio y  $U(0,1)$  a la función que genera el número aleatorio uniforme entre cero (0) y uno (1).
3. **Improvisar la nueva armonía:** El proceso de generación de una nueva armonía es llamado *improvisación*. El nuevo vector armónico,  $x^* = x_1^*, x_2^*, \dots, x_N^*$  se genera utilizando las siguientes reglas: consideración de la memoria, ajuste del tono y selección aleatoria. Este procedimiento se muestra a continuación [8]. En la línea 006,  $r$  es un número aleatorio

uniforme entre 0 y 1, y el valor bw es un ancho de banda arbitrario de la distancia para variables continuas.

```
001 para cada  $i \in [1, N]$  hacer
002   si  $U(0,1) < HMCR$  entonces   % consideración de la armonía
003     inicio
004        $x_i^j = x_i^j$  donde  $j \sim U(1, \dots, HMS)$ 
005       si  $U(0,1) \leq PAR$  entonces   % ajuste del tono
006          $x_i^j = x_i^j + r \times bw$ 
007       fin si
008     fin
009   si no   % selección aleatoria
010      $x_i^j = LB_i + r \times (UB_i - LB_i)$ 
011   fin si
012 fin para
```

4. **Actualizar la memoria armónica:** La nueva armonía,  $x^* = x_1^*, x_2^*, \dots, x_N^*$  reemplaza la peor armonía almacenada en la memoria armónica si el error es menor que el de la peor armonía.
5. **Verificar el criterio de parada:** Termina cuando el número máximo de improvisaciones (NI) se alcanza o el error logra la tolerancia deseada.

A manera de ejemplo se explica el funcionamiento del algoritmo. Se muestra paso a paso como se improvisa una nueva armonía [9].

Encontrar el mínimo de la siguiente ecuación:

$$\text{Min } f(x) = (x_1 - 2)^2 + (x_2 - 3)^4 + (x_3 - 1)^2 + 3 \quad (3)$$

### **Paso 1. Inicializar los parámetros del problema y los parámetros de búsqueda armónica.**

Al momento de iniciar el algoritmo este necesita de algunos parámetros los cuales son el HMS, PAR, HMRC, BW (para este ejemplo no es necesario el ancho de banda debido a que este parámetro solo es usado para variables continuas y no discretas) y el NI; los cuales son dados por la persona que lo va ejecutar.

Para este ejemplo se darán los siguientes valores:

**Tabla 1. Parámetros de inicialización de a memoria armónica. (Fuente: autores)**

PARAMETRO	VALOR
HMS	3
HMCR	0.9
PAR	0.3
Variables	3
Improvisaciones	100

Como se indicó anteriormente se eligen los límites inferior y superior de cada variable.

**Tabla 2. Límites inferior y superior de cada variable de decisión. (Fuente: autores)**

VARIABLE	$LB_i - UB_i$
$X_1$	1 – 5
$X_2$	2 – 4
$X_3$	1 – 4

### Paso 2. Inicializar la memoria armónica.

En este paso se procede a crear la memoria armónica la cual va tener 3 filas (HMS) con 3 columnas (número de Variables), para luego ser llenada fila a fila tomando un valor de forma aleatoria de cada rango permitido para las variables de decisión. La Figura 2, muestra la memoria armónica para este ejemplo.

**Tabla 3. Valores aleatorios de la primera fila de la memoria armónica. (Fuente: autores)**

VARIABLE	$LB_i \leq x_i \leq UB_i$
$X_1$	$1 \leq 2 \leq 5$
$X_2$	$2 \leq 2 \leq 4$
$X_3$	$1 \leq 1 \leq 4$

Figura 2. Inicialización de la memoria armónica. (Fuente: autores)

	$x1$	$x2$	$x3$	$F$
$HS(1)$	2	2	1	4
$HS(2)$	1	3	4	13
$HS(3)$	5	3	3	16

### Paso 3. Improvisar la nueva armonía.

Para improvisar una nueva armonía se tienen en cuenta los siguientes parámetros: consideración de la memoria armónica y ajuste de tono. La primera tarea que realiza el algoritmo para obtener un nuevo vector armónico es generar un número aleatorio entre 0 – 1 para cada variable. Dependiendo del valor de los números obtenidos el algoritmo procede de alguna de las siguientes formas.

#### Caso 1.

$$U_1(0,1) > \text{Consideracion de la memoria(HMCR)}$$

Si esto ocurre la variable del nuevo vector armónico tomara aleatoriamente un valor de su rango permitido, en este caso no se usara el ajuste de tono (PAR).

#### Caso 2.

$$U_1(0,1) < \text{Consideracion de la memoria(HMCR)} \quad U_2(0,1) > \text{Ajuste de tono(PAR)}$$

Si esto ocurre la variable del nuevo vector armónico tomara aleatoriamente de la memoria armónica un valor de los ya almacenados para esta. Una vez hecho esto se genera un segundo numero aleatorio también entre 0 – 1, si este número es mayor al ajuste de tono, el nuevo valor de la variable no es modificado.

### Caso 3.

$$U_1(0,1) < \text{Consideracion de la memoria(HMCR)} \quad U_2(0,1) \leq \text{Ajuste de tono(PAR)}$$

Si esto ocurre la variable del nuevo vector armónico tomara aleatoriamente de la memoria armónica un valor de los ya almacenados para esta. Una vez hecho esto se genera un segundo numero aleatorio también entre 0 – 1, si este número es menor al ajuste de tono, el nuevo valor de la variable es modificado.

Para este caso tendremos entonces:

**Tabla 4. Valores generados aleatoriamente y parámetros anteriores. (Fuente: autores)**

VARIABLE	VALOR
$U_1$	0.7
$U_2$	0.5
HMCR	0.9
PAR	0.3

Para el primer número aleatorio es 0.7. Este número es menor que el HMCR lo cual indica que la variable del nuevo vector armónico tomara uno de los valores ya almacenados. Posteriormente al comparar el segundo número aleatorio con el PAR nos indica que el valor de la variable del nuevo vector no será modificado. Por tanto la nueva armonía podría conformarse de la siguiente manera [1 2 3].

### Paso 4. Actualizar la memoria armónica.

Para este paso se comparan las respuestas según el criterio del problema, (minimización) y si el nuevo vector armónico es mejor, se procede a remplazar el peor vector por este.

En este punto el nuevo vector [1 2 3] da un numero mas pequeño, lo que indica que este debe remplazar al vector [5 3 3]. La Figura 3, muestra la memoria armónica actualizada.

Figura 3. Memoria armónica actualizada. (Fuente: autores)

	$x_1$	$x_2$	$x_3$	$F$
$HS(1)$	2	2	1	4
$HS(2)$	1	2	3	9
$HS(3)$	1	3	4	13

**Paso 5. Verificar el criterio de parada.**

Finalmente se verifica si el algoritmo debe terminar, puede ser por el número de iteraciones o cuando el error de la respuesta llega a una tolerancia específica.

Para este ejemplo el algoritmo seguiría buscando una mejor respuesta por lo tanto regresaría a los pasos 3 y 4 de nuevo hasta cuando termine con todas las iteraciones; para efectos prácticos la siguiente iteración o improvisación podría generar el vector  $[2 \ 3 \ 1]$ , el cual al momento de ser remplazado en la función objetivo daría un valor de 3 que es el mínimo global de esta función.

### 3.4 Modelo matemático de los dispositivos no-lineales utilizados

Para el desarrollo de este trabajo de grado se han utilizado los siguientes modelos matemáticos de los dispositivos electrónicos utilizados:

#### 3.4.1 Diodo

El modelo matemático del diodo [10], [11], se presenta en la ecuación (4).

$$I = I_s \cdot (e^{\frac{V_D}{nV_T}} - 1) \quad (4)$$

Dónde:

**Tabla 5. Parámetros del diodo. (Fuente: autores)**

<b>Parámetros del diodo</b>	
$V_D$	<i>Voltaje del diodo</i>
$n$	<i>Coeficiente de emisión</i>
$I_s$	<i>Corriente de saturación</i>
$V_T$	<i>Voltaje térmico</i>
$K = 1.381 \times 10^{-23} \text{ J/K}$	<i>Constante de Boltzman</i>
$q = 1.602 \times 10^{-19} \text{ c}$	<i>Carga del electrón</i>

#### 3.4.2 Transistor MOSFET

El modelo matemático del transistor MOSFET [10], [11], se presenta para tres regiones de operación: saturación, triodo y corte para el dispositivo NMOS Y PMOS. Las ecuaciones son presentadas en la Tabla 6 y Tabla 7:

Tabla 6. Ecuaciones para las tres regiones de operación del transistor NMOS. (Fuente: autores)

**Transistor NMOS**

---

$$i_D = \begin{cases} \frac{1}{2} K_n \frac{W}{L} (V_{GS} - V_t)^2 (1 + \lambda V_{DS}) & V_{DS} \geq V_{GS} - V_t \text{ y } V_{GS} > V_t \\ K_n \frac{W}{L} ((V_{GS} - V_t)V_{DS} - \frac{1}{2} V_{DS}^2) & V_{DS} < V_{GS} - V_t \text{ y } V_{GS} > V_t \\ 0 & V_{GS} < V_t \end{cases}$$

Tabla 7. Ecuaciones para las tres regiones de operación del transistor PMOS. (Fuente: autores)

**Transistor PMOS**

---

$$i_D = \begin{cases} \frac{1}{2} K_p \frac{W}{L} (|V_{GS}| - |V_t|)^2 (1 + \lambda |V_{DS}|) & |V_{DS}| \geq |V_{GS}| - |V_t|, |V_{GS}| > |V_t| \\ K_p \frac{W}{L} ( (|V_{GS}| - |V_t|) |V_{DS}| - \frac{1}{2} V_{DS}^2 ) & |V_{DS}| < |V_{GS}| - |V_t|, |V_{GS}| > |V_t| \\ 0 & |V_{GS}| < |V_t| \end{cases}$$

Dónde:

Tabla 8. Parámetros del transistor. (Fuente: autores)

<b>Parámetros del Transistor</b>	
$K_n, K_p$	<i>Parámetros de transconductancia</i>
$u_n, u_p$	<i>Movilidad efectiva de los portadores de carga</i>
$C_{ox}$	<i>Capacitancia del óxido por unidad de carga</i>
$W$	<i>Ancho de la puerta</i>
$L$	<i>Longitud de la puerta</i>
$V_{GS}$	<i>Tensión diferencial de puerta a fuente</i>
$V_{DS}$	<i>Tensión diferencial de dren a fuente</i>
$\lambda$	<i>Modulación de longitud de canal</i>
$V_t$	<i>Tensión de umbral</i>

## 4. EXPERIMENTACIÓN

En esta sección se describen las simulaciones llevadas a cabo. Los análisis y resultados que se muestran a continuación son una serie de pruebas que se hicieron con distintas funciones objetivo. Se inicia verificando el correcto funcionamiento del algoritmo con funciones de prueba ya conocidas en la literatura como: Himmelblau, Ackley y Rosenbrock. Los resultados de estas pruebas se muestran en la sección 3.2.

Posteriormente se da solución a cinco topologías distintas de circuitos no lineales con el algoritmo de búsqueda armónica (HS). Estas topologías incluyen dispositivos no lineales como el diodo y el MOSFET.

Las especificaciones del ordenador utilizado para realizar dichas pruebas se muestran en la Tabla 9.

**Tabla 9. Especificaciones del hardware usado. (Fuente: autores)**

<b>Fabricante</b>	Toshiba®
<b>Modelo</b>	Satellite™ A665-S6092
<b>Procesador</b>	Intel® Core™ i7 Q720 @ 1.60GHz
<b>RAM</b>	4.00 GB
<b>Sistema operativo</b>	Microsoft® Windows™ 7 Ultimate 64-bit

El software utilizado para la implementación del algoritmo es la herramienta de computo MathWorks® Matlab® 7.04, donde se desarrolló cada uno de los guiones para el algoritmo y sus correspondientes pruebas.

### 4.1 Convenciones para parámetros y resultados

Las siguientes convenciones, tienen el objetivo de hacer comprensible los parámetros y resultados de la ejecución del algoritmo de búsqueda armónica. En el Anexo A se presenta de forma detallada el procedimiento de utilización del algoritmo.

***XL<sub>i</sub>***: límite inferior de la variable  $i$

***XU<sub>i</sub>***: límite superior de la variable  $i$

***HMS***: Tamaño de la memoria armónica

***MaxImp***: Número máximo de iteraciones

***HMCR***: Tasa de la consideración de la memoria armónica

***PAR***: Tasa del ajuste del tono.

***Tol***: Tolerancia del error

***b***: Ancho de banda

***X<sub>i</sub>***: solución de la variable  $i$

***Error***: es el valor que se obtiene al evaluar la función objetivo en los resultados obtenidos para  $x_i$ .

***Ut***: última iteración del algoritmo

***N***: cantidad de variables de la función objetivo

***Te***: tiempo de ejecución del programa

## **4.2 Funciones de prueba**

Algunas de las funciones de prueba se presentan con el fin de dar una idea de las diferentes situaciones que los algoritmos de optimización tienen que enfrentar. Para esto se recurre a tres funciones Himmelblau, Ackley y Rosenbrock donde se desea buscar el mínimo global de cada una. Además se muestra la síntesis de una serie de pruebas donde se hace variar parámetros propios del algoritmo.

Las pruebas consisten en hacer variar los parámetros (*HMCR* y *PAR*) en un rango de 0.1 a 0.9 cada uno. El objetivo de estas pruebas es encontrar la mejor relación (*HMCR* – *PAR*) que conlleve al algoritmo a converger rápidamente y a la vez entregar una buena respuesta.

Las funciones de prueba, Himmelblau, Ackley y Rosenbrock son problemas de optimización de un solo objetivo, para efectos de este trabajo de grado no se tratarán funciones multi-objetivo.

### 4.2.1 Función Himmelblau

En optimización, la función de la Himmelblau es una función multi-modal, que se utiliza para probar el rendimiento de los algoritmos de optimización. La ecuación para dos dimensiones está dada por:

$$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \quad (5)$$

Esta función posee cuatro mínimos globales, los cuales son:

$$f(x_1, x_2) = f(3.000000000000000, 2.000000000000000) = 0$$

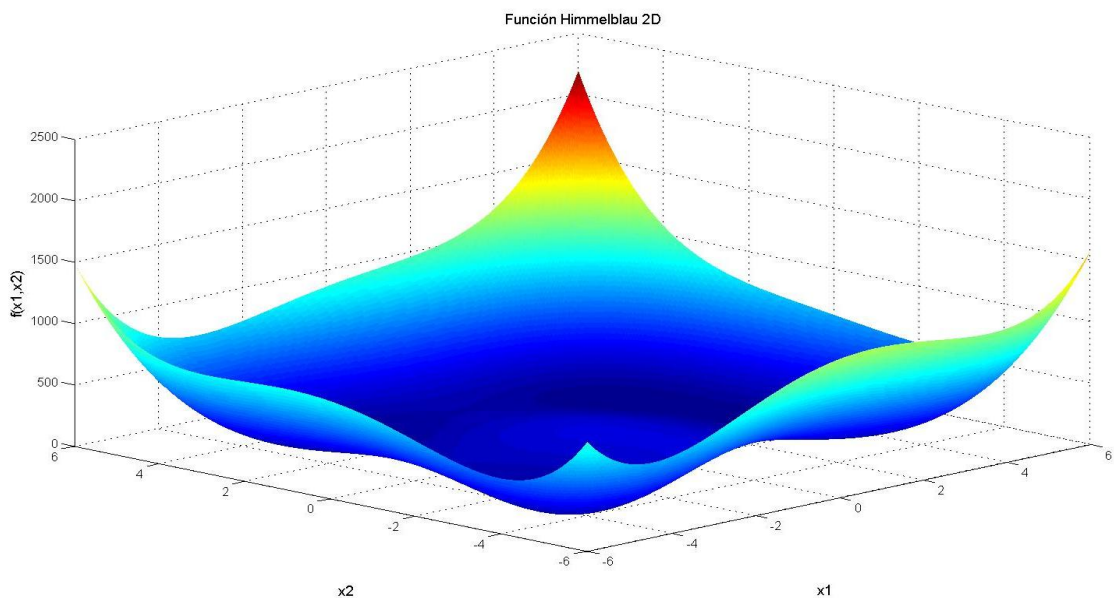
$$f(x_1, x_2) = f(-3.779310253377747, -3.2831859912861) = 0$$

$$f(x_1, x_2) = f(-2.805118086952745, 3.131312518250573) = 0$$

$$f(x_1, x_2) = f(3.584428340330492, -1.848126526964404) = 0$$

La gráfica de esta función se presenta a continuación.

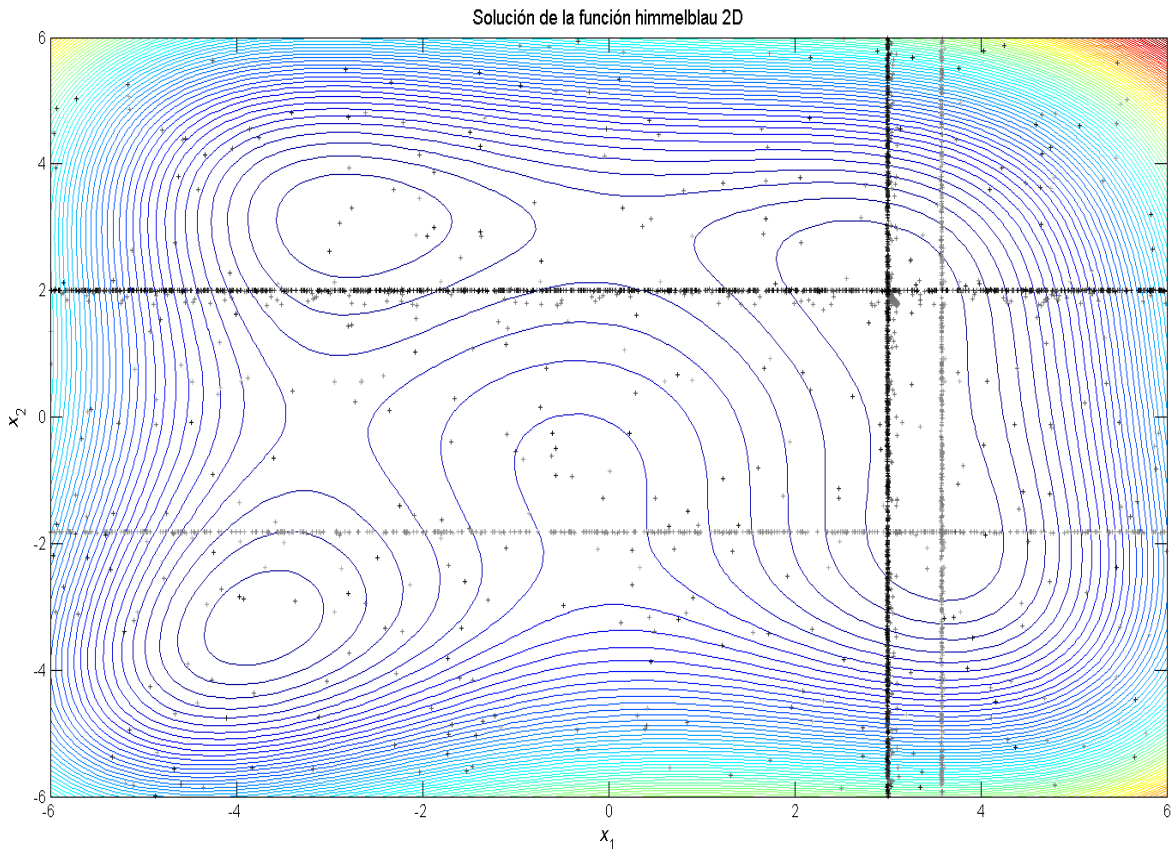
**Figura 4. Función Himmelblau 2D. (Fuente: autores)**



Para dar solución a esta función con el algoritmo de búsqueda armónica (HS), se escoge el mínimo global  $f(x_1, x_2) = f(3.0, 2.0) = 0$ , para hacer la respectiva comparación entre la solución del algoritmo y la solución analítica propuesta por la literatura.

La Figura 5 muestra una serie de puntos correspondientes al número de improvisaciones que realiza el algoritmo para dar solución a la función Himmelblau. Estos puntos se intensifican en  $f(3.0, 2.0)$  formando dos líneas rectas en  $x_1 = 3.0$  y  $x_2 = 2.0$ , la intersección de estas corresponden exactamente con el mínimo global  $f(x_1, x_2) = f(3.0, 2.0) = 0$  de la función anteriormente mencionada.

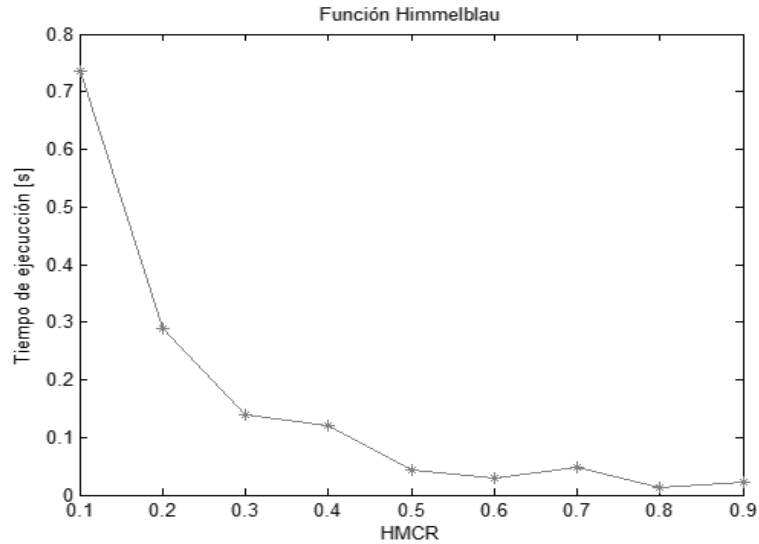
**Figura 5. Solución de la función de Himmelblau 2D. (Fuente: autores)**



Se enfatiza en el tiempo de convergencia del algoritmo debido a que los parámetros *HMCR* y *PAR* afectan significativamente a este. Este efecto no es tan evidente cuando se ajusta los otros parámetros (*HMS* y *b*). Además no se grafican los valores de las respuestas ya que estos no se afectan considerablemente con el cambio del *HMCR* y *PAR*.

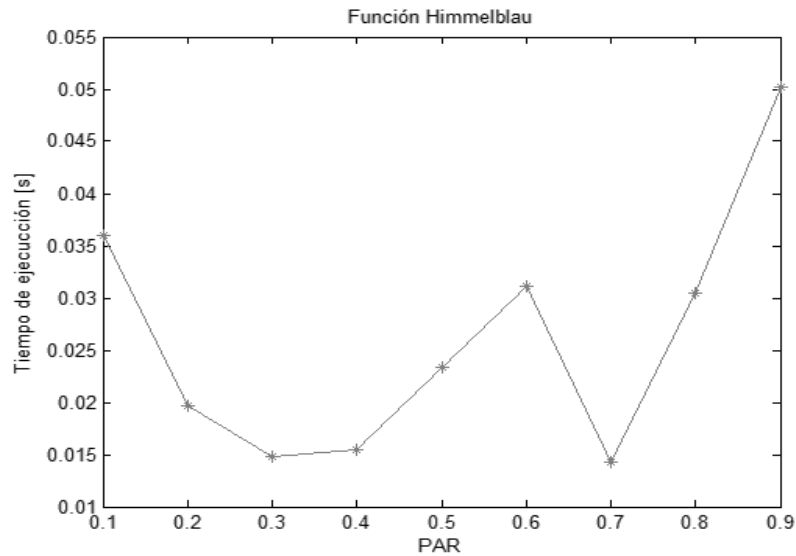
La Figura 6 sintetiza los resultados obtenidos de las pruebas realizadas al algoritmo, donde se hace variar el parámetro  $HMCR$  en un rango de 0.1 a 0.9, dejando constante el valor del  $PAR = 0.4$ .

**Figura 6. Variación del parámetro  $HMCR$  en la función Himmelblau. (Fuente: autores)**



La Figura 7 sintetiza los resultados obtenidos de las pruebas realizadas al algoritmo, donde se hace variar el parámetro  $PAR$  en un rango de 0.1 a 0.9, dejando constante el valor del  $HMCR = 0.8$ .

**Figura 7. Variación del parámetro  $PAR$  en la función Himmelblau. (Fuente: autores)**



La Tabla 10, muestra los parámetros utilizados y los resultados obtenidos del algoritmo de búsqueda armónica (HS), con la función Himmelblau. Se realizan tres ejecuciones del algoritmo, en cada una se modifica el parámetro *HMCR* (0.7, 0.8 y 0.9) con *PAR* (0.4) con el propósito de mostrar los tiempos de convergencia del algoritmo, error de las respuestas y el número de iteraciones empleadas.

Los valores de los parámetros anteriormente mencionados son la mejor relación encontrada para el algoritmo. Esta afirmación está basada en los análisis hechos a las variaciones de los parámetros *HMCR* y *PAR*.

**Tabla 10. Parámetros y resultados del algoritmo con la función Himmelblau 2D.**

(Fuente: autores)

	<b>Ejecución 1</b>	<b>Ejecución 2</b>	<b>Ejecución 3</b>
<b><i>X1</i></b>	2.999886044130365	3.58433499442985	3.584299002286712
<b><i>X2</i></b>	2.000027183562257	-1.848156603301847	-1.848098719678995
<b><i>Error</i></b>	0.43107028230e-06	0.11486243483e-06	0.86276748617e-06
<b><i>Ut</i></b>	2840	770	962
<b><i>XL1;XU1</i></b>	-6; 6	-6; 6	-6; 6
<b><i>XL2;XU2</i></b>	-6; 6	-6; 6	-6; 6
<b><i>HMS</i></b>	10	10	10
<b><i>HMCR</i></b>	0.7	0.8	0.9
<b><i>PAR</i></b>	0.4	0.4	0.4
<b><i>b</i></b>	0.001	0.001	0.001
<b><i>MaxImp</i></b>	10000000	10000000	10000000
<b><i>Tol</i></b>	1e-6	1e-6	1e-6
<b><i>Te</i></b>	0.04904047 [s]	0.01382773 [s]	0.02164031 [s]

#### 4.2.2 Función Ackley

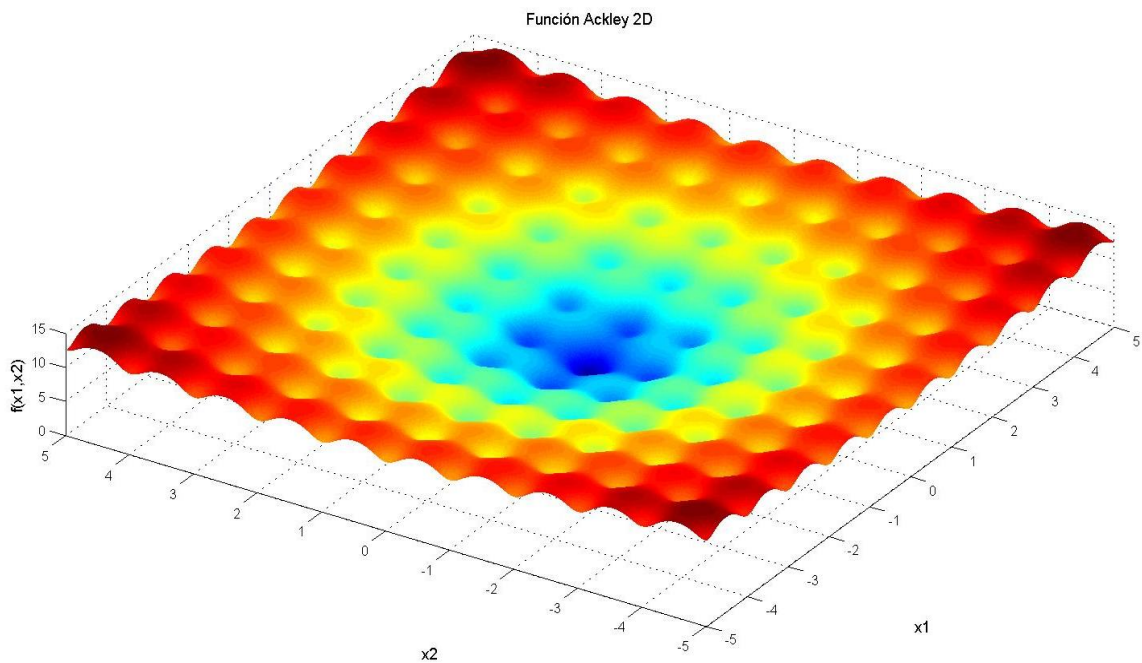
En optimización, la función de la Ackley es un Problema de minimización, que se utiliza para probar el rendimiento de los algoritmos de optimización, La ecuación para dos dimensiones esta dada por:

$$f(x) = 20(1 - \exp(-0.2 \cdot \sqrt{0.5(x_1^2 + x_2^2)})) - \exp(0.5(\cos(2 \cdot \pi \cdot x_1) + \cos(2 \cdot \pi \cdot x_2))) + \exp(1); \quad (6)$$

El mínimo global se encuentra  $f(x_1, x_2) = f(0,0) = 0$ .

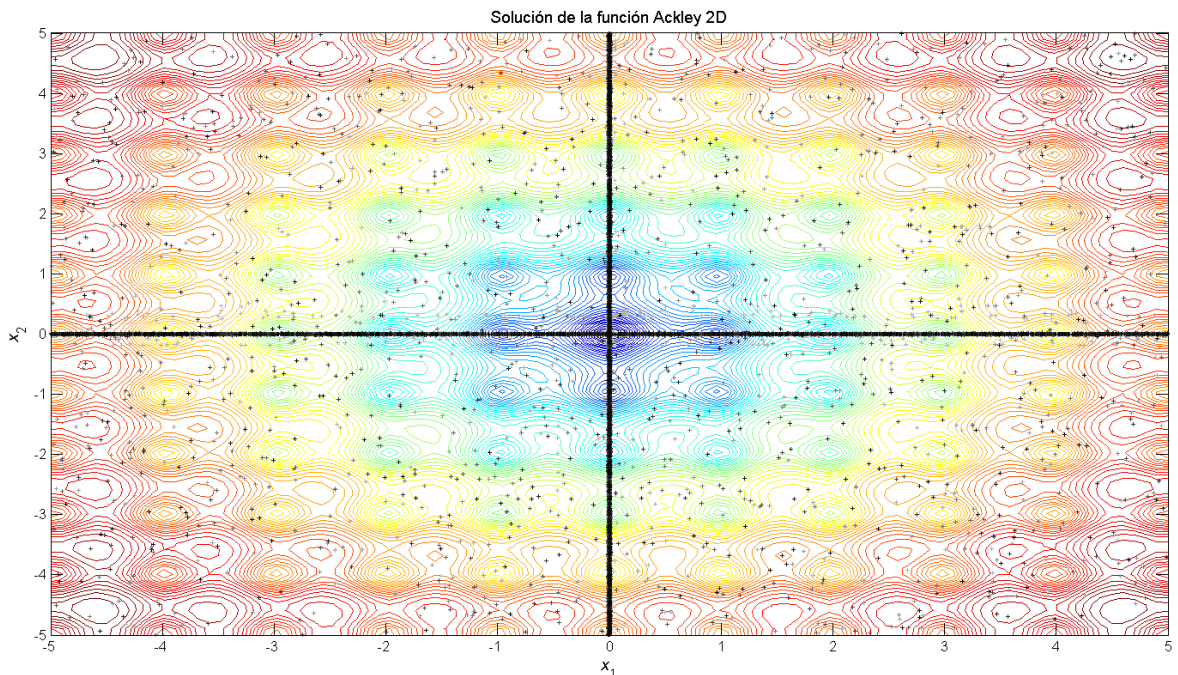
La gráfica de esta función se presenta a continuación.

**Figura 8. Función Ackley 2D. (Fuente: autores)**



La Figura 9 muestra una serie de puntos correspondientes al número de improvisaciones que realiza el algoritmo para dar solución a la función Ackley. Estos puntos se intensifican en  $f(0.0,0.0)$  formando dos líneas rectas en  $x_1 = 0.0$  y  $x_2 = 0.0$ , la intersección de estas corresponden exactamente con el mínimo global  $f(x_1, x_2) = f(0.0,0.0) = 0$  de la función anteriormente mencionada.

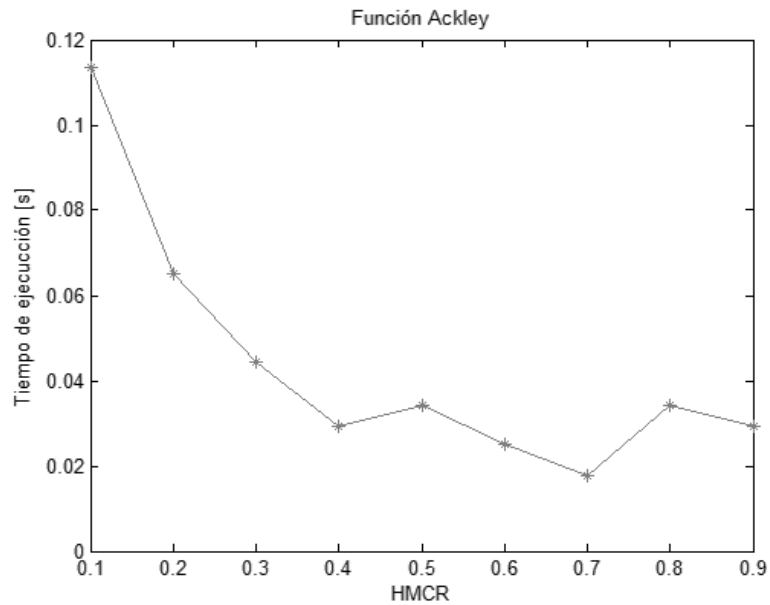
Figura 9. Solución de la función de Ackley 2D. (Fuente: autores)



Se enfatiza en el tiempo de convergencia del algoritmo debido a que los parámetros *HMCR* y *PAR* afectan significativamente a este. Este efecto no es tan evidente cuando se ajusta los otros parámetros (*HMS* y *b*). Además no se grafican los valores de las respuestas ya que estos no se afectan considerablemente con el cambio del *HMCR* y *PAR*.

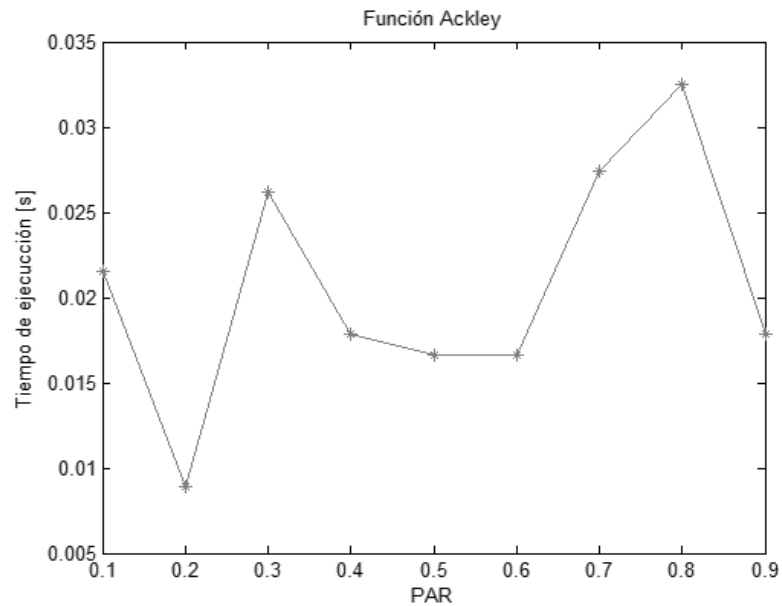
La Figura 10 sintetiza los resultados obtenidos de las pruebas realizadas al algoritmo, donde se hace variar el parámetro *HMCR* en un rango de 0.1 a 0.9, dejando constante el valor del *PAR* (0.4).

**Figura 10. Variación del parámetro HMCR en la función Ackley. (Fuente: autores)**



La Figura 11 sintetiza los resultados obtenidos de las pruebas realizadas al algoritmo, donde se hace variar el parámetro *PAR* en un rango de 0.1 a 0.9, dejando constante el valor del *HMCR* (0.8).

**Figura 11. Variación del parámetro PAR en la función Ackley. (Fuente: autores)**



La Tabla 11, muestra los parámetros utilizados y los resultados obtenidos del algoritmo de búsqueda armónica (HS), con la función Ackley. Se realizan tres ejecuciones del algoritmo, en cada una se modifica el parámetro *HMCR* (0.7, 0.8 y 0.9) con *PAR* (0.4) con el propósito de mostrar los tiempos de convergencia del algoritmo.

Los valores de los parámetros anteriormente mencionados son la mejor relación encontrada para el algoritmo. Esta afirmación esta basada en los análisis hechos a las variaciones de los parámetros *HMCR* y *PAR*.

**Tabla 11. Parámetros y resultados del algoritmo para la función Ackley 2D. (Fuente: autores)**

	<b>Ejecución 1</b>	<b>Ejecución 2</b>	<b>Ejecución 3</b>
<b><i>X1</i></b>	0	0	0
<b><i>X2</i></b>	0	0	0
<b><i>Error</i></b>	0	0	0
<b><i>Ut</i></b>	1442	830	1255
<b><i>XL1;XU1</i></b>	-3; 3	-3; 3	-3; 3
<b><i>XL2;XU2</i></b>	-3; 3	-3; 3	-3; 3
<b><i>HMS</i></b>	10	10	10
<b><i>HMCR</i></b>	0.7	0.8	0.9
<b><i>PAR</i></b>	0.4	0.4	0.4
<b><i>b</i></b>	0.001	0.001	0.001
<b><i>MaxImp</i></b>	10000000	10000000	10000000
<b><i>Tol</i></b>	1e-6	1e-6	1e-6
<b><i>Te</i></b>	0.029409651 [s]	0.0177467 [s]	0.034357447 [s]

### 4.2.3 Función Rosenbrock

En optimización, la función Rosenbrock es una función no-convexa utilizada como un problema de prueba de rendimiento de los algoritmos de optimización introducidos por Howard H. Rosenbrock en 1960. La ecuación para  $n$  dimensiones esta dada por:

$$f(x) = \sum_{i=1}^{N-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \quad (7)$$

La función Rosenbrock esta definida para dos o más variables. Esta función posee un mínimo global en  $x_i = 1, i = 1, 2, \dots, n$ ; y su valor mínimo es cero.

#### ***Función Rosenbrock para dos variables***

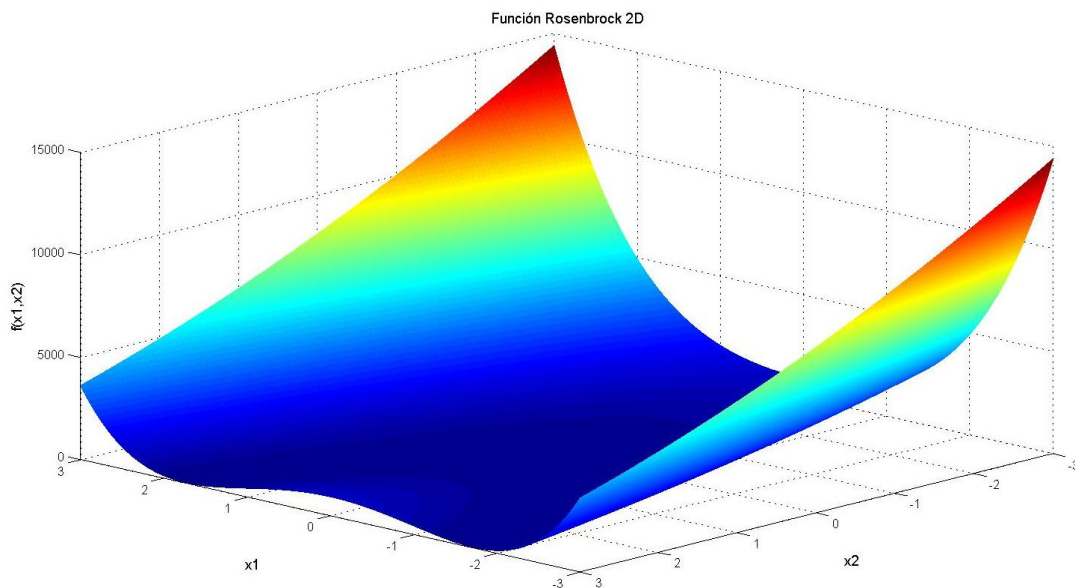
La ecuación para dos dimensiones esta dada por:

$$f(x) = (1 - x_1)^2 + 100(x_2 - x_1)^2 \quad (8)$$

El mínimo global se encuentra  $f(x_1, x_2) = f(1, 1) = 0$ .

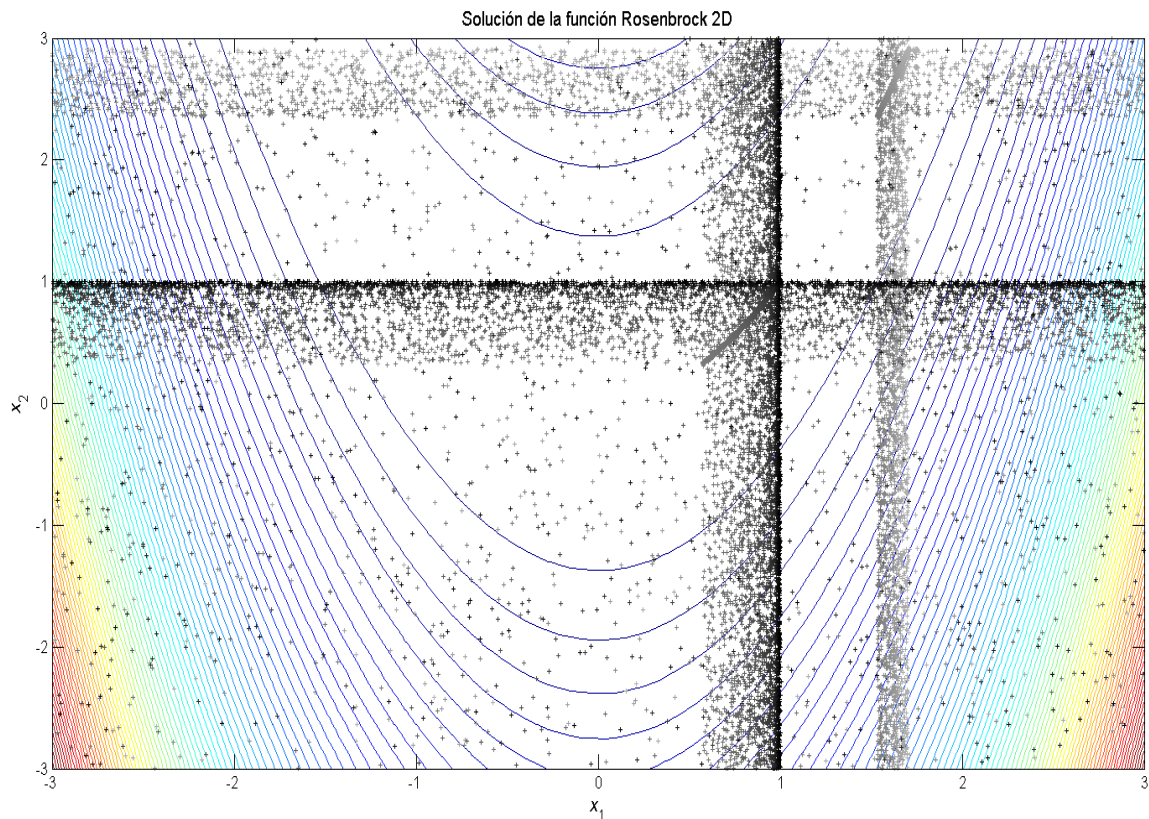
La gráfica de esta función se presenta a continuación.

**Figura 12. Función Rosenbrock 2D. (Fuente: autores)**



La Figura 13 muestra una serie de puntos correspondientes al número de improvisaciones que realiza el algoritmo para dar solución a la función Rosenbrock. Estos puntos se intensifican en  $f(1.0,1.0)$  formando dos líneas rectas en  $x_1 = 1.0$  y  $x_2 = 1.0$ , la intersección de estas corresponden exactamente con el mínimo global  $f(x_1, x_2) = f(1.0,1.0) = 0$  de la función anteriormente mencionada.

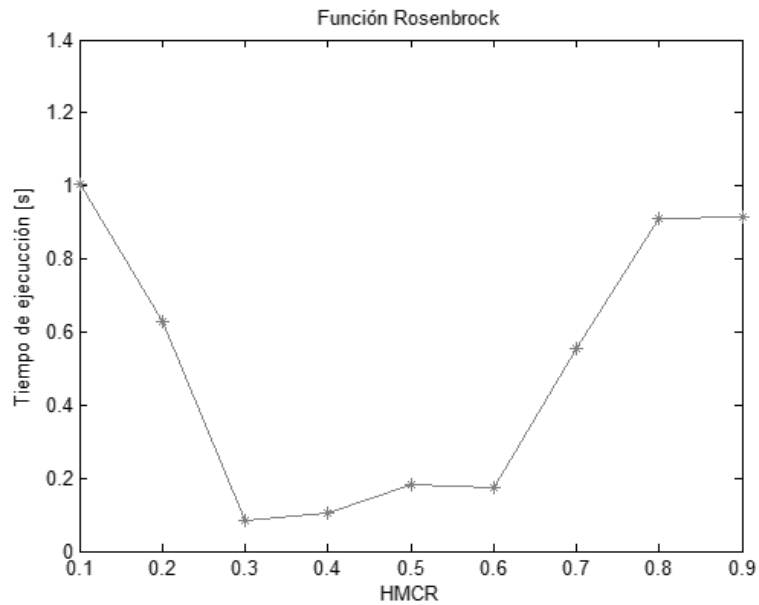
**Figura 13. Solución de la función de Rosenbrock 2D. (Fuente: autores)**



Se enfatiza en el tiempo de convergencia del algoritmo debido a que los parámetros *HMCR* y *PAR* afectan significativamente a este. Este efecto no es tan evidente cuando se ajusta los otros parámetros (*HMS* y *b*). Además no se grafican los valores de las respuestas ya que estos no se afectan considerablemente con el cambio del *HMCR* y *PAR*.

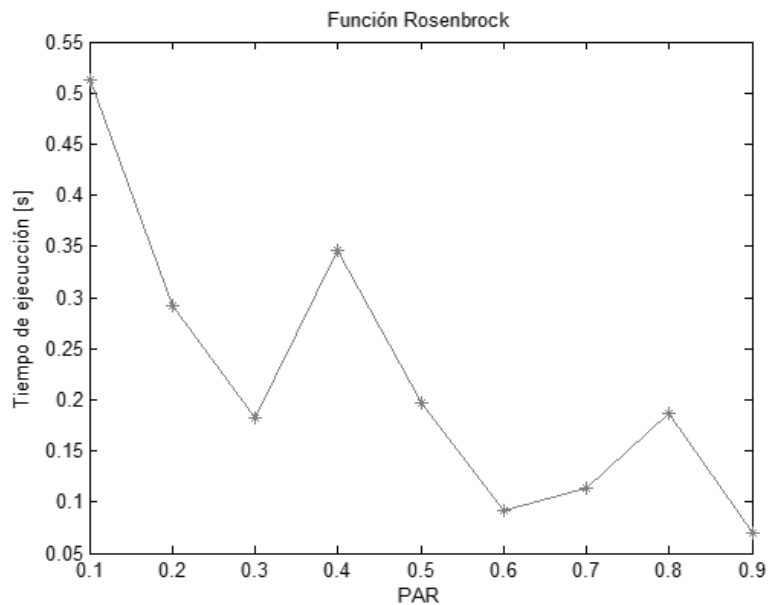
La Figura 14 sintetiza los resultados obtenidos de las pruebas realizadas al algoritmo, donde se hace variar el parámetro *HMCR* en un rango de 0.1 a 0.9, dejando constante el valor del *PAR* (0.4).

Figura 14. Variación del parámetro HMCR en la función Rosenbrock 2D. (Fuente: autores)



La Figura 15 sintetiza los resultados obtenidos de las pruebas realizadas al algoritmo, donde se hace variar el parámetro *PAR* en un rango de 0.1 a 0.9, dejando constante el valor del *HMCR* (0.8).

Figura 15. Variación del parámetro PAR en la función Rosenbrock 2D. (Fuente: autores)



La tabla 12 muestra los parámetros utilizados y los resultados obtenidos del algoritmo de búsqueda armónica (HS), con la función Rosenbrock 2D. Se realizan tres ejecuciones del algoritmo, en cada una se modifica el parámetro *HMCR* (0.7, 0.8 y 0.9) con *PAR* (0.4) con el propósito de mostrar los tiempos de convergencia del algoritmo.

Los valores de los parámetros anteriormente mencionados son la mejor relación encontrada para el algoritmo. Esta afirmación está basada en los análisis hechos a las variaciones de los parámetros *HMCR* y *PAR*.

**Tabla 12** Parámetros y resultados del algoritmo con la función Rosenbrock 2D. (Fuente: autores)

	Ejecución 1	Ejecución 2	Ejecución 3
<b>X1</b>	0.999202873886	1.00008431845	1.00031663355
<b>X2</b>	0.998397966324	1.00011067427	1.00063472988
<b>Error</b>	0.642494391801e-06	0.34315873147e-06	0.10044245456e-06
<b>Ut</b>	27292	44833	41315
<b>XL1;XU1</b>	-3; 3	-3; 3	-3; 3
<b>XL2;XU2</b>	-3; 3	-3; 3	-3; 3
<b>HMS</b>	10	10	10
<b>HMCR</b>	0.7	0.8	0.9
<b>PAR</b>	0.4	0.4	0.4
<b>b</b>	0.001	0.001	0.001
<b>MaxImp</b>	10000000	10000000	10000000
<b>Tol</b>	1e-6	1e-6	1e-6
<b>Te</b>	0.554956631 [s]	0.91268374813 [s]	0.9152172847 [s]

### ***Función Rosenbrock para tres variables***

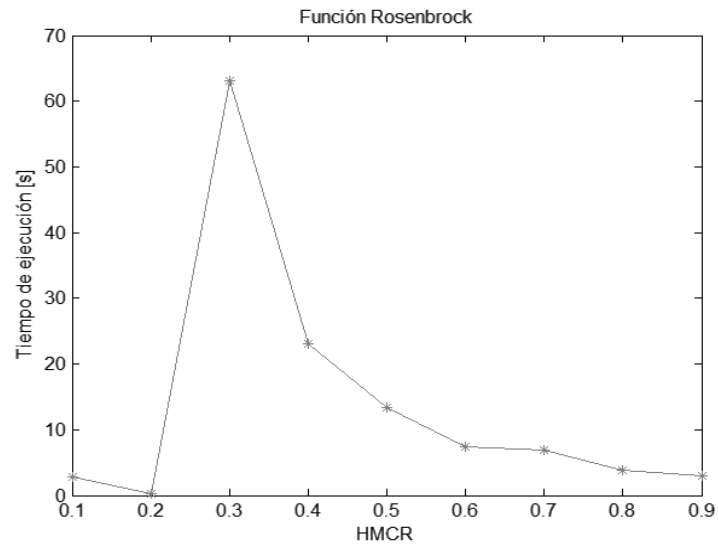
La ecuación para cuatro dimensiones está dada por:

$$f(x_1, x_2, x_3) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 100(x_3 - x_2^2)^2 + (1 - x_2)^2 \quad (9)$$

El mínimo global se encuentra  $f(x_1, x_2, x_3) = f(1, 1, 1) = 0$ .

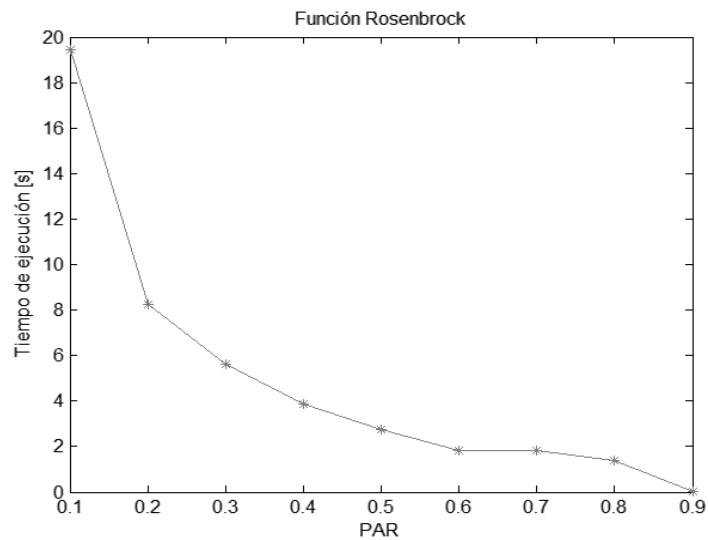
La Figura 16 sintetiza los resultados obtenidos de las pruebas realizadas al algoritmo, donde se hace variar el parámetro  $HMCR$  en un rango de 0.1 a 0.9, dejando constante el valor del  $PAR$  (0.4).

**Figura 16. Variación del parámetro  $HMCR$  en la función Rosenbrock 3D. (Fuente: autores)**



La Figura 17 sintetiza los resultados obtenidos de las pruebas realizadas al algoritmo, donde se hace variar el parámetro  $PAR$  en un rango de 0.1 a 0.9, dejando constante el valor del  $HMCR$  (0.8).

**Figura 17. Variación del parámetro  $PAR$  en la función Rosenbrock 3D. (Fuente: autores)**



La Tabla 13 reporta los parámetros y la solución a la ecuación (9) del algoritmo de búsqueda armónica (HS). Se realizan tres ejecuciones del algoritmo, en cada una se modifica el parámetro *HMCR* (0.7, 0.8 y 0.9) con *PAR* (0.4) con el propósito de mostrar los tiempos de convergencia del algoritmo.

Los valores de los parámetros anteriormente mencionados son la mejor relación encontrada para el algoritmo. Esta afirmación está basada en los análisis hechos a las variaciones de los parámetros *HMCR* y *PAR*.

**Tabla 13. Parámetros y resultados del algoritmo con la función Rosenbrock 3D. (Fuente: autores)**

	<b>Ejecución 1</b>	<b>Ejecución 2</b>	<b>Ejecución 3</b>
<b><i>X1</i></b>	1.00446470055	0.99559946690	1.00404667365
<b><i>X2</i></b>	1.00880385704	0.99113116153	1.00823353051
<b><i>X3</i></b>	1.01771725557	0.98234787623	1.01680882138
<b><i>Error</i></b>	0.99660438296e-04	0.98785029056e-04	0.93205345122e-04
<b><i>Ut</i></b>	366551	189771	155989
<b><i>XL1;XU1</i></b>	0; 2	0; 2	0; 2
<b><i>XL2;XU2</i></b>	0; 2	0; 2	0; 2
<b><i>XL3;XU3</i></b>	0; 2	0; 2	0; 2
<b><i>HMS</i></b>	10	10	10
<b><i>HMCR</i></b>	0.7	0.8	0.9
<b><i>PAR</i></b>	0.4	0.4	0.4
<b><i>b</i></b>	0.001	0.001	0.001
<b><i>MaxImp</i></b>	1000000000	1000000000	1000000000
<b><i>Tol</i></b>	1e-4	1e-4	1e-4
<b><i>Te</i></b>	6.873712 [s]	3.82817684 [s]	3.0578087 [s]

### ***Función Rosenbrock para cuatro variables***

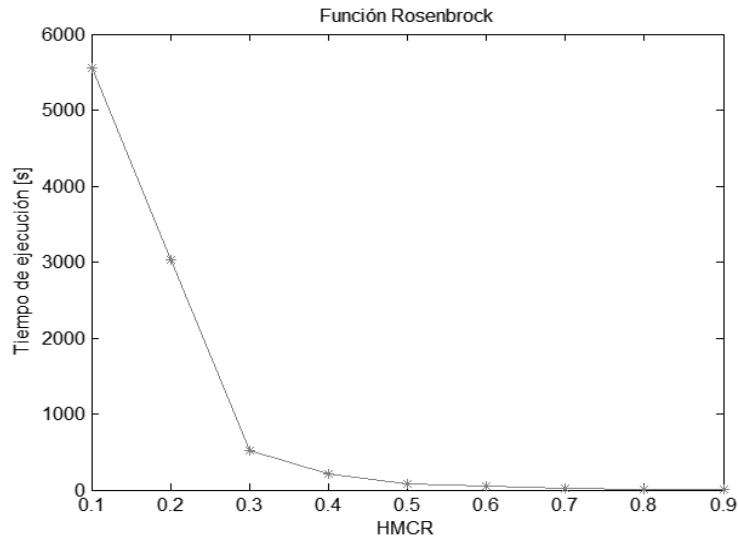
La ecuación para cuatro dimensiones está dada por:

$$f(x_1, x_2, x_3) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 100(x_3 - x_2^2)^2 + (1 - x_2)^2 + 100(x_4 - x_3^2)^2 + (1 - x_3)^2 \quad (10)$$

El mínimo global se encuentra  $f(x_1, x_2, x_3, x_4) = f(1,1,1,1) = 0$ .

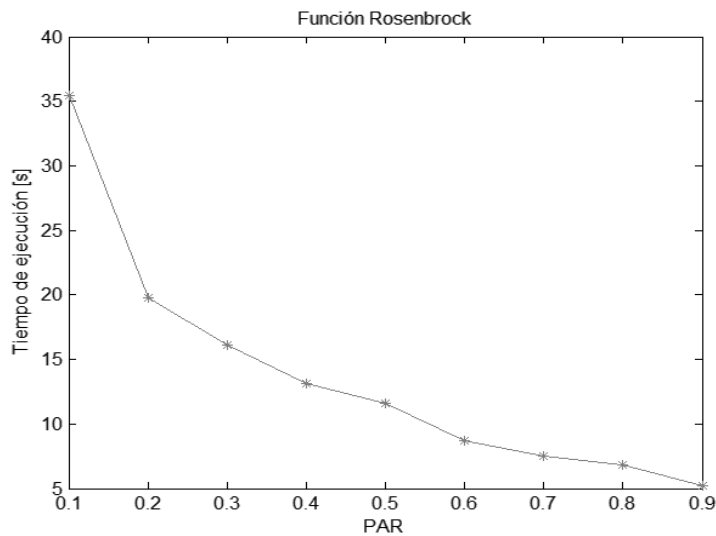
La Figura 18 y sintetiza los resultados obtenidos de las pruebas realizadas al algoritmo, donde se hace variar el parámetro *HMCR* en un rango de 0.1 a 0.9, dejando constante el valor del *PAR* (0.4).

**Figura 18. Variación del parámetro HMCR en la función Rosenbrock 4D. (Fuente: autores)**



La Figura 19 sintetiza los resultados obtenidos de las pruebas realizadas al algoritmo, donde se hace variar el parámetro *PAR* en un rango de 0.1 a 0.9, dejando constante el valor del *HMCR* (0.8).

**Figura 19. Variación del parámetro PAR en la función Rosenbrock 4D. (Fuente: autores)**



La Tabla 14 reporta los parámetros y la solución a la ecuación (9) del algoritmo de búsqueda armónica (HS). Se realizan tres ejecuciones del algoritmo, en cada una se modifica el parámetro *HMCR* (0.7, 0.8 y 0.9) con *PAR* (0.4) con el propósito de mostrar los tiempos de convergencia del algoritmo.

Los valores de los parámetros anteriormente mencionados son la mejor relación encontrada para el algoritmo. Esta afirmación está basada en los análisis hechos a las variaciones de los parámetros *HMCR* y *PAR*.

**Tabla 14. Parámetros y resultados del algoritmo con la función Rosenbrock 4D. (Fuente: autores)**

	<b>Ejecución 1</b>	<b>Ejecución 2</b>	<b>Ejecución 3</b>
<b>X1</b>	0.99780271229	0.99789171346	0.99776906132
<b>X2</b>	0.99578671460	0.99569717524	0.99581927702
<b>X3</b>	0.99174781966	0.99137544207	0.99172471951
<b>X4</b>	0.98343658060	0.98285721238	0.98348621455
<b>Error</b>	0.97862672305e-04	0.98406854817e-04	0.991354976305e-04
<b>Ut</b>	1252741	563178	382471
<b>XL1;XU1</b>	0; 2	0; 2	0; 2
<b>XL2;XU2</b>	0; 2	0; 2	0; 2
<b>XL3;XU3</b>	0; 2	0; 2	0; 2
<b>XL4;XU4</b>	0; 2	0; 2	0; 2
<b>HMS</b>	10	10	10
<b>HMCR</b>	0.7	0.8	0.9
<b>PAR</b>	0.4	0.4	0.4
<b>b</b>	0.001	0.001	0.001
<b>MaxImp</b>	1000000000	1000000000	1000000000
<b>Tol</b>	1e-4	1e-4	1e-4
<b>Te</b>	26.8212 [s]	13.1662 [s]	8.5281530 [s]

En el Anexo B se presenta la solución de la función Rosenbrock con diez, veinte y treinta variables.

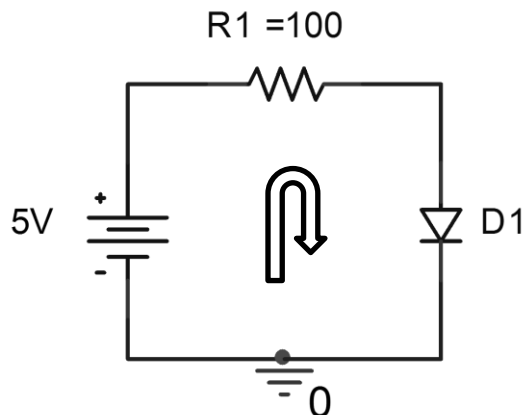
### 4.3 SOLUCIÓN DE CIRCUITOS ELECTRONICOS NO LINEALES DE C.C

Esta sección muestra el cumplimiento de los objetivos específicos 3 y 4 (ver página 15), por lo cual se da solución a diferentes circuitos electrónicos no lineales mediante el algoritmo de Búsqueda Armónica (HS). Cada una de las respuestas entregada por dicho algoritmo se contrasta con la simulación de la herramienta software *OrCAD Capture®* y con el método de optimización (Método de la espiral [2]).

#### 4.3.1 Circuito de corriente directa con un diodo

Se resuelve el circuito de corriente directa con un diodo de la Figura 20, con el fin de mostrar la capacidad del método de optimización de Búsqueda Armónica (HS).

Figura 20. Circuito de c.c con un diodo. (Fuente: autores)



Se procede a resolver el circuito, para esto se utiliza la ley de Voltajes de Kirchhoff, donde  $I$  es la corriente de malla.

$$-5 + 100I + V_D = 0 \quad (11)$$

Además se sabe que el modelo de la corriente de un diodo es el siguiente.

$$I_D = I_S(e^{\frac{V_D}{nV_T}} - 1) \quad (12)$$

Dónde:

$n : 1$ .

$I_S: 10x^{-12}$  [A].

$V_T = 25.85$  [mV].

*Los parámetros propios del diodo son los mismos dados para el trabajo de grado [2] para que la comparación de los algoritmos sea de la mejor forma posible.*

Remplazando la ecuación (12) en (11), para expresar todo en términos de una sola variable la expresión queda de la forma siguiente.

$$-5 + 100I_S \left( e^{\frac{V_D}{nV_T}} - 1 \right) + V_D = 0 \quad (13)$$

Empleando la ecuación (2), la función objetivo que resulta es la siguiente.

Donde  $V_D = X_1$  el voltaje del diodo.

$$f(X_1) = (-5 + 100I_S \left( e^{\frac{X_1}{nV_T}} - 1 \right) + X_1)^2 = 0 \quad (14)$$

El resultado entregado por el algoritmo es:

$$X_1 = V_D = 0.633321216756516[V]$$

La Tabla 15 muestra la comparación entre el resultado dado por el algoritmo de búsqueda armónica y la solución dada por *Orcad Capture*. Se observa que la respuesta que entrega el algoritmo es muy semejante a la entregada por la simulación, lo que indica que el método de Búsqueda Armónica resolvió satisfactoriamente el circuito.

**Tabla 15. Comparación de resultados entre el algoritmo de Búsqueda Armónica y la simulación para un circuito de c.c con un diodo. (Fuente: autores)**

Variable	Búsqueda Armónica, [V]	OrCAD Capture, [V]
X1	0.633321216756516	0.6334

Los parámetros utilizados para ejecutar el algoritmo de Búsqueda Armónica se muestran en la Tabla 16.

**Tabla 16. Parámetro de inicialización para la ejecución del algoritmo de Búsqueda Armónica para la solución de un circuito c.c con un diodo. (Fuente: autores)**

Parámetros	Valor
<i>XU1</i>	1
<i>XL1</i>	0
<i>HMS</i>	10
<i>HMCR</i>	0.8
<i>PAR</i>	0.4
<i>b</i>	0.0001
<i>MaxImp</i>	10000000
<i>Tol</i>	$1 \times 10^{-25}$

En la Tabla 17, se observa los resultados entregados por el algoritmo de Búsqueda Armónica y el método de la espiral [2], los resultados entregados son: la precisión de la respuesta, el error que se obtiene al evaluar la función objetivo en el resultado obtenido, la cantidad de iteraciones para llegar a la tolerancia deseada y el tiempo de ejecución.

**Tabla 17. Comparación de resultados entre los métodos de optimización de Búsqueda Armónica y la Espiral de un circuito de c.c con un diodo. (Fuente: autores)**

	Búsqueda Armónica	Método de la Espiral
<i>X1</i>	0.633321216756516	0.633321216757110
<i>Error</i>	$9.4392 \times 10^{-26}$	$9.9959 \times 10^{-26}$
<i>Ut</i>	487904	605
<i>Te</i>	220.0007 [s]	0.381762 [s]

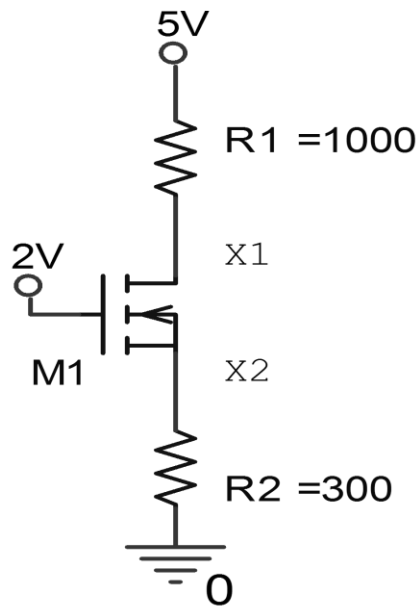
Se analiza la tabla anterior y se observa que el orden del valor del error es muy pequeño, lo que indica una alta precisión de la respuesta de ambos algoritmos. También se aprecia que aunque el método de Búsqueda armónica resolvió satisfactoriamente el circuito, este necesito más tiempo para obtener la respuesta,

a diferencia del método de la espiral que necesito menos tiempo para dar una respuesta semejante.

#### 4.3.2 Circuito de corriente directa con un MOSFET

Se resuelve el modelo matemático de un circuito de corriente directa con un MOSFET, con el fin de verificar la capacidad del algoritmo de Búsqueda Armónica para solucionar otros modelos de elementos no lineales. La Figura 21 muestra la configuración típica de un amplificador single con un MOSFET, la cual tiene modulación de canal N, sin efecto cuerpo ( $V_{SB} = 0$ ).

Figura 21. Circuito de amplificador Single con un MOSFET. (Fuente: autores)



Para este caso la tensión de drenador ( $V_D$ ) será la primera variable (X1) y la tensión de surtidor ( $V_S$ ) será la segunda variable (X2), lo cual permitirá escribir la corriente de MOSFET en términos de  $V_{GS}$  y  $V_{DS}$ . Para efectos prácticos de este trabajo de grado se consideró la región de saturación para dicho transistor.

Aplicando análisis nodal en las tensiones de drenador y de surtidor se obtienen las siguientes ecuaciones.

$$f_1 = \frac{1}{2} K_n \frac{W}{LZ} (2 - V_S - V_t)^2 (1 + \lambda(V_D - V_S)) - \frac{5 - V_D}{1000} = 0 \quad (15)$$

$$f_2 = \frac{1}{2} K_n \frac{W}{L} (2 - V_S - V_t)^2 (1 + \lambda(V_D - V_S)) - \frac{V_S}{300} = 0 \quad (16)$$

Dónde:

$$K_n: 2 \times 10^{-5}$$

$$\frac{W}{L}: \frac{20}{0.18}$$

$$V_t: 0.7$$

$$\lambda: 0.1$$

Los parámetros propios del transistor fueron tomados del trabajo de grado [2] para que la comparación de los algoritmos sea semejante.

Utilizando la ecuación (2), la función objetivo que usa el método de Búsqueda Armónica es la siguiente.

$$f(X_1, X_2) = \left( \frac{1}{2} K_n \frac{W}{L} (2 - X_2 - V_t)^2 (1 + \lambda(X_1 - X_2)) - \frac{5 - X_1}{1000} \right)^2 + \left( \frac{1}{2} K_n \frac{W}{L} (2 - X_2 - V_t)^2 (1 + \lambda(X_1 - X_2)) - \frac{X_2}{300} \right)^2 = 0 \quad (17)$$

El resultado entregado por el algoritmo de búsqueda armónica es:

$$X_1 = V_D = 3.738924120418978 [V], X_2 = V_S = 0.378322763932001 [V]$$

Se puede verificar en la Tabla 18 que el método de búsqueda armónica dio correcta solución al modelo matemático a prueba, al comparar los resultados con la simulación y ver que son semejantes.

**Tabla 18. Comparación de resultados entre el algoritmo de Búsqueda Armónica y la simulación de un amplificador Single con un MOSFET. (Fuente: autores)**

Variable	Búsqueda Armónica, [V]	OrCAD Capture, [V]
X1	3.738924120418978	3.739
X2	0.378322763932001	0.3783

Los parámetros utilizados para ejecutar el algoritmo de Búsqueda Armónica se muestran en la Tabla 19.

**Tabla 19. Parámetro de inicialización para la ejecución del algoritmo de Búsqueda Armónica de un amplificador Single con un MOSFET. (Fuente: autores)**

Parámetros	Valor
<i>XL1 - XU1</i>	0 - 5
<i>XL2 - XU2</i>	0 - 5
<i>HMS</i>	10
<i>HMCR</i>	0.8
<i>PAR</i>	0.4
<i>b</i>	0.00001
<i>MaxImp</i>	10000000
<i>Tol</i>	$1 \times 10^{-25}$

En la Tabla 20 se cuantifica el desempeño del algoritmo de búsqueda armónica y se compara con las respuestas entregadas por el método de la espiral.

**Tabla 20. Comparación de resultados entre el método de optimización de Búsqueda Armónica y el método de la Espiral de un amplificador Single con un MOSFET. (Fuente: autores)**

	Búsqueda Armónica	Método de la Espiral
<i>X1</i>	3.738924120418978[V]	3.738924114949117[V]
<i>X2</i>	0.378322763932001[V]	0.378322749299765[V]
<i>Error</i>	$1.9532 \times 10^{-26}$	$9.8791 \times 10^{-26}$
<i>Ut</i>	643157	440
<i>Te</i>	11.316086 [s]	9.668829 [s]



Para este modelo tendremos en cuenta 4 tensiones (X1, X2, X3 y X4), las cuales permiten expresar el voltaje de compuerta-surtidor y surtidor-drenador de todos los transistores, por lo tanto se puede conocer el valor de la corriente que pasa por cada MOSFET. Las ecuaciones de la Tabla 6 y Tabla 7 son empleadas para conocer el valor de las tensiones mencionadas anteriormente.

- Análisis de nodos en X1:

$$f_1 = \frac{1}{2}K_{n1}\frac{W}{L}(2 - X3 - V_{tn})^2(1 + \lambda_n(X1 - X3)) - \frac{1}{2}K_{p3}\frac{W}{L}(5 - X1 - V_{tp})^2(1 + \lambda_p(5 - X1)) = 0 \quad (18)$$

- Análisis de nodos en X2:

$$f_2 = \frac{1}{2}K_{n2}\frac{W}{L}(X4 - X3 - V_{tn})^2(1 + \lambda_n(X2 - X3)) - \frac{1}{2}K_{p4}\frac{W}{L}(5 - X1 - V_{tp})^2(1 + \lambda_p(5 - X2)) = 0 \quad (19)$$

- Análisis de nodos en X3:

$$f_3 = \frac{1}{2}K_{n1}\frac{W}{L}(2 - X3 - V_{tn})^2(1 + \lambda_n(X1 - X3)) + \frac{1}{2}K_{n2}\frac{W}{L}(X4 - X3 - V_{tn})^2(1 + \lambda_n(X2 - X3)) - 0.2X10^{-3} = 0 \quad (20)$$

- Análisis de nodos en X4:

$$f_4 = \frac{1}{2}K_{n5}\frac{W}{L}(X2 - X4 - V_{tn})^2(1 + \lambda_n(5 - X4)) - \frac{X4}{10X10^3} - 0.8X10^{-3} = 0 \quad (21)$$

Dónde:

$$K_n: 2x10^{-5}$$

$$V_{tn}: 0.7$$

$$\lambda_n: 0.1$$

$$K_p: 5x10^{-6}$$

$$V_{tp}: 0.8$$

$$\lambda_p: 0.2$$

$$\frac{W}{L} \cdot \frac{20}{0.18}$$

Los parámetros propios del transistor fueron tomados del trabajo de grado [2] para que la comparación entre los algoritmos sea de la mejor forma posible.

Utilizando la ecuación (2), la función objetivo resultante es la siguiente.

$$\begin{aligned}
 f(X_1, X_2, X_3, X_4) = & \\
 & \left( \frac{1}{2} K_{n1} \frac{W}{L} (2 - X_3 - V_{tn})^2 (1 + \lambda_n (X_1 - X_3)) \right. \\
 & \quad \left. - \frac{1}{2} K_{p3} \frac{W}{L} (5 - X_1 - V_{tp})^2 (1 + \lambda_p (5 - X_1)) \right)^2 \\
 & + \left( \frac{1}{2} K_{n2} \frac{W}{L} (X_4 - X_3 - V_{tn})^2 (1 + \lambda_n (X_2 - X_3)) \right. \\
 & \quad \left. - \frac{1}{2} K_{p4} \frac{W}{L} (5 - X_1 - V_{tp})^2 (1 + \lambda_p (5 - X_2)) \right)^2 \tag{22} \\
 & + \left( \frac{1}{2} K_{n1} \frac{W}{L} (2 - X_3 - V_{tn})^2 (1 + \lambda_n (X_1 - X_3)) \right. \\
 & \quad \left. + \frac{1}{2} K_{n2} \frac{W}{L} (X_4 - X_3 - V_{tn})^2 (1 + \lambda_n (X_2 - X_3)) - 0.2X10^{-3} \right)^2 \\
 & + \left( \frac{1}{2} K_{n5} \frac{W}{L} (X_2 - X_4 - V_{tn})^2 (1 + \lambda_n (5 - X_4)) - \frac{X_4}{10X10^3} - 0.8X10^{-3} \right)^2 = 0
 \end{aligned}$$

Los resultados entregados por el algoritmo son:

$$X_1 = 3.669543499350950[V], X_2 = 3.536556588979137 [V]$$

$$X_3 = 1.034500364935996 [V], X_4 = 2.004197311439884 [V]$$

En la Tabla 21 se observa que el método de búsqueda armónica dio una muy buena respuesta a la solución del modelo matemático del circuito comparándolo con la simulación de OrCAD capture.

**Tabla 21. Comparación de resultados entre el algoritmo de Búsqueda Armónica y la simulación de un convertidor de corriente a voltaje de c.c. (Fuente: autores)**

<b>Variable</b>	<b>Búsqueda Armónica, [V]</b>	<b>OrCAD Capture, [V]</b>
X1	3.669543499350950	3.670
X2	3.536556588979137	3.537
X3	1.034500364935996	1.035
X4	2.004197311439884	2.004

La Tabla 22 muestra los parámetros utilizados para ejecutar el algoritmo de Búsqueda Armónica.

**Tabla 22. Parámetro de inicialización para la ejecución del algoritmo de Búsqueda Armónica de un convertidor de corriente a voltaje de c.c. (Fuente: autores)**

<b>Parámetros</b>	<b>Valor</b>
<b><i>XU1</i></b>	4
<b><i>XL1</i></b>	3
<b><i>XU2</i></b>	4
<b><i>XL2</i></b>	3
<b><i>XU3</i></b>	2
<b><i>XL3</i></b>	0
<b><i>XU4</i></b>	2.5
<b><i>XL4</i></b>	1.5
<b><i>HMS</i></b>	10
<b><i>HMCR</i></b>	0.9
<b><i>PAR</i></b>	0.4
<b><i>b</i></b>	$1 \times 10^{-7}$
<b><i>MaxImp</i></b>	10000000
<b><i>Tol</i></b>	$1 \times 10^{-25}$

En la Tabla 23 se compara el desempeño del algoritmo de búsqueda armónica con las respuestas dadas por el método de la espiral.

**Tabla 23. Comparación de resultados entre los métodos de optimización de Búsqueda Armónica y la Espiral de un circuito convertidor de corriente a voltaje. (Fuente: autores)**

	<b>Búsqueda Armónica</b>	<b>Método de la Espiral</b>
<b>X1</b>	3.669543499350950 [V]	3.669543499212045 [V]
<b>X2</b>	3.536556588979137 [V]	3.536556589015261 [V]
<b>X3</b>	1.034500364935996 [V]	1.034500364872692 [V]
<b>X4</b>	2.004197311439884 [V]	2.004197311536540 [V]
<b>Error</b>	$9.8088 \times 10^{-26}$	$8.9285 \times 10^{-26}$
<b>Ut</b>	7427276	393
<b>Te</b>	167.34165 [s]	1279.886145 [s]

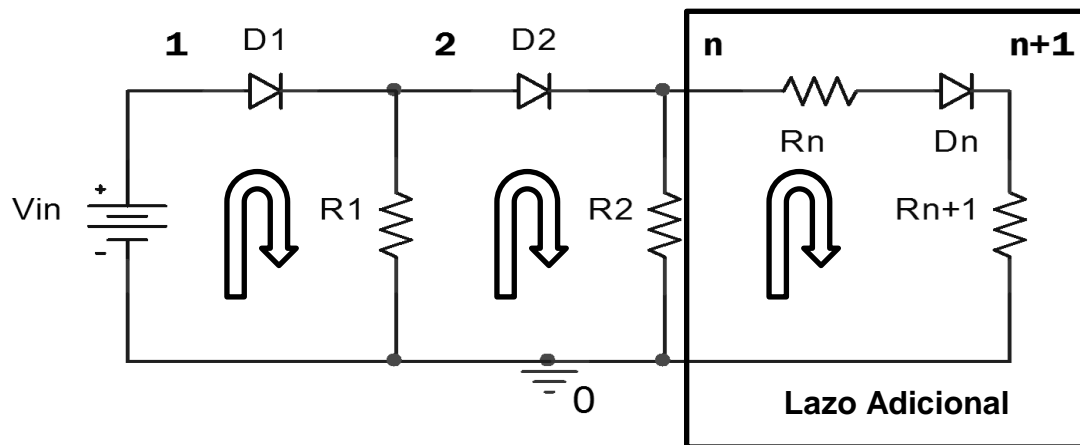
En la Tabla 23, se observa que la respuesta entregada por los dos algoritmos es muy precisa, esto se corrobora porque al remplazar esta en la función objetivo, se produce un error de orden muy pequeño. Comparando otro punto de este caso, se ve que el método de búsqueda armónica realizó un considerable número de iteraciones por segundo. Para terminar se resalta la gran diferencia entre los dos algoritmos, el método de la espiral incremento significativamente el tiempo de convergencia en relación al algoritmo de búsqueda armónica.

#### 4.4 Circuito de corriente directa con lazos adicionales en la carga.

La topología del circuito de la Figura 23 se presenta con el objetivo de verificar el rendimiento del algoritmo de búsqueda armónica (HS) de una forma dinámica sin tener que modificar la estructura del circuito como se ha realizado hasta ahora en cada uno de los ítems de la sección 3.3.

La sección 3.3, solo trata topologías de circuitos que pueden expresarse en sistemas de ecuaciones de una incógnita hasta cuatro incógnitas como máximo, cambiando la estructura y el modelo matemático de cada circuito, lo que resulta engorroso si se quiere resolver topologías de circuitos que tengan un número mayor de incógnitas. Por lo tanto para que la idea anteriormente mencionada se lleve a cabo, se ha creado un circuito base y se ha buscado la forma de agregar lazos al circuito base sin tener que cambiar la estructura del circuito electrónico en general.

Figura 23. Circuito de c.c con lazos adicionales en la carga. (Fuente: autores)



La solución de esta topología está fundamentada en la ley de voltajes de Kirchhoff, la expresión resultante son las corrientes del circuito base, como lo muestra la ecuación (23) y (24).

$$-V_{in} + n \cdot V_t \cdot \log\left(\frac{I_1}{I_S} + 1\right) + (I_1 - I_2) \cdot R_1 = 0 \quad (23)$$

$$I_2 \cdot R_2 + n \cdot V_t \cdot \log\left(\frac{I_2}{I_S} + 1\right) + (I_2 - I_1) \cdot R_1 = 0 \quad (24)$$

Con estas expresiones y utilizando la transformación del sistema de ecuaciones en un problema de optimización, se construye la función objetivo del sistema base.

$$f_{obj} = \{-Vin + n \cdot V_t \cdot \log\left(\frac{I_1}{I_S} + 1\right) + (I_1 - I_2) \cdot R_1\}^2 + \{I_2 \cdot R_2 + n \cdot V_t \cdot \log\left(\frac{I_2}{I_S} + 1\right) + (I_2 - I_1) \cdot R_1\}^2 = 0 \quad (25)$$

Ahora si se agrega lazos al circuito base es necesario ajustar el modelo matemático, esto se logra adicionando las siguientes ecuaciones.

$$f_n = f_n^{obj} - I_{n+1} \cdot R_2 = 0 \quad (26)$$

$$f_{n+1} = (I_{n+1} - I_n) \cdot R_2 + I_{n+1} \cdot (R_n + R_{n+1}) + n \cdot V_t \cdot \log\left(\frac{I_{n+1}}{I_S} + 1\right) = 0 \quad (27)$$

Lo que resulta en la función objetivo general del sistema.

$$F_{obj} = \{f_n\}^2 + \{f_{n+1}\}^2 = 0 \quad (28)$$

En los siguientes ítems se realiza una serie de ejecuciones modificando el número de lazos (0, 1, 8, 13 lazos adicionales). El modelo matemático inicial consta de un sistema de 2x2, al agregar un lazo su tamaño cambia a 3x3, hasta alcanzar un sistema de 15x15 con trece lazos.

Por cada modificación se realizan tres ejecuciones para la topología del circuito de la Figura 23. En cada ejecución se ajusta el parámetro *HCMR* (0.7, 0.8 y 0.9) y se deja constante el *PAR* (0.4), esto se realiza con el fin de observar el comportamiento del algoritmo, debido a que los valores del *HCMR* y *PAR* son la mejor la relación encontrada (Ver sección 3.2).

Los resultados obtenidos por el algoritmo de búsqueda armónica (HS) son comparados con la solución dada por el software comercial *OrCAD capture*.

#### 4.4.1 Circuito de corriente directa sin lazos adicionales

Para este primer circuito no se adicionan lazos ( $n=0$ ) a la topología base.

Se presenta los parámetros del circuito están dados en la Tabla 24.

**Tabla 24. Parámetros del circuito (n=0). (Fuente: autores)**

Parámetro	Valor
$V_{in}$	10 [V]
$R_1$	100 [ $\Omega$ ]
$R_2$	200 [ $\Omega$ ]
$n \cdot V_t$	0.026 [V]
$I_s$	$1 \times 10^{-12}$ [A]

La solución del circuito mediante el algoritmo de búsqueda armónica (HS) se presenta en la Tabla 25.

**Tabla 25. Resultados y parámetros del algoritmo (n=0). (Fuente: autores)**

	Ejecución 1	Ejecución 2	Ejecución 3
<b>X1</b>	0.13697465964 [A]	0.1368429320 [A]	0.1367243027 [A]
<b>X2</b>	0.04355583755 [A]	0.0434674510 [A]	0.0434594563 [A]
<b>Error</b>	6.9670227e-09	2.9826274e-09	2.2202061e-09
<b>Ut</b>	2676	3806	4773
<b>XL1;XU1</b>	0; 2	0; 2	0; 2
<b>XL2;XU2</b>	0; 2	0; 2	0; 2
<b>HMS</b>	10	10	10
<b>HMCR</b>	0.8	0.7	0.9
<b>PAR</b>	0.4	0.4	0.4
<b>MaxImp</b>	10000000	10000000	10000000
<b>Tol</b>	1e-8	1e-8	1e-8
<b>Te</b>	0.1440656 [s]	0.2080169 [s]	0.2460623 [s]

En la Tabla 26. Se presenta la comparación en los resultados obtenidos entre el algoritmo de búsqueda armónica (HS) y OrCAD capture.

Tabla 26. Comparación de resultados entre el algoritmo (HS) y OrCAD. (Fuente: autores)

Variable	Búsqueda Armónica, [A]	OrCAD Capture, [A]
<b>X1</b>	0.13697465964	0.1368
<b>X2</b>	0.04355583755	0.04349

#### 4.4.2 Circuito de corriente directa con un lazo adicional

Para este caso se adicionan un lazos ( $n=1$ ) a la topología base del circuito.

Se presenta los parámetros del circuito están dados en la Tabla 27.

Tabla 27. Parámetros del circuito ( $n=1$ ). (Fuente: autores)

Parámetro	Valor
$V_{in}$	10 [V]
$R_1$	100 [ $\Omega$ ]
$R_2$	200 [ $\Omega$ ]
$n \cdot V_t$	0.026 [V]
$I_s$	$1 \times 10^{-12}$ [A]
$R_n$	200 [ $\Omega$ ]
$R_{n+1}$	200 [ $\Omega$ ]

La solución del circuito mediante el algoritmo de búsqueda armónica (HS) se presenta en la Tabla 28.

Tabla 28. Resultados y parámetros del algoritmo ( $n=1$ ). (Fuente: autores)

	Ejecución 1	Ejecución 2	Ejecución 3
<b>X1</b>	0.1566683959 [A]	0.1566977964 [A]	0.1570755966 [A]
<b>X2</b>	0.0634487555 [A]	0.0634640925 [A]	0.0636880455 [A]
<b>X3</b>	0.0201097847 [A]	0.0201231390 [A]	0.0202038329 [A]
<b>Error</b>	9.30378111e-09	6.83794955e-09	7.2455333e-09
<b>Ut</b>	20919	1126	16461

<b><i>XL1;XU1</i></b>	0; 2	0; 2	0; 2
<b><i>XL2;XU2</i></b>	0; 2	0; 2	0; 2
<b><i>XL3;XU3</i></b>	0; 2	0; 2	0; 2
<b><i>HMS</i></b>	10	10	10
<b><i>HMCR</i></b>	0.8	0.7	0.9
<b><i>PAR</i></b>	0.4	0.4	0.4
<b><i>MaxImp</i></b>	10000000	10000000	10000000
<b><i>Tol</i></b>	1e-8	1e-8	1e-8
<b><i>Te</i></b>	1.1924378 [s]	0.07482551 [s]	0.9866840 [s]

En la Tabla 29. Se presenta la comparación en los resultados obtenidos entre el algoritmo de búsqueda armónica (HS) y OrCAD capture.

**Tabla 29. Comparación de resultados entre el algoritmo (HS) y OrCAD. (Fuente: autores)**

<b>Variable</b>	<b>Búsqueda Armónica, [A]</b>	<b>OrCAD Capture, [A]</b>
<b>X1</b>	0.1566977964	0.1568
<b>X2</b>	0.0634640925	0.06358
<b>X3</b>	0.0201231390	0.02018

#### 4.4.3 Circuito de corriente directa con ocho lazos adicionales

Para este caso se adicionan ocho lazos ( $n=8$ ) a la topología base del circuito.

Se presenta los parámetros del circuito están dados en la Tabla 30.

**Tabla 30. Parámetros del circuito ( $n=8$ ). (Fuente: autores)**

<b>Parámetro</b>	<b>Valor</b>
$V_{in}$	500 [V]
$R_1$	100 [ $\Omega$ ]
$R_2$	200 [ $\Omega$ ]
$n \cdot V_t$	0.026 [V]

$I_s$	$1 \times 10^{-12}$ [A]
$R_n$	200 [ $\Omega$ ]
$R_{n+1}$	200 [ $\Omega$ ]

La solución del circuito mediante el algoritmo de búsqueda armónica (HS) se presenta en la Tabla 31.

**Tabla 31. Resultados y parámetros del algoritmo (n=8). (Fuente: autores)**

	<b>Ejecución 1</b>	<b>Ejecución 2</b>	<b>Ejecución 3</b>
<b>X1</b>	9.0203607825 [A]	9.022670157 [A]	9.02240326088 [A]
<b>X2</b>	4.0284461155 [A]	4.0299923878 [A]	4.02980920752 [A]
<b>X3</b>	1.5364220922 [A]	1.5372067301 [A]	1.53716736877 [A]
<b>X4</b>	0.5846177626 [A]	0.5851026690 [A]	0.58512620233 [A]
<b>X5</b>	0.2211958918 [A]	0.2215182244 [A]	0.22153083782 [A]
<b>X6</b>	0.0824849002 [A]	0.0827116366 [A]	0.08275370322 [A]
<b>X7</b>	0.0296280758 [A]	0.0297851493 [A]	0.0298283362 2[A]
<b>X8</b>	0.0096403780 [A]	0.0097030519 [A]	0.00973932444 [A]
<b>X9</b>	0.0022600057 [A]	0.0023215286 [A]	0.00233063805 [A]
<b>X10</b>	0.0000132906 [A]	0.0000272488 [A]	0.00002630385 [A]
<b>Error</b>	9.402733e-06	9.6894325e-06	9.90518349e-06
<b>Ut</b>	6658784	4071960	900285
<b>XL1;XU1</b>	8; 10	8; 10	8; 10
<b>XL2;XU2</b>	3; 5	3; 5	3; 5
<b>XL3;XU3</b>	1; 2	1; 2	1; 2
<b>XL4;XU4</b>	0; 1	0; 1	0; 1
<b>XL5;XU5</b>	0; 0.5	0; 0.5	0; 0.5
<b>XL6;XU6</b>	0; 0.3	0; 0.3	0; 0.3
<b>XL7;XU7</b>	0; 0.2	0; 0.2	0; 0.2
<b>XL8;XU8</b>	0; 0.2	0; 0.2	0; 0.2
<b>XL9;XU9</b>	0; 0.2	0; 0.2	0; 0.2
<b>XL10;XU10</b>	0; 0.1	0; 0.1	0; 0.1

<b>HMS</b>	10	10	10
<b>HMCR</b>	0.8	0.7	0.9
<b>PAR</b>	0.4	0.4	0.4
<b>MaxImp</b>	100000000	100000000	100000000
<b>Tol</b>	1e-5	1e-5	1e-5
<b>Te</b>	546.626 [s]	285.31185 [s]	69.182620 [s]

En la Tabla 32. Se presenta la comparación en los resultados obtenidos entre el algoritmo de búsqueda armónica (HS) y OrCAD capture.

**Tabla 32. Comparación de resultados entre el algoritmo (HS) y OrCAD. (Fuente: autores)**

<b>Variable</b>	<b>Búsqueda Armónica, [A]</b>	<b>OrCAD Capture, [A]</b>
<b>X1</b>	9.0224032608	9.002
<b>X2</b>	4.0298092075	4.018
<b>X3</b>	1.5371673687	1.532
<b>X4</b>	0.5851262023	0.5831
<b>X5</b>	0.2215308378	0.2207
<b>X6</b>	0.0827537032	0.08234
<b>X7</b>	0.0298283362	0.02964
<b>X8</b>	0.0097393244	0.009667
<b>X9</b>	0.0023306380	0.002313
<b>X10</b>	0.0000263038	0.00002591

#### **4.4.4 Circuito de corriente directa con trece lazos adicionales**

Para este caso se adicionan ocho lazos ( $n=13$ ) a la topología base del circuito.

Se presenta los parámetros del circuito están dados en la Tabla 33.

**Tabla 33. Parámetros del circuito (n=13). (Fuente: autores)**

Parámetro	Valor
$V_{in}$	200000 [V]
$R_1$	100 [ $\Omega$ ]
$R_2$	200 [ $\Omega$ ]
$n \cdot V_t$	0.026 [V]
$I_s$	$1 \times 10^{-12}$ [A]
$R_n$	200 [ $\Omega$ ]
$R_{n+1}$	200 [ $\Omega$ ]

La solución del circuito mediante el algoritmo de búsqueda armónica (HS) se presenta en la Tabla 34.

**Tabla 34. Resultados y parámetros del algoritmo (n=13). (Fuente: autores)**

	Ejecución 1	Ejecución 2	Ejecución 3
<b>X1</b>	3617.9962112 [A]	3618.00828148 [A]	3618.0061494 [A]
<b>X2</b>	1618.0085278 [A]	1618.01678378 [A]	1618.0153105 [A]
<b>X3</b>	618.02086003 [A]	618.025011844 [A]	618.02429259 [A]
<b>X4</b>	236.05972134 [A]	236.062255677 [A]	236.06176582 [A]
<b>X5</b>	90.163773509 [A]	90.1655390022 [A]	90.165118795 [A]
<b>X6</b>	34.436665712 [A]	34.4380506776 [A]	34.437637497 [A]
<b>X7</b>	13.150956506 [A]	13.1520259320 [A]	13.151705110 [A]
<b>X8</b>	5.0206770606 [A]	5.02154245511 [A]	5.0212595471 [A]
<b>X9</b>	1.9153157871 [A]	1.91606919244 [A]	1.9157779300 [A]
<b>X10</b>	0.7293066449 [A]	0.72992450005 [A]	0.7296734353 [A]
<b>X11</b>	0.2763873430 [A]	0.27693592666 [A]	0.2766908671 [A]
<b>X12</b>	0.1035019909 [A]	0.10394324282 [A]	0.1037426312 [A]
<b>X13</b>	0.0375847393 [A]	0.03791731992 [A]	0.0377724088 [A]
<b>X14</b>	0.0125518359 [A]	0.01286127144 [A]	0.0126959679 [A]
<b>X15</b>	0.0032311048 [A]	0.00337946272 [A]	0.0032906047 [A]

<b>Error</b>	9.753358e-06	9.2881249e-06	9.5915886e-06
<b>Ut</b>	97254126	92824965	42846327
<b>XL1;XU1</b>	3500; 3700	3500; 3700	3500; 3700
<b>XL2;XU2</b>	1500; 1700	1500; 1700	1500; 1700
<b>XL3;XU3</b>	600; 650	600; 650	600; 650
<b>XL4;XU4</b>	200; 250	200; 250	200; 250
<b>XL5;XU5</b>	85; 95	85; 95	85; 95
<b>XL6;XU6</b>	30; 40	30; 40	30; 40
<b>XL7;XU7</b>	10; 15	10; 15	10; 15
<b>XL8;XU8</b>	4; 6	4; 6	4; 6
<b>XL9;XU9</b>	1; 2.5	1; 2.5	1; 2.5
<b>XL10;XU10</b>	0; 1	0; 1	0; 1
<b>XL11;XU11</b>	0; 0.5	0; 0.5	0; 0.5
<b>XL12;XU12</b>	0; 0.2	0; 0.2	0; 0.2
<b>XL13;XU13</b>	0; 0.1	0; 0.1	0; 0.1
<b>XL14;XU14</b>	0; 0.1	0; 0.1	0; 0.1
<b>XL15;XU15</b>	0; 0.1	0; 0.1	0; 0.1
<b>HMS</b>	10	10	10
<b>HMCR</b>	0.8	0.7	0.9
<b>PAR</b>	0.4	0.4	0.4
<b>MaxImp</b>	100000000	100000000	100000000
<b>Tol</b>	1e-5	1e-5	1e-5
<b>Te</b>	8016.352 [s]	7862.52 [s]	3553.517 [s]

En la Tabla 35. Se presenta la comparación en los resultados obtenidos entre el algoritmo de búsqueda armónica (HS) y OrCAD capture.

Tabla 35. Comparación de resultados entre el algoritmo (HS) y OrCAD. (Fuente: autores)

<b>Variable</b>	<b>Búsqueda Armónica, [A]</b>	<b>OrCAD Capture, [A]</b>
<b>X1</b>	3618.0061494	3610
<b>X2</b>	1618.0153105	1614
<b>X3</b>	618.02429259	616.2
<b>X4</b>	236.06176582	235.3
<b>X5</b>	90.165118795	89.86
<b>X6</b>	34.437637497	34.31
<b>X7</b>	13.151705110	13.10
<b>X8</b>	5.0212595471	5.001
<b>X9</b>	1.9157779300	1.908
<b>X10</b>	0.7296734353	0.7263
<b>X11</b>	0.2766908671	0.2753
<b>X12</b>	0.1037426312	0.1032
<b>X13</b>	0.0377724088	0.03756
<b>X14</b>	0.0126959679	0.01262
<b>X15</b>	0.0032906047	0.00327

## 5. OBSERVACIONES Y CONCLUSIONES

- Se cumplieron los objetivos propuestos en el plan de proyecto de grado.
- Se propuso una aplicación del algoritmo de búsqueda armónica (HS), hasta ahora no reportada en la literatura, consistente en determinar la solución numérica de modelos matemáticos (en forma de ecuaciones) no lineales que describen circuitos electrónicos de corriente continua.
- Para las aplicaciones del algoritmo HS en la solución del modelo, no es necesario delimitar rigurosamente la región factible, puesto que el algoritmo tiene la ventaja de no caer en mínimos locales.
- Se observa que la precisión de las respuestas del algoritmo de búsqueda armónica (HS) fundamentados en el análisis de resultados es insensible a los parámetros propios del método como son: *HMCR* y el *PAR*. A pesar de lo anterior, el correcto ajuste de estos parámetros tiene gran influencia en el tiempo de convergencia del algoritmo (HS).
- El método utilizado en este trabajo de grado modalidad investigación presenta una gran ventaja frente a otros métodos de optimización ya estudiados en nuestro grupo de investigación y relacionada con su tiempo de computación. Esto se debe al parecer porque el HS no realiza cálculos muy demandantes. El tiempo de convergencia del algoritmo marca la diferencia, por ejemplo, entre el método de búsqueda armónica y el método con el cual se comparó directamente (método de la espiral); esta diferencia se hace más relevante conforme se aumenta el nivel de complicación del circuito electrónico.
- El parámetro ancho de banda (*BW*) se debe ajustar de acuerdo con la exactitud que se requiera dar la respuesta. Ello se corrobora puesto que una elección inapropiada del *BW* puede dar errores en la respuesta tales como: no llegar a la tolerancia especificada, o simplemente dar una respuesta equivocada.

- Se concluye tomando en cuenta lo anterior, que el algoritmo de búsqueda armónica es un método recomendable para la solución de modelos matemáticos (sistema de ecuaciones) de circuitos electrónicos no lineales de corriente continua y de considerable tamaño (10 o más ecuaciones).

## 6. RECOMENDACIONES

- Se recomienda utilizar el método de búsqueda armónica como una nueva estrategia de solución de circuitos electrónicos no lineales, debido a la eficiencia que presenta en cada uno de los circuitos propuestos en la sección 3.3.

## BIBLIOGRAFIA

- [1] 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.
- [2] J. A. Portilla, "Solución de las Ecuaciones que Modelan un Circuito no Lineal de Corriente Directa Mediante el Método de la Espiral," Universidad Industrial de Santander, 2012.
- [3] R. Correa, I. Amaya, and J. Cruz, "Real Roots of Nonlinear Systems of Equations Through a Metaheuristic algorithm," *Revista Dyna*, vol. 78, pp. 15–23, 2011.
- [4] J. Gonzales and C. Bayona, "Aplicacion del metodo de optimizacion por fuerza central (CFO), Al diseño de un absorbedor electromagnetico optimo," Universidad Industrial de Santander, 2013.
- [5] J. Ramirez and F. Osorio, "Algoritmo de enjambre de particulas unificado para la solucion de ecuaciones diofánticas lineales comúnmente encontradas en problemas de ingeniería electronica," Universidad Industrial de Santander, 2013.
- [6] J. Contreras and C. Villanueva, "Solución de un Sistema de Ecuaciones no Lineales, Utilizando una Estrategia Basada en el Algoritmo de Harmony Search," Universidad Industrial de Santander, 2013.
- [7] L. A. Gomez, "Solución de Sistemas de Ecuaciones No-Lineales Mediante La Tecnica de Optimización Global Por Analisis de Intervalos," Universidad Industrial de Santander, 2012.
- [8] C. Cobos, J. Pérez, and D. Estupiñan, "Una revisión de la búsqueda armónica A survey of harmony search," *Revista Avances en Sistemas e Informatica*, vol. 8, no. 2, pp. 1–14, 2011.
- [9] G. V. Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search," *Simulation*, vol. 76, no. 2, pp. 60–68, Feb. 2001.
- [10] A. S. and K. Smith, *Circuitos Microelectronicos*, 4th ed. Mexico: Oxford University Press, 2002, pp. 25 –759.

- [11] B. Razavi, *Design of analog CMOS integrated circuits*, 1 st. Singapore: Mc Graw Hill Interntational Edition, 2001, pp. 9–37.
- [12] A. Gilat, *MATLAB Una Introduccion Con Ejemplos Practicos*, 2 nd. Hoboken (NJ): John Wiley & Sons, 2005, pp. 5–71.
- [13] P. Gray, *Analysis and design of analog integrated circuits*, 4 th. California: John Wiley & Sons, 2001, pp. 12–47.
- [14] R. Baldick, *Applied Optimization Formulation and Algorithms for Engineering Systems*, 1 st. New York: Cambridge University Press, 2006, pp. 334–340.
- [15] Z. W. Geem, *Music-Inspired Harmony Search Algorithm*, 1 th. Berlin: Springer, 2009, pp. 13–203.
- [16] C. Hong and G. Xing-sheng, “Multi-HM Adaptive Harmony Search Algorithm and its Application to Continuous Function Optimization,” *Research Journal of Applied Sciences*, vol. 4, no. 2, pp. 100–103, 2012.
- [17] A. R. Yıldız, “Hybrid Taguchi-Harmony Search Algorithm for Solving Engineering Optimization Problems,” *International Journal of Industrial*, vol. 15, no. 3, pp. 286–293, 2008.

## ANEXOS

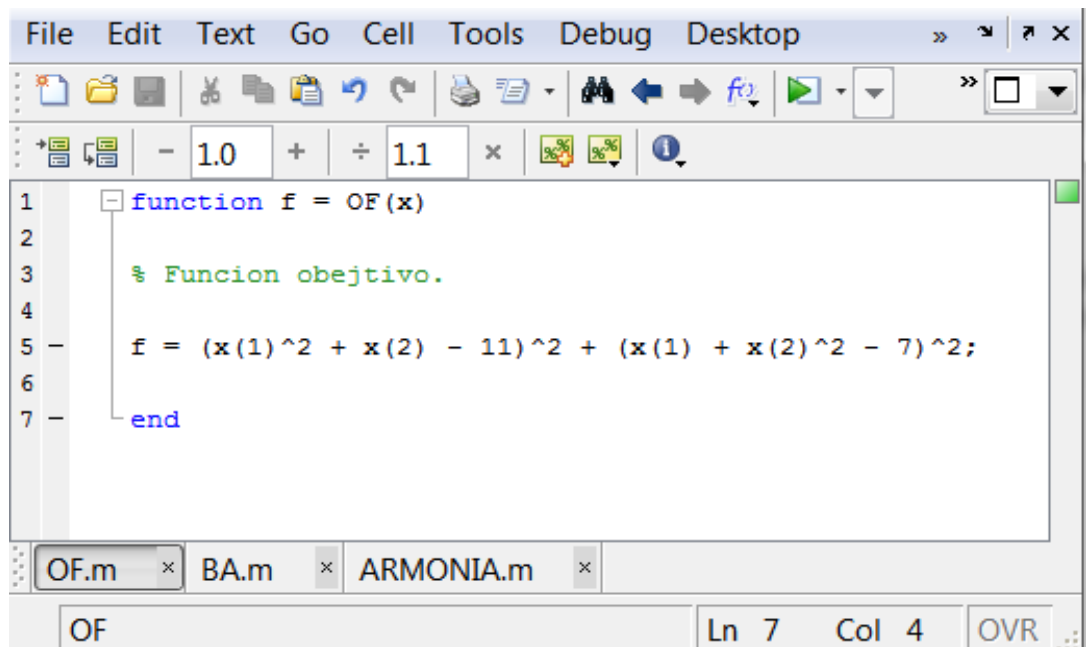
### ANEXO A. Tutorial para la ejecución del algoritmo de búsqueda armónica

Para ejecutar el algoritmo de búsqueda armónica se necesita de tres archivos BA.m, ARMONIA.m y OF.m, los cuales se encuentran en la carpeta llamada *Búsqueda armónica*. Los pasos para la ejecución son los siguientes:

#### Paso 1.

Se procede a abrir el archivo OF.m (función objetivo) en el cual se escribe el sistema de ecuaciones en forma de función objetivo, empleando la ecuación (2).

Figura 24. Script OF.m (Fuente: autores)



```
File Edit Text Go Cell Tools Debug Desktop
- 1.0 + ÷ 1.1 x % % !
1 function f = OF(x)
2
3 % Funcion oboectivo.
4
5 f = (x(1)^2 + x(2) - 11)^2 + (x(1) + x(2)^2 - 7)^2;
6
7 end
OF.m BA.m ARMONIA.m
OF Ln 7 Col 4 OVR
```

#### Paso 2.

Este paso se inicia al abrir el archivo BA.m, donde se coloca los valores de límites superior e inferior en los vectores  $XU$  y  $XL$  respectivamente como lo muestra la Figura 25.



#### Paso 4.

El algoritmo de búsqueda armónica finaliza y entrega los resultados en la ventana de comandos.

**Figura 27. Resultado final de la ejecución del algoritmo de búsqueda armónica. (Fuente: autores)**

```
File Edit Debug Parallel Desktop Window Help
C:\Users\OSCAR\Desktop\harmony search\C
Shortcuts How to Add What's New
Ultima iteración :
225
La Mejor solucion encontrada es:
3.583455309041295 -1.848541561027815
Valor de la funcion = 0.00005492128190
Tiempo de computo:
0.028124908997369
fx >>
Start OVR
```

## ANEXO B. Función Rosenbrock para diez, veinte y treinta variables

### *Función Rosenbrock para diez variables*

Se presenta en la Tabla 36, los resultados obtenidos de la ejecución del algoritmo de búsqueda armónica con la función Rosenbrock de diez variables.

**Tabla 36. Parámetros y resultados de la ejecución del algoritmo (HS) con la función Rosenbrock de diez variables. (Fuente: autores)**

Variable		Valor de la función	
<b>X1</b>	<b>X6</b>	1.00003948141	1.00108243082
<b>X2</b>	<b>X7</b>	1.00006641773	1.00215901393
<b>X3</b>	<b>X8</b>	1.00012774780	1.00431094419
<b>X4</b>	<b>X9</b>	1.00026249665	1.00865087648
<b>X5</b>	<b>X10</b>	1.00052387476	1.01741797089
Parámetro		Valor	
<b>XL1; XL2;... XL10</b>		0.5	
<b>XU1; XU2;... XU10</b>		1.5	
<b>Error</b>		9.996305617422070e-05	
<b>Ut</b>		897653	
<b>HMS</b>		10	
<b>HMCR</b>		0.9	
<b>PAR</b>		0.9	
<b>MaxImp</b>		1000000000	
<b>Tol</b>		1e-4	
<b>Te</b>		22.511 [s]	

### ***Función Rosenbrock para veinte variables***

Se presenta en la Tabla 37, los resultados obtenidos de la ejecución del algoritmo de búsqueda armónica con la función Rosenbrock de veinte variables.

**Tabla 37. Parámetros y resultados de la ejecución del algoritmo (HS) con la función Rosenbrock de veinte variables. (Fuente: autores)**

<b>Variable</b>		<b>Valor de la función</b>	
<b><i>X1</i></b>	<b><i>X11</i></b>	1.00000158138	1.00003796418
<b><i>X2</i></b>	<b><i>X12</i></b>	0.99999488800	1.00007204531
<b><i>X3</i></b>	<b><i>X13</i></b>	0.99999492238	1.00014317304
<b><i>X4</i></b>	<b><i>X14</i></b>	1.00000569812	1.00026543341
<b><i>X5</i></b>	<b><i>X15</i></b>	1.00000403084	1.00052347452
<b><i>X6</i></b>	<b><i>X16</i></b>	1.00000019689	1.00106239301
<b><i>X7</i></b>	<b><i>X17</i></b>	0.99999490230	1.00214263507
<b><i>X8</i></b>	<b><i>X18</i></b>	0.99998989990	1.00430153415
<b><i>X9</i></b>	<b><i>X19</i></b>	1.00000622283	1.00865419801
<b><i>X10</i></b>	<b><i>X20</i></b>	1.00001489946	1.01741318225
<b>Parámetro</b>		<b>Valor</b>	
<b><i>XL1; XL2;... XL20</i></b>		0.5	
<b><i>XU1; XU2;... XU20</i></b>		1.5	
<b><i>Error</i></b>		9.99277231318e-05	
<b><i>Ut</i></b>		6015765	
<b><i>HMS</i></b>		10	
<b><i>HMCR</i></b>		0.9	
<b><i>PAR</i></b>		0.9	
<b><i>MaxImp</i></b>		1000000000	
<b><i>Tol</i></b>		1e-4	
<b><i>Te</i></b>		199.736 [s]	

### ***Función Rosenbrock para treinta variables***

Se presenta en la Tabla 38, los resultados obtenidos de la ejecución del algoritmo de búsqueda armónica con la función Rosenbrock de treinta variables.

**Tabla 38. Parámetros y resultados de la ejecución del algoritmo (HS) con la función Rosenbrock de treinta variables. (Fuente: autores)**

<b>Variable</b>		<b>Valor de la función</b>	
<b><i>X1</i></b>	<b><i>X16</i></b>	0.99994202514	0.99981862756
<b><i>X2</i></b>	<b><i>X17</i></b>	0.99990743537	0.99983938347
<b><i>X3</i></b>	<b><i>X18</i></b>	0.99989781862	0.99985256877
<b><i>X4</i></b>	<b><i>X19</i></b>	0.99987164888	0.99986548623
<b><i>X5</i></b>	<b><i>X20</i></b>	0.99985651881	0.99985650232
<b><i>X6</i></b>	<b><i>X21</i></b>	0.99984790328	0.99986104918
<b><i>X7</i></b>	<b><i>X22</i></b>	0.99985402034	0.99984918099
<b><i>X8</i></b>	<b><i>X23</i></b>	0.99984646464	0.99985587318
<b><i>X9</i></b>	<b><i>X24</i></b>	0.99980286080	0.99986676539
<b><i>X10</i></b>	<b><i>X25</i></b>	0.99974092047	0.99989837167
<b><i>X11</i></b>	<b><i>X26</i></b>	0.99967395567	0.99995617545
<b><i>X12</i></b>	<b><i>X27</i></b>	0.99962729063	1.00005340606
<b><i>X13</i></b>	<b><i>X28</i></b>	0.99968302599	1.00018551252
<b><i>X14</i></b>	<b><i>X29</i></b>	0.99974496136	1.00042476292
<b><i>X15</i></b>	<b><i>X30</i></b>	0.99979243862	1.00088471103
<b>Parámetro</b>		<b>Valor</b>	
<b><i>XL1; XL2;... XL30</i></b>		0.5	
<b><i>XU1; XU2;... XU30</i></b>		1.5	
<b><i>Error</i></b>		9.99972529835e-05	
<b><i>Ut</i></b>		1465410	
<b><i>HMS</i></b>		10	
<b><i>HMCR</i></b>		0.9	
<b><i>PAR</i></b>		0.9	
<b><i>MaxImp</i></b>		1000000000	

<b><i>Tol</i></b>	1e-4
<b><i>Te</i></b>	57.91790 [s]