

ALGORITMOS BIO-INSPIRADOS, TEXTURAS Y SU APLICACIÓN A LA FÍSICA DE ALTAS ENERGÍAS

Tesis de maestría

Autor:

Martínez Guerrero Emmanuel

Supervisada por:

Dr. Noriega Papaqui Roberto

Dr. A. Miranda Romagnoli Pedro

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO
ÁREA ACADÉMICA DE COMPUTACIÓN Y ELECTRÓNICA



Universidad Autónoma del Estado de Hidalgo

Instituto de Ciencias Básicas e Ingeniería

School of Engineering and Basic Sciences

Área Académica de Computación y Electrónica

Department of Electronics and Computer Science

Mineral de la Reforma, Hgo., 21 de junio del 2024

Número de control: ICBI-AACyE/1017/2024

Asunto: Autorización de impresión de tema de tesis.

MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO
DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH

El Comité Tutorial de la TESIS del Programa Educativo de Posgrado titulado "Algoritmos bio-inspirados, texturas y su aplicación a la física de altas energías", realizado por el sustentante **Martínez Guerrero Emmanuel**, con número de cuenta 278938, perteneciente al Programa de la Maestría en Ciencias en Computación Avanzada y Electrónica, una vez que se ha revisado, analizado y evaluado el documento recepcional de acuerdo a lo estipulado en el Artículo 110 del Reglamento de Estudios de Posgrado, tiene a bien extender la presente:

AUTORIZACIÓN DE IMPRESIÓN

Por lo que el sustentante deberá cumplir con los requisitos del Reglamento de Estudios de Posgrado y con lo establecido en el proceso de grado vigente.

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO



Instituto de Ciencias Básicas e Ingeniería
Área Académica de Computación y Electrónica

Dr. Roberto Noriega Papaqui
Director de Tesis

Atentamente
"Amor. Orden y Progreso"

Dr. Pedro A. Miranda Romagnoli
Director de Tesis

Comité Tutorial

Dr. Omar López Ortega	Presidente	UAEH	
Dr. Esteban Rueda Soriano	Secretario	UAEH	
Dr. Roberto Noriega Papaqui	Vocal	UAEH	
Dr. Pedro A. Miranda Romagnoli	Suplente	UAEH	

OPC/APL



Ciudad del Conocimiento, Carretera Pachuca-Tulancingo Km. 4.5 Colonia Carboneras, Mineral de la Reforma, Hidalgo, México. C.P. 42184
Teléfono: 52 (771) 71 720 00 Ext. 40052, 40053
aacye_icbi@uaeh.edu.mx, jesus_ordaz@uaeh.edu.mx



uaeh.edu.mx

Resumen

En distintos campos del conocimiento, existen problemas de optimización complejos que representan una oportunidad para ser abordados mediante herramientas computacionales avanzadas. Un problema dentro de la fenomenología de la Física de Altas Energías (FAE), es la validación de modelos teóricos de matrices de masa con ceros de textura. Estos modelos, relevantes para entender la estructura e interacciones de partículas elementales como los quarks, pueden analizarse numéricamente utilizando el método de ajuste de Chi-cuadrado (χ^2) para evaluar su viabilidad según datos experimentales actuales. Tradicionalmente, se han empleado métodos exhaustivos y técnicas exactas de optimización para esta tarea. Sin embargo, debido a la estructura matemática de la función χ^2 , es necesario recurrir a otras técnicas de optimización poco utilizadas en la FAE, para explorar ampliamente el espacio de búsqueda de la función χ^2 y encontrar los parámetros óptimos del modelo que mejor se ajusten a los datos experimentales. En esta tesis, se propone el uso de Algoritmos Bio-Inspirados (ABI) para optimizar la función χ^2 en el análisis numérico del modelo de matrices de masa con cuatro ceros de textura. Se presenta una nueva variante del Algoritmo de Evolución Diferencial basado en Optimización por Enjambre de Partículas con mutación auto-adaptativa (DEPSO) llamada Algoritmo de Evolución Diferencial basado en Optimización por Enjambre de Partículas con Información Élite Histórica (HE-DEPSO), que utiliza una estrategia de mutación basada en información histórica de los mejores individuos e implementa la adaptación de parámetros del algoritmo “*Success-History based Adaptive DE*” (SHADE). Los resultados obtenidos al comparar el rendimiento de HE-DEPSO con otros algoritmos, en la optimización de la función χ^2 y en el conjunto de funciones de prueba “*Congress on Evolutionary Computation*” (CEC) 2017, muestran buenos resultados en precisión y estabilidad. Al implementar HE-DEPSO en el análisis numérico del modelo de texturas con cuatro ceros, se pudo comprobar la viabilidad del modelo y analizar sus implicaciones fenomenológicas.

Abstract

In several fields of knowledge, there are complex optimization problems that represent an opportunity to be addressed using advanced computational tools. One problem within the phenomenology of the High Energy Physics (HEP) is the validation of theoretical mass matrix models with texture zeros. These models, relevant for understanding the structure and interactions of elementary particles such as quarks, can be analyzed numerically using the Chi-square (χ^2) fitting method to evaluate their viability based on current experimental data. Traditionally, exhaustive methods and exact optimization techniques have been used for this task. However, due to the mathematical structure of the χ^2 function, it is necessary to resort to other optimization techniques that are not widely used in the HEP, to explore the search space of the χ^2 function extensively and find the optimal model parameters that best fit the experimental data. In this thesis, the use of ABI is proposed to optimize the χ^2 function in the numerical analysis of the four texture zeros mass matrix model. A new variant of DEPSO called HE-DEPSO is presented, which uses a mutation strategy based on historical information of the best individuals and implements the parameter adaptation of the SHADE algorithm. The results obtained when comparing the performance of HE-DEPSO with other algorithms, in the optimization of the χ^2 function and in the CEC 2017 test function set, show good results in accuracy and stability. By implementing HE-DEPSO in the numerical analysis of the four texture zeros model, it was possible to verify the viability of the model and analyze its phenomenological implications.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas e instituciones que me han apoyado durante la realización de esta tesis. Todas ellas merecen que, de alguna manera, su apoyo y ayuda brindada quede reflejada en este trabajo.

En primer lugar, he de agradecer a mis directores de tesis. El Dr. Pedro A. Miranda Romagnoli, por su constante orientación y apoyo a lo largo de todo el proceso y al Dr. Roberto Noriega Papaqui, por aceptarme como estudiante de maestría.

Además de mis directores, también debo agradecer al Dr. Omar López Ortega y al Dr. Esteban Rueda Soriano por formar parte de mi comité evaluador. Sus observaciones y recomendaciones han sido de gran valor para mejorar la calidad de este trabajo.

Extiendo mi gratitud a la Dra. Anilu Franco Arcega por sus valiosos comentarios y sugerencias durante la redacción de esta tesis, así como al Dr. Joel Suárez Cansino y Dr. Pedro Lagos Eulogio por haberme compartido su conocimiento.

Aprovecho para agradecer a todos los profesores, compañeros y compañeras, y personas que conocí durante este tiempo, con las cuales compartí experiencias, conocimientos o simplemente tuve una buena charla.

Estoy profundamente agradecido con mi familia, especialmente con mi madre, por su apoyo inquebrantable y su motivación continua, los cuales han sido fundamentales para alcanzar mis metas.

Finalmente quiero agradecer a la Universidad Autónoma del Estado de Hidalgo (UAEH) por todas las facilidades que me otorgaron durante mis estudios de maestría. De la misma manera, agradezco al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) por el apoyo económico otorgado a través de la beca para estudios de maestría.

Contenido

Resumen	I
Abstract	III
Agradecimientos	V
Índice de Figuras	XI
Índice de Tablas	XV
Lista de Acrónimos	XVIII
1 Introducción	1
1.1 Planteamiento del Problema	2
1.2 Justificación	3
1.3 Hipótesis	4
1.4 Objetivo General	5
1.4.1 Objetivos Específicos	5
1.5 Metodología	5
1.6 Organización de la Tesis	7
2 Marco Teórico	9
2.1 Optimización	9
2.1.1 Problemas de Optimización Global	9
2.1.2 Algoritmos de Optimización	16
2.1.2.1 Algoritmos de Optimización Exactos	16
2.1.2.2 Algoritmos de Optimización Aproximados	18
2.1.2.3 The No Free Lunch Theorem	18
2.1.3 Metaheurísticas	19
2.2 Algoritmos Bio-Inspirados	22
2.2.1 Algoritmos Evolutivos	23

2.2.1.1	Representación de soluciones	24
2.2.1.2	Inicialización de soluciones	28
2.2.1.3	Selección	29
2.2.1.4	Cruce	31
2.2.1.5	Mutación	34
2.2.1.6	Reemplazo	36
2.2.1.7	Criterio de parada	38
2.2.2	Evolución Diferencial	39
2.2.2.1	Mutación	40
2.2.2.2	Cruce	40
2.2.2.3	Reemplazo	41
2.2.2.4	Variantes del Algoritmo de Evolución Diferencial	42
2.2.3	Algoritmos Basados en Inteligencia de Enjambre	44
2.2.4	Optimización por Enjambre de Partículas	45
2.2.4.1	Configuración de parámetros	49
2.2.4.2	Estructura de red social y variaciones	49
2.3	El Modelo Estándar	52
2.3.1	El problema del sabor	54
2.3.2	Sector electrodébil	55
2.3.2.1	Sector de Yukawa	56
2.3.2.2	Sector de quarks de Yukawa	56
2.3.3	La matriz de mezcla V_{ckm}	61
2.3.3.1	Parametrización Chau-Keng	61
2.3.3.2	Valores experimentales	63
2.4	Formalismo de texturas	63
2.4.1	Modelos de matrices de masa tipo texturas	64
2.4.2	Viabilidad de los modelos de texturas	65
3	Estado del Arte	67
3.1	Análisis numérico de la estructura tipo texturas de cuatro ceros	67
3.2	Algoritmos bio-inspirados dentro de la física de altas energías	70
3.3	Mejoras en el algoritmo de Evolución Diferencial	72
3.4	Discusión	77
4	Algoritmo propuesto HE-DEPSO	81
4.1	Motivación	81

4.2	Operador de mutación híbrido adaptativo	81
4.2.1	Estrategia “DE/current-to-EHE/1”	82
4.2.2	Selección de estrategia de mutación para el operador de mutación híbrido adaptativo	85
4.3	El algoritmo HE-DEPSO completo	86
4.4	Resultados experimentales y análisis	87
4.4.1	Funciones de prueba	88
4.4.2	Algoritmos y configuración de parámetros	88
4.4.3	Comparación con DE, PSO y tres variantes avanzadas del algoritmo DE	90
4.4.3.1	Resultados de optimización	98
4.4.3.2	Propiedades de convergencia	99
4.4.4	Discusión	102
5	Análisis numérico del modelo de texturas de cuatro ceros con HE- DEPSO	103
5.1	Modelo de texturas de cuatro ceros	103
5.2	Análisis numérico del modelo de texturas de cuatro ceros	110
5.2.1	Análisis de χ^2	110
5.2.2	Optimización de χ^2 con HE-DEPSO	112
5.2.3	Resultados y discusión	118
5.2.3.1	Regiones permitidas para A_u, A_d, ϕ_1 y ϕ_2	119
5.2.3.2	Predicciones de los elementos de la matriz V_{CKM}	124
5.2.3.3	Espacio completo de parámetros de las matrices de masa de quarks	127
6	Conclusiones y trabajo futuro.	133
6.1	Aportaciones	135
6.2	Trabajo futuro	135
	Referencias	148
	Apéndices	
A	Funciones de prueba CEC 2017	149
A.1	Funciones Básicas	149
A.2	Funciones unimodales	153
A.3	Funciones multimodales	155

A.4	Funciones híbridas	159
A.5	Funciones de composición	163
B	Curvas de convergencia para el conjunto de funciones de prueba CEC 2017	167
C	Método de ajuste χ^2	171
D	Rangos de valores de los parámetros A_u y A_d	173
E	Detalles en el criterio χ_{red}^2 utilizado	177

Índice de Figuras

1.5.1	Metodología propuesta.	6
2.1.1	Restricciones en un Problema de Optimización Global considerando un espacio de búsqueda de dos dimensiones ($n = 2$).	11
2.1.2	Tipos de problemas de optimización global de acuerdo a su dominio.	12
2.1.3	Soluciones x_1 y x_2 (puntos color rojo y azul) para la función unidimensional f y su correspondiente vecindario (S_1 y S_2 respectivamente).	13
2.1.4	Óptimos locales y globales en una función bidimensional f	14
2.1.5	Función objetivo unimodal (Función Esférica) y sus respectivas curvas de nivel.	15
2.1.6	Función objetivo multimodal (Función de Griewank) y sus respectivas curvas de nivel.	15
2.1.7	Problemas de optimización global con uno y múltiples óptimos globales.	16
2.1.8	Mecanismo general de un algoritmo metaheurístico para la búsqueda de una solución óptima.	20
2.1.9	Clasificación de algunas de las metaheurísticas más conocidas (Adaptado y traducido al español de Behesht et al. [1]).	21
2.2.1	Clasificación de los Algoritmos Bio-Inspirados.	22
2.2.2	Espacio fenotipo y genotipo para la representación de una solución.	25
2.2.3	Representación binaria: fenotipo y genotipo.	26
2.2.4	Representación entera: fenotipo y genotipo.	27
2.2.5	Representación de punto flotante: fenotipo y genotipo.	28
2.2.6	Selección proporcional al ajuste en donde cada sección de la ruleta corresponde a la probabilidad de selección de cada individuo de la población. El individuo x_1 es seleccionado.	30
2.2.7	Ejemplos de operadores de cruce provenientes de la representación binaria y adoptados por la representación real.	33

2.2.8	Ejemplo de cruce uniforme.	33
2.2.9	Ejemplos de cruce aritmético: simple, de una sola posición y completo.	34
2.2.10	Procedimiento para obtener el vector mutado $v_{i,G}$ a través del operador de mutación.	41
2.2.11	Esquema del proceso de actualización de la posición de una partícula en dos dimensiones.	48
2.3.1	Tabla de partículas elementales del modelo estándar.	53
4.2.1	Curva de selección de estrategia.	86
4.4.1	Curvas de convergencia de la medida de error de solución para las funciones F_1, F_3, F_5 y F_8 , con $D = 10, 30$. El eje horizontal y vertical representan la iteraciones y los valores de media de error para las 31 repeticiones independientes.	100
4.4.2	Curvas de convergencia de la medida de error de solución para las funciones F_1, F_3, F_5 y F_8 , con $D = 10, 30$. El eje horizontal y vertical representan la iteraciones y los valores de media de error para las 31 repeticiones independientes.	101
5.2.1	Proyecciones de la función χ^2 sobre las variables A_u, A_d, ϕ_1, ϕ_2	113
5.2.2	Curvas de convergencia de la medida de error de solución para la función χ^2 . El eje horizontal y vertical representan la iteraciones y los valores de media de error para las 31 repeticiones independientes.	117
5.2.3	Diagramas de caja de los mejores valores de ajuste global obtenidos por HE-DEPSO y los algoritmos DEPSO, SHADE, CoDE, DE y PSO en las 31 repeticiones independientes en la función χ^2 . En eje horizontal se tienen a los algoritmos a comparar y en el eje vertical valores de ajuste global.	118
5.2.4	Regiones permitidas para A_u/m_t y A_d/m_b restringidas según datos experimentales actuales con diferentes niveles de precisión: $\chi_{Red}^2 < 1$ (color negro), $\chi_{Red}^2 < 1 \times 10^{-1}$ (color azul) y $\chi_{Red}^2 < 1 \times 10^{-2}$ (color naranja), en cada uno de los cuatro casos de estudio.	120
5.2.5	Regiones permitidas para $\phi_1/\pi, \phi_2/\pi$ restringidas según datos experimentales actuales con diferentes niveles de precisión: $\chi_{Res}^2 < 1$ (color negro), $\chi_{Red}^2 < 1 \times 10^{-1}$ (color azul) y $\chi_{Red}^2 < 1 \times 10^{-2}$ (color naranja), en cada uno de los cuatro casos de estudio.	122

5.2.6	Predicción de $ V_{cd} $ y $ V_{ud} $ en cada uno de los cuatro casos de estudio.	126
5.2.7	Predicción de $ V_{cs} $ y $ V_{tb} $ en cada uno de los cuatro casos de estudio.	127
5.2.8	Predicción de $ V_{td} $ y $ V_{ts} $ en cada uno de los cuatro casos de estudio.	128
5.2.9	Regiones permitidas para $ B_u /A_u$ y $ B_d /A_d$ restringidas según datos experimentales actuales con diferentes niveles de precisión: $\chi_{Red}^2 < 1$ (color negro), $\chi_{Red}^2 < 1 \times 10^{-1}$ (color azul) y $\chi_{Red}^2 < 1 \times 10^{-2}$ (color naranja), en cada uno de los cuatro casos de estudio.	129
5.2.10	Regiones permitidas para C_u/A_u y C_d/A_d restringidas según datos experimentales actuales con diferentes niveles de precisión: $\chi_{Red}^2 < 1$ (color negro), $\chi_{Red}^2 < 1 \times 10^{-1}$ (color azul) y $\chi_{Red}^2 < 1 \times 10^{-2}$ (color naranja), en cada uno de los cuatro casos de estudio.	130
5.2.11	Regiones permitidas para $ D_u /A_u$ y $ D_d /A_d$ restringidas según datos experimentales actuales con diferentes niveles de precisión: $\chi_{Red}^2 < 1$ (color negro), $\chi_{Red}^2 < 1 \times 10^{-1}$ (color azul) y $\chi_{Red}^2 < 1 \times 10^{-2}$ (color naranja), en cada uno de los cuatro casos de estudio.	131
A.2.1	Funciones de prueba unimodales y sus respectivas curvas de nivel ($D = 2$).	154
A.3.1	Funciones de prueba multimodales y sus respectivas curvas de nivel ($D = 2$).	158
A.4.1	Funciones de prueba híbridas y sus respectivas curvas de nivel ($D = 2$).	162
A.5.1	Funciones de prueba compuestas y sus respectivas curvas de nivel ($D = 2$).	166
B.1	Curvas de convergencia de la medida de error de solución para el conjunto CEC 2017 con ($D = 10$). El eje horizontal y vertical representan la iteraciones y los valores de media de error para las 31 repeticiones independientes.	168
B.2	Curvas de convergencia de la medida de error de solución para el conjunto CEC 2017 con ($D = 30$). El eje horizontal y vertical representan la iteraciones y los valores de media de error para las 31 repeticiones independientes.	169

E.1	Regiones permitidas para A_u/m_t , A_d/m_b , ϕ_1/π y ϕ_2/π en el caso de estudio I, mostrando soluciones válidas según restricciones de χ_{red}^2 por término (puntos verdes) y soluciones válidas según restricción general de χ_{red}^2 (puntos púrpura).	179
E.2	Regiones permitidas para A_u/m_t , A_d/m_b , ϕ_1/π y ϕ_2/π en el caso de estudio I, restringidas según datos experimentales actuales con diferentes niveles de precisión ($\chi_{Red}^2 < 1$, $\chi_{Red}^2 < 1 \times 10^{-1}$ y $\chi_{Red}^2 < 1 \times 10^{-2}$), generadas utilizando las soluciones válidas bajo una restricción de χ_{red}^2 por término.	180
E.3	Predicciones para los elementos $ V_{cd} $, $ V_{ud} $, $ V_{cs} $, $ V_{tb} $, $ V_{td} $ y $ V_{ts} $ en el caso de estudio I, generadas utilizando las soluciones válidas bajo una restricción de χ_{red}^2 por término.	181

Índice de Tablas

2.2.1	Codificación binaria y Gray para un valor decimal.	26
2.3.1	Propiedades de los Quarks.	52
2.3.2	Propiedades de los Leptones.	53
2.3.3	Propiedades de los Bosones.	54
3.1.1	Trabajos relacionados al análisis de la textura de cuatro ceros del estado del arte y algunas de sus características.	70
3.2.1	Trabajos relacionados a las técnicas de optimización bio-inspiradas y sus aplicaciones dentro del área de física de altas energías.	72
3.3.1	Resumen de los trabajos más importantes del estado del arte relacionados con la mejora en el rendimiento del algoritmo DE.	78
4.4.1	Funciones de prueba CEC 2017.	89
4.4.2	Configuración de parámetros de todos los algoritmos.	90
4.4.3	Comparación de resultados entre HE-DEPSO, DE, PSO y tres variantes avanzadas del algoritmo DE, sobre el conjunto de funciones de prueba CEC 2017 con $D = 10$	92
4.4.4	Comparación de resultados entre HE-DEPSO, DE, PSO y tres variantes avanzadas del algoritmo DE, sobre el conjunto de funciones de prueba CEC 2017 con $D = 30$	95
5.2.1	Diferentes casos de estudio.	115
5.2.2	Espacios de búsqueda conforme a las posibles combinaciones de η_u y η_d	115
5.2.3	Comparación de resultados entre HE-DEPSO, DE, PSO y tres variantes avanzadas del algoritmo DE, sobre la función χ^2	116
5.2.4	Correlaciones entre R , ϕ_1 y ϕ_2 en los cuatro casos de estudio.	124

E.1	Porcentaje de las soluciones encontradas para los parámetros A_u , A_d , ϕ_1 y ϕ_2 por caso de estudio y precisión de χ_{red}^2 , en las cuales cada una de las contribuciones de χ_{12}^2 , χ_{13}^2 , χ_{23}^2 y χ_{JJ}^2 satisfacen ser menores que 1.	179
-----	--	-----

Lista de Acrónimos

ABI	Algoritmos Bio-Inspirados	I
ACO	<i>“Ant Colony Optimization”</i>	73
AE	Algoritmos Evolutivos	3
ANN	<i>“Artificial Neural Networks”</i>	73
CE	Computo Evolutivo	23
CEC	<i>“Congress on Evolutionary Computation”</i>	I
CoDE	<i>“Composite DE”</i>	72
DE	Algoritmo de Evolución Diferencial	4
DEPSO	Algoritmo de Evolución Diferencial basado en Optimización por Enjambre de Partículas con mutación auto-adaptativa . . .	I
FAE	Física de Altas Energías	I
GA	Algoritmos Genéticos	4
GD	<i>“Gradient Descent”</i>	71
GS	<i>“Grid Search”</i>	71
HE-DEPSO	Algoritmo de Evolución Diferencial basado en Optimización por Enjambre de Partículas con Información Élite Histórica . .	I
HEP	High Energy Physics	III
IC	Inteligencia Computacional	23
IE	Inteligencia de Enjambre	44
JADE	<i>“Self-Adaptive Differential Evolution”</i>	72
ME	Modelo Estándar de la Física de Partículas	1
MEED	Modelo Estándar Electrodebil	63
POG	Problema de Optimización Global	9

PSO	Algoritmo de Optimización por Enjambre de Partículas	4
RNG	<i>"Random Guessing"</i>	71
SaDE	<i>"Self-Adaptive DE"</i>	73
SHADE	<i>"Success-History based Adaptive DE"</i>	I

Capítulo 1

Introducción

La física, como ciencia fundamental, se puede dividir en tres áreas principales que se complementan entre sí para comprender y explicar los fenómenos naturales: la física teórica, que utiliza un formalismo matemático riguroso para desarrollar modelos y teorías; la física experimental, enfocada en la adquisición de datos y la realización de experimentos para validar o refutar las teorías propuestas; y la fenomenología, que actúa como un puente entre ambas, vinculando los modelos teóricos con los resultados experimentales para validar y refinar los modelos, determinar parámetros, distinguir entre diferentes teorías y estudiar las consecuencias observables de los modelos en el mundo real, avanzando así en el conocimiento y la comprensión de los fenómenos físicos desde la formulación de teorías hasta la validación experimental y la aplicación práctica de los modelos desarrollados.

Un claro ejemplo de la aplicación de estas áreas de la física es el Modelo Estándar de la Física de Partículas (ME), que utiliza un formalismo matemático para describir las interacciones fundamentales entre las partículas elementales, y que ha sido validado experimentalmente a través de experimentos en aceleradores de partículas como el LHC en el CERN. A pesar del éxito del Modelo Estándar en explicar muchos fenómenos, sigue teniendo dificultades en explicar ciertas propiedades y estructuras de las partículas fundamentales, como la masa de los quarks y leptones.

Uno de los problemas más importantes es el conocido como “problema del sabor”, que se refiere a grandes rasgos a la falta de explicación sobre por qué los quarks y leptones tienen las masas y propiedades específicas que tienen, y por qué no hay más partículas con propiedades similares. En este contexto, las investigaciones sobre la generación de masa de los fermiones y la reproducción de los elementos de la matriz V_{CKM} , conocida como la matriz de mezcla, son una parte importante de

la física de partículas. Estas investigaciones tienen sus inicios en los primeros años de la década de los 70s, poco después del establecimiento del ME. Desde entonces, se han desarrollado diferentes enfoques, como los mecanismos radiactivos [2, 3], texturas [4–6], simetrías entre familias [7, 8] y mecanismos de seesaw [9–11], que se encuentran relacionados entre sí y buscan explicar la estructura y propiedades de las partículas fundamentales, incluyendo el problema del sabor.

En el ámbito del ME, las masas de los quarks se originan a partir de matrices hermiticas 3×3 conocidas como matrices de masa (una para los quarks tipo u y otro para los quarks tipo d) [12–14], como los valores absolutos de sus eigenvalores y la matriz V_{CKM} como el producto de la matriz que diagonaliza a los quarks tipo u y la matriz que diagonaliza a los quarks tipo d. Sin embargo, debido al formalismo matemático utilizado, las matrices de masa permanecen desconocidas, lo que impide predecir teóricamente las masas de los quarks y la matriz de mezcla V_{CKM} , siendo el experimento quien proporciona el valor numérico de estas cantidades.

El formalismo de texturas nace al considerar que ciertas entradas de la matriz de masa son cero, de modo que se pueda calcular analíticamente la matriz que la diagonaliza y, por ende, la matriz V_{CKM} . En 1977, Harald Fritzsch creó este formalismo al utilizar las texturas hermiticas de 6 ceros como un modelo viable [15]. En 2005, con datos experimentales de aquella época, encontró que las texturas hermiticas con 4 ceros eran viables para generar las masas de los quarks y la matriz de mezcla. Sin embargo, es interesante constatar la viabilidad de estas texturas con los datos experimentales actuales, así como los nuevos criterios utilizados hoy en día.

1.1. Planteamiento del Problema

Considerando un enfoque fenomenológico de la física de partículas, se estudia el modelo teórico de texturas de cuatro ceros para las matrices de masa de quarks para verificar su validez, comparando la matriz V_{CKM} obtenida del modelo con datos experimentales actuales. En este proceso de comparación, se requiere de un análisis numérico en el cual se utiliza el método de Chi-cuadrado (χ^2) para evaluar la concordancia entre la parte teórica y la experimental. La validación del

modelo involucra construir la función χ^2 y encontrar valores permisibles de los parámetros libres del modelo que satisfagan un criterio propuesto. En este trabajo se utilizará el criterio de Chi-cuadrado reducido (χ_{red}^2) y se buscarán soluciones de la función χ^2 que cumplan con la condición $0 < \chi_{red}^2 < 1$. Sin embargo, la manipulación algebraica compleja de las expresiones involucradas en la construcción de la función χ^2 puede dificultar la búsqueda de estas soluciones.

Este problema debe ser tratado como un problema de optimización global de un solo objetivo con restricciones de frontera, el cual puede ser modelado de la siguiente forma:

$$\min \chi^2(X), \quad X = (x_1, x_2, \dots, x_N) \quad (1.1.1)$$

donde $\chi^2 : \mathbb{R}^N \rightarrow \mathbb{R}$ define una función objetivo con valores reales, y N la dimensión o número de parámetros libres del modelo de texturas. Los parámetros x_1, x_2, \dots, x_N se encuentran delimitados por sus respectivas fronteras, es decir, $x_N \in [a_N, b_N]$, donde a_N es la frontera inferior del N -ésimo parámetro libre y b_N la frontera superior del N -ésimo parámetro libre.

1.2. Justificación

En el análisis numérico de modelos de texturas, la construcción de la función χ^2 implica el manejo de expresiones matemáticas complejas que componen la matriz V_{CKM} . Estas expresiones no triviales dificultan la obtención directa de derivadas o gradientes, lo que limita o imposibilita la aplicación de técnicas de optimización exactas como el descenso del gradiente. Además, el uso de técnicas exhaustivas puede requerir tiempos de cómputo significativos y no obtener buenos resultados. Por lo tanto, se hace necesario recurrir a técnicas de optimización más avanzadas para abordar este problema de manera eficiente.

Los ABI, han sido poco explorados en la FAE, en contextos experimentales se han implementado Algoritmos Evolutivos (AE) con el fin de mejorar el análisis de grandes cantidades de datos [16], así como la optimización del diseño de aceleradores de partículas [17]. Referente a la fenomenología, Algoritmos Genéticos

(GA) han sido utilizados con el fin discriminar modelos provenientes de teorías supersimétricas [18]. La aplicación de estos algoritmos ha probado ser favorable. Particularmente, el Algoritmo de Evolución Diferencial (DE) y el Algoritmo de Optimización por Enjambre de Partículas (PSO) son técnicas de optimización prometedoras para resolver el problema de optimización de la función χ^2 en el análisis numérico del modelo de texturas de cuatro ceros considerado en este trabajo.

El algoritmo DE es eficiente en la exploración global, pero puede tener dificultades para converger rápidamente. PSO se destaca por su rápida convergencia, pero puede quedar atrapado en óptimos locales. Un algoritmo híbrido que combinara estas fortalezas podría llevar a una optimización más eficiente y precisa de la función χ^2 . El algoritmo podría alternar entre exploración y explotación, utilizando DE para ampliar el espacio de búsqueda y PSO para refinar la solución.

DEPSO [19] es una hibridación efectiva entre DE y PSO, pero puede presentar dificultades debido a la selección aleatoria de los parámetros C_r y F , y de convergencia prematura debido a su estrategia de mutación. Una nueva variante mejorada del algoritmo DEPSO podría incorporar la información histórica de los mejores individuos y ajustes dinámicos de los parámetros de control, lo que permitiría una mejor capacidad de exploración y adaptación a las características específicas del problema de optimización de la función χ^2 en el análisis numérico del modelo de texturas de cuatro ceros. Esto podría llevar a una optimización más precisa y estable de la función χ^2 , así como en otros problemas de optimización.

1.3. Hipótesis

Incorporar algoritmos bio-inspirados dentro del análisis numérico de modelos de texturas ayuda en el proceso de optimización de la función χ^2 necesaria en dicho análisis, permitiendo de forma particular validar y analizar a mayor profundidad el modelo de texturas con cuatro ceros.

1.4. Objetivo General

Desarrollar un algoritmo computacional que optimice la función χ^2 con alta precisión utilizando técnicas bio-inspiradas para su implementación en el análisis numérico del modelo de texturas con cuatro ceros.

1.4.1. Objetivos Específicos

- Revisar la literatura reciente sobre el paradigma metaheurístico bio-inspirado, destacando sus aplicaciones, beneficios y limitaciones, con un enfoque especial en su utilización en la física de altas energías.
- Contribuir al avance en el campo de la hibridación de técnicas metaheurísticas, mediante el diseño y implementación de un algoritmo bio-inspirado híbrido innovador que integre la potencia de diferentes enfoques para resolver problemas complejos.
- Integrar estrategias que permitan mejorar las capacidades de exploración y explotación dentro del espacio de búsqueda, mediante las cuales, el algoritmo propuesto adquiera características deseables como: control adaptativo de parámetros, precisión, y estabilidad.
- Definir el conjunto de funciones de prueba a través del cual se mediará el desempeño del algoritmo propuesto.
- Validar el desempeño del algoritmo propuesto mediante la comparación con otros algoritmos relacionados y analizar las diferencias en su rendimiento.
- Mediante el uso del algoritmo propuesto analizar la viabilidad del modelo de texturas con cuatro ceros.

1.5. Metodología

La metodología propuesta para llevar a cabo el presente trabajo se muestra en la Figura 1.5.1.

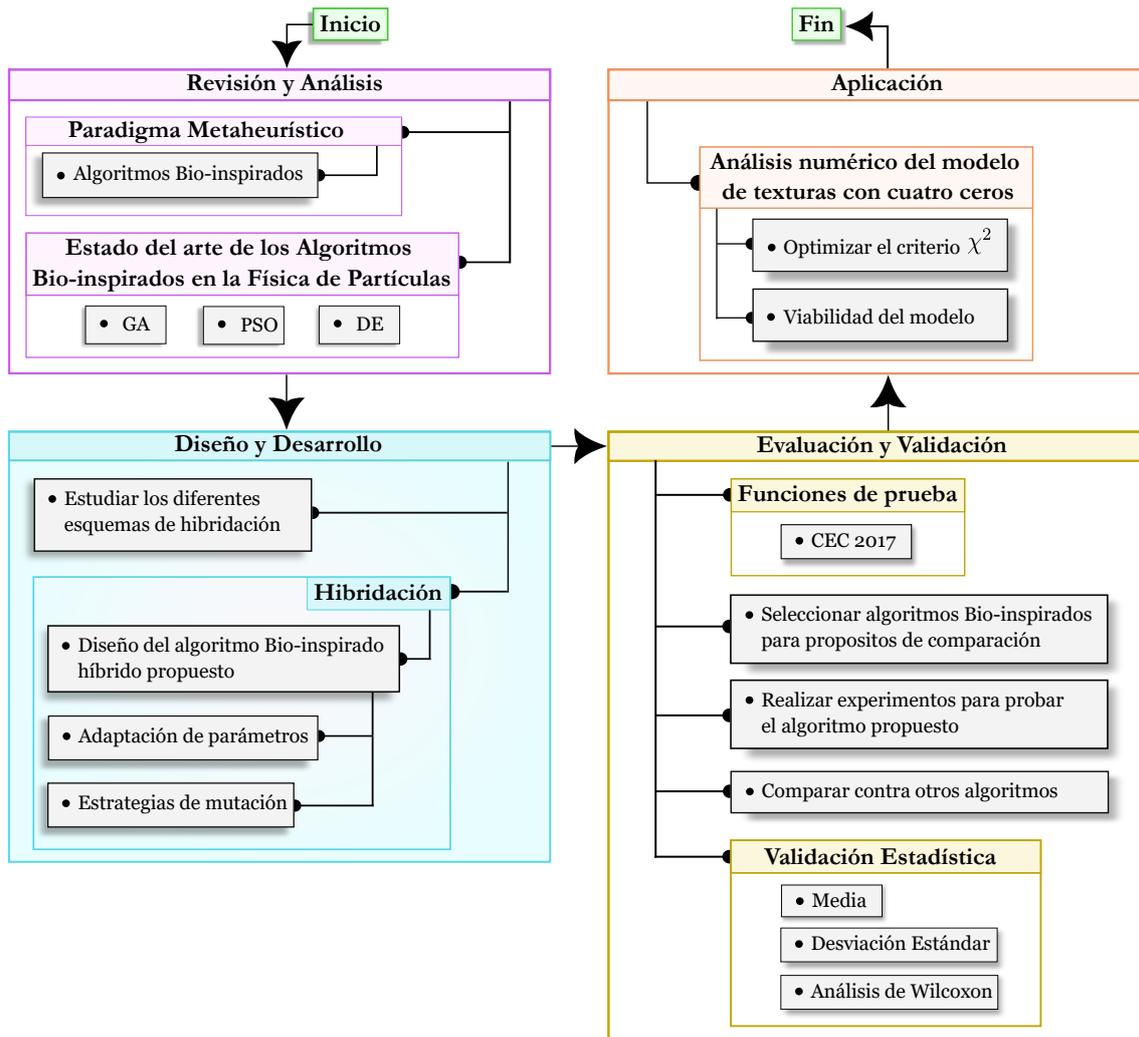


Figura 1.5.1: Metodología propuesta.

En primer lugar, se tiene una etapa de revisión y análisis. En esta etapa, se revisará el paradigma metaheurístico y se realizará un estudio del estado del arte sobre el uso de algoritmos bio-inspirados en física de partículas. Después de conocer los algoritmos bio-inspirados más representativos en el área de física de partículas, así como sus ventajas y desventajas, en la siguiente etapa, denominada diseño y desarrollo, se estudiarán los esquemas para hibridar dos algoritmos bio-inspirados. Considerando lo anterior, en esta etapa se desarrollará el algoritmo bio-inspirado híbrido propuesto. Para evaluar y validar el rendimiento del algoritmo propuesto, en la etapa de evaluación y validación, se seleccionarán algunos algoritmos bio-inspirados relacionados para un estudio comparativo. Finalmente, en la última etapa de la metodología, se implementará el algoritmo propuesto como parte del análisis del modelo de texturas con cuatro ceros. En esta última etapa, según la

información obtenida mediante el algoritmo propuesto, se analizará la viabilidad de este modelo teórico.

1.6. Organización de la Tesis

Este documento de tesis se encuentra organizado en 6 capítulos. En los primeros dos (incluyendo este capítulo), se describen conceptos básicos relacionados con el tema de investigación. El Capítulo 2 proporciona un marco teórico sobre la optimización, así como las técnicas metaheurísticas, incluyendo algoritmos de optimización bio-inspirados. También se presentan algunos conceptos teóricos importantes relacionados con el modelo estándar. Este capítulo concluye con la teoría relacionada con el formalismo de texturas. En el Capítulo 3, se presenta el estado del arte concerniente al análisis numérico del modelo de texturas con cuatro ceros, los algoritmos de optimización bio-inspirados utilizados dentro de la física de partículas y las principales mejoras realizadas sobre el algoritmo DE. El Capítulo 4 presenta la descripción detallada del algoritmo propuesto, y en el Capítulo 5, se analiza la viabilidad e implicaciones fenomenológicas del modelo de texturas con cuatro ceros, teniendo en cuenta la información obtenida mediante el algoritmo propuesto. Finalmente, en el Capítulo 6, se proporcionan las conclusiones generales y el trabajo futuro.

Capítulo 2

Marco Teórico

Este capítulo se centra en presentar los conceptos básicos relacionados con la presente investigación, abarcando desde el tipo de problema de optimización global que se intenta resolver hasta las técnicas computacionales de optimización utilizadas para abordarlo. Además, se introducen conceptos relevantes del modelo estándar de la física de altas energías y se presentan los conceptos fundamentales dentro del formalismo de texturas en el que se desarrolla el problema a resolver.

2.1. Optimización

La optimización se enfoca en identificar los extremos de funciones o sistemas, siendo un área fundamental de las matemáticas. Sus raíces se remontan a filósofos y matemáticos como Pitágoras, Platón, Aristóteles, Arquímedes y Herón, quienes establecieron los fundamentos. Posteriormente, científicos como Newton, Leibniz, Euler y Lagrange contribuyeron significativamente a su desarrollo [20]. El continuo progreso en esta área, unido al avance tecnológico, ha permitido resolver problemas complejos en diversas áreas y escenarios.

2.1.1. Problemas de Optimización Global

Un Problema de Optimización Global (POG) es formulado de la siguiente manera [21]:

Dada una función $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ y un conjunto $S \in \mathbb{R}^n$, se busca encontrar \mathbf{x}^* tal que:

$$f(\mathbf{x}^*) = \min \{f(\mathbf{x}) \mid \forall \mathbf{x} \in S\}, \quad (2.1.1)$$

en donde $\mathbf{x} = (x_1, x_2, \dots, x_n)$ es un vector de parámetros, S es el espacio de búsqueda y n corresponde a la dimensión del POG. La función f , también conocida como función objetivo, función de pérdida, criterio o función de ajuste [22–24], es utilizada para evaluar la calidad de las soluciones candidatas dentro del espacio de búsqueda.

Siguiendo la formulación de un POG descrita anteriormente, se asume un problema que busca minimizar la función objetivo f . Aunque se está asumiendo la minimización, no se restringe la generalidad de los resultados, ya que se puede establecer una equivalencia entre problemas de maximización y minimización de la siguiente forma [25]:

$$\max \{f(\mathbf{x}) \mid \forall \mathbf{x} \in S\} = \min \{-f(\mathbf{x}) \mid \forall \mathbf{x} \in S\}. \quad (2.1.2)$$

Las restricciones son un componente clave en problemas de optimización global, ya que establecen los límites y condiciones que deben considerarse al buscar la mejor solución. Estas pueden presentarse como funciones o como restricciones de frontera. Las restricciones de frontera quedan establecidas de la siguiente manera [26, 27]:

$$a_i \leq x_i \leq b_i, \quad i \in \{1, 2, \dots, n\}, \quad (2.1.3)$$

donde a_i es la frontera inferior y b_i es la frontera superior correspondiente al parámetro i . Un POG el cual requiera exclusivamente de restricciones de frontera, es denominado como un POG sin restricciones [24].

Por otra parte, las funciones de restricción, se pueden identificar en dos tipos, las restricciones de igualdad (Ecuación (2.1.4)) y las restricciones de desigualdad (Ecuación (2.1.5) y Ecuación (2.1.6)) [28].

$$g_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, P, \quad (2.1.4)$$

donde P es el número de igualdades.

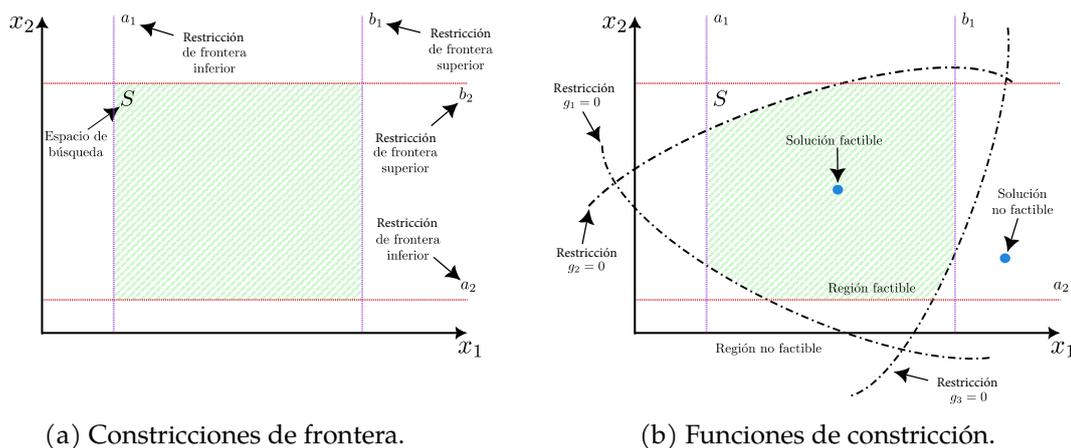
$$h_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, Q, \quad (2.1.5)$$

$$h_j(\mathbf{x}) \geq 0, \quad j = 1, 2, \dots, Q, \quad (2.1.6)$$

donde Q es número de desigualdades. Estas funciones de restricción, definen lo que se conoce como región factible.

En la Figura 2.1.1, se muestran ambos tipos de restricciones en un POG en un espacio de búsqueda bidimensional ($n = 2$). La Figura 2.1.1(a) ilustra las restricciones de frontera, donde el espacio de búsqueda S se define por la región delimitada por las intersecciones de las restricciones inferior y superior respectivas a cada eje coordenado. La Figura 2.1.1(a) muestra cómo las funciones de restricción generan la región factible, que es la región en la que todos los vectores de parámetros son soluciones factibles para el POG. En contraste, aquellos vectores de parámetros fuera de esta región son considerados no factibles. En un POG sin restricciones, la región factible coincide con el espacio de búsqueda.

El término “*Problema de Optimización Global sin restricciones*” puede conducir a una confusión debido a que, en realidad, se está utilizando un solo tipo de restricciones, las de frontera. El motivo de este término es que la mayoría de problemas de optimización global tienen este tipo de restricciones, que son fáciles de satisfacer en comparación con las funciones de restricción.



(a) Constricciones de frontera.

(b) Funciones de constricción.

Figura 2.1.1: Restricciones en un Problema de Optimización Global considerando un espacio de búsqueda de dos dimensiones ($n = 2$).

Al considerar el tipo de espacio de búsqueda S dentro del cual se encuentra definido un POG, se identifican tres tipos de problemas de optimización: continuos,

discretos (combinatorios) y una combinación entre los dos anteriormente mencionados [23]. En la Figura 2.1.2 se ilustran dos tipos de problemas de optimización global de acuerdo a su dominio. La Figura 2.1.2(a) muestra el caso en el cual S pertenece a un dominio continuo. En este caso la función f es una función suave en todo $\mathbf{x} \in S$. Por otro lado, en la Figura 2.1.2(b) se presenta el caso discreto. Al estar S dentro de un dominio discreto, se cuenta con la particularidad de que \mathbf{x} puede tomar una cantidad contable de valores en S .

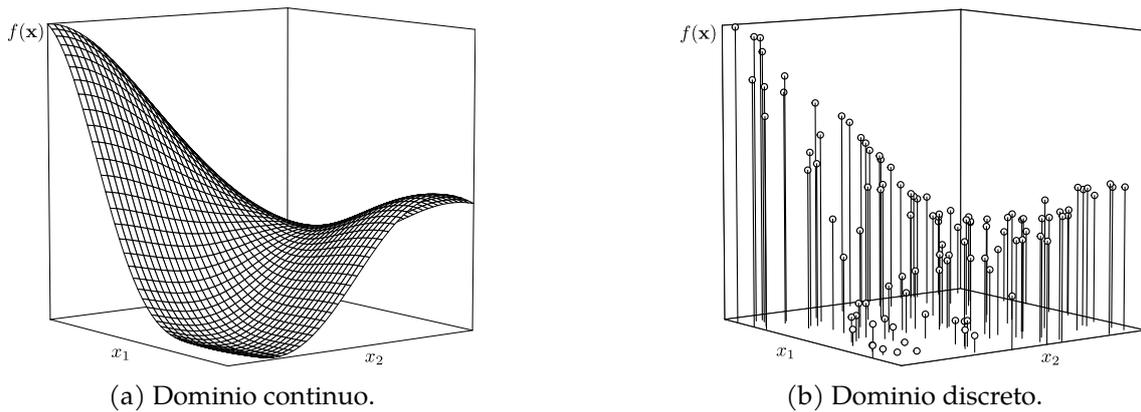


Figura 2.1.2: Tipos de problemas de optimización global de acuerdo a su dominio.

Al resolver problemas de optimización global, es importante definir una medida de cercanía o proximidad entre soluciones. Dos soluciones se consideran próximas si se encuentran dentro del mismo vecindario. Siguiendo este contexto, la estructura de vecindario de una solución en un problema de optimización se puede definir como [22]:

$$N : S \rightarrow S. \tag{2.1.7}$$

de forma que, para cada $\mathbf{x} \in S$ le corresponde un conjunto $S_i \subseteq S$. Además se tiene que, si \mathbf{x}_1 se encuentra en el vecindario de \mathbf{x}_2 , entonces \mathbf{x}_2 se encuentra en el vecindario de \mathbf{x}_1 , en otras palabras, $\mathbf{x}_2 \in S_1$ si y solo si $\mathbf{x}_1 \in S_2$. La figura Figura 2.1.3 ilustra esta situación en una función unidimensional f . La solución \mathbf{x}_1 (color rojo) tiene definido el vecindario S_1 (representado por la región en color rojo) el cual contiene a la solución \mathbf{x}_2 , mientras que la solución \mathbf{x}_2 (color azul) cuyo vecindario es S_2 (representado por la región en color azul) abarca a la solución \mathbf{x}_1 . De esta forma ambas soluciones permanecen dentro de la intersección de sus vecindarios.

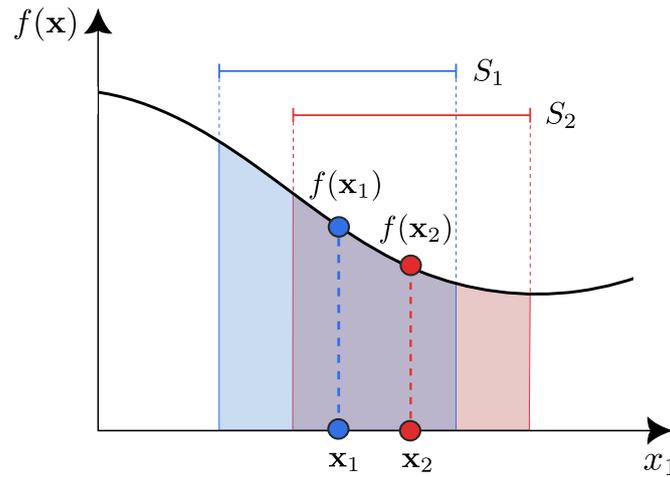


Figura 2.1.3: Soluciones x_1 y x_2 (puntos color rojo y azul) para la función unidimensional f y su correspondiente vecindario (S_1 y S_2 respectivamente).

En la optimización de un solo objetivo, se busca maximizar o minimizar una función objetivo única. Una solución óptima en este contexto, representa un valor máximo o mínimo [29]. Esta solución a su vez, puede ser óptima solo dentro de su vecindario y no sobre todo el espacio de búsqueda. Lo anterior da pie a la definición de óptimo local.

Un **óptimo local** $x_1 \in S_l$ de una función $f : S_l \rightarrow \mathbb{R}$ es una solución la cual satisface la siguiente desigualdad (asumiendo minimización) [22, 29]:

$$f(x_1) < f(x), \quad \forall x \in S_l. \quad (2.1.8)$$

Por otro lado, un **óptimo global** sigue esta desigualdad considerando todo el espacio de búsqueda, es decir, un óptimo global $x^* \in S$ de una función $f : S \rightarrow \mathbb{R}$ es una solución la cual satisface la siguiente desigualdad (asumiendo minimización):

$$f(x^*) < f(x), \quad \forall x \in S. \quad (2.1.9)$$

En la Figura 2.1.4, se muestran los óptimos locales y globales de una función f en dos dimensiones. La figura presenta dos óptimos locales, un mínimo y un máximo, y dos óptimos globales, también un mínimo y un máximo. En ambos casos, un óptimo global es el mejor resultado dentro de todo el espacio de búsqueda S ,

mientras que un óptimo local es el mejor resultado solo en un subconjunto de S o en su vecindario inmediato.

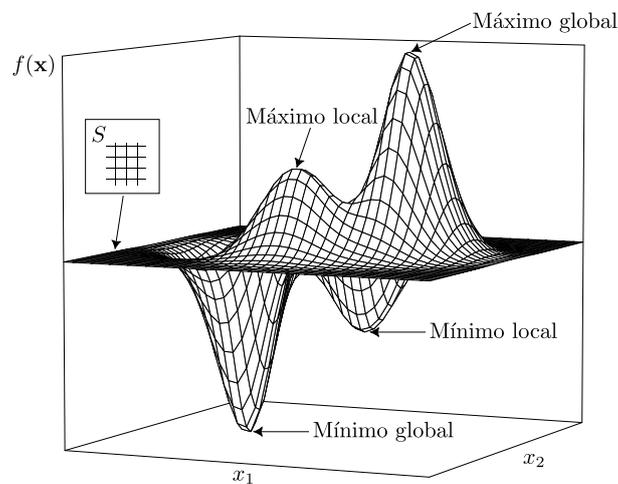


Figura 2.1.4: Óptimos locales y globales en una función bidimensional f .

En problemas de optimización global, la función objetivo f puede presentar varios valores óptimos, clasificados en óptimos locales y globales. Si f tiene un solo óptimo en todo el espacio de búsqueda, se considera **unimodal**. Por otro lado, si hay múltiples óptimos en el espacio de búsqueda, se considera **multimodal** [24, 29].

La Figura 2.1.5 ilustra una función unimodal bidimensional f (Función Esférica [24]), junto con sus curvas de nivel. Estas curvas de nivel visualizan claramente la existencia de un solo óptimo, que es el mínimo global $f(\mathbf{x}) = 0$ situado en $\mathbf{x} = (0, 0)$. Por otro lado, la Figura 2.1.6 muestra una función multimodal de dos dimensiones (Función de Griewank [24]), con sus respectivas curvas de nivel. Aunque esta función presenta múltiples óptimos locales, fácilmente identificables mediante sus curvas de nivel, su mínimo global es $f(\mathbf{x}) = 0$ en $\mathbf{x} = (0, 0)$.

Es posible que un POG tenga múltiples óptimos globales [24], en cuyo caso se puede definir un **conjunto óptimo** X^* que agrupe todas las soluciones óptimas (globales) [29]. La Figura 2.1.7 muestra dos ejemplos de problemas de optimización global unidimensionales con uno y múltiples mínimos globales. En la Figura 2.1.7(a) se presenta un problema de optimización mediante la función $f(\mathbf{x}) = x_1^2$ dentro del espacio de búsqueda $S = [-1, 1]$. Esta función unidimensional tiene un solo óptimo global ubicado en $\mathbf{x} = \mathbf{0} = (0)$, con lo cual es este caso, el

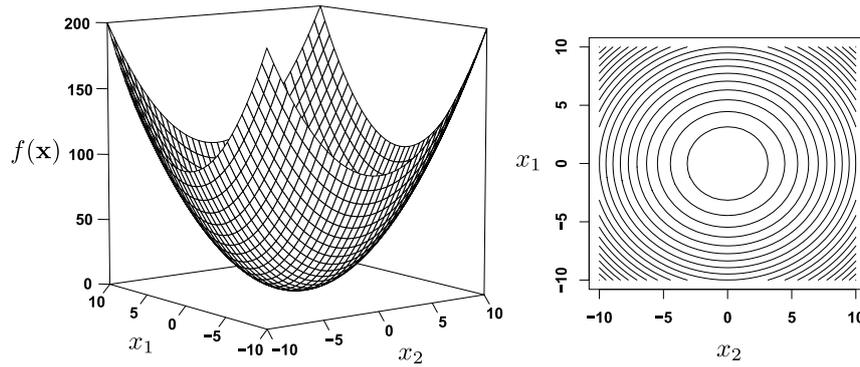


Figura 2.1.5: Función objetivo unimodal (Función Esférica) y sus respectivas curvas de nivel.

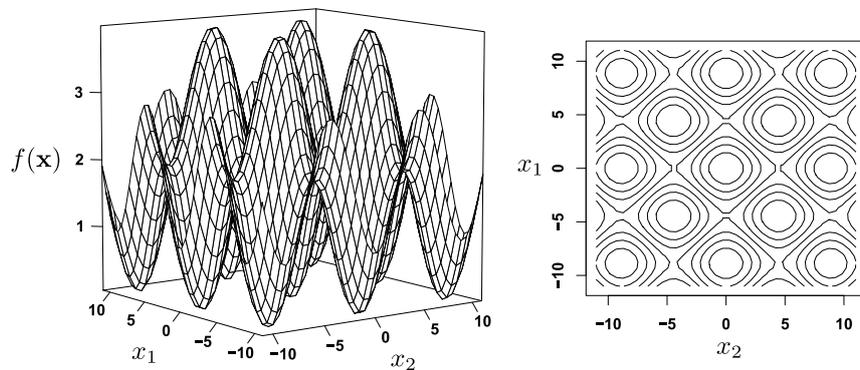
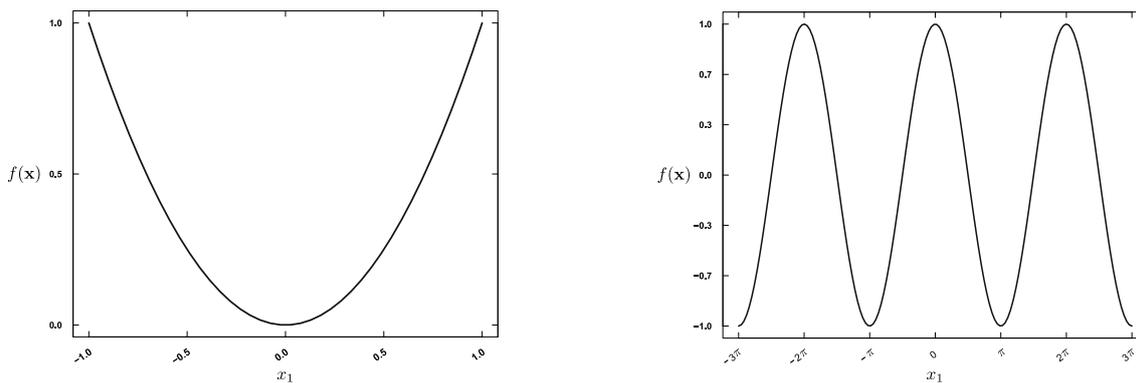


Figura 2.1.6: Función objetivo multimodal (Función de Griewank) y sus respectivas curvas de nivel.

conjunto óptimo \mathbf{X}^* es un conjunto óptimo unitario, es decir, $\mathbf{X}^* = \{0\}$. Por otro lado, la Figura 2.1.7(b) se tiene un problema de optimización representado por la función $f(x) = \cos(x_1)$ dentro del espacio de búsqueda $S = [-3\pi, 3\pi]$. En este caso se tienen múltiples mínimos globales ubicados en: $\mathbf{x} = (-3\pi)$, $\mathbf{x} = (-\pi)$, $\mathbf{x} = (\pi)$ y $\mathbf{x} = (3\pi)$. En este contexto, $\mathbf{X}^* = \{(-3\pi), (-\pi), (\pi), (3\pi)\}$ es un conjunto contable con un número finito de elementos. De forma general, esta función presenta múltiples mínimos globales ubicados en $\mathbf{x}_i = ((2i + 1))\pi$ y múltiples máximos globales ubicados en $\mathbf{x}_i = (2i\pi)$, para todo $i \in \mathbb{Z}$. El conjunto óptimo $\mathbf{X}^* = \{((2i + 1)\pi) : i \in \mathbb{Z}\}$ de mínimos globales y el conjunto óptimo $\mathbf{X}^* = \{(2i\pi) : i \in \mathbb{Z}\}$ de máximos globales son definidos como conjuntos contables con un número infinito de elementos.

Esta investigación se enfoca en la solución de un POG de un solo objetivo, sin restricciones, con variables (parámetros) continuas.



(a) Función con un solo óptimo global, donde $f(\mathbf{x}) = x_1^2$.

(b) Función con múltiples óptimos globales, donde $f(\mathbf{x}) = \cos(x_1)$.

Figura 2.1.7: Problemas de optimización global con uno y múltiples óptimos globales.

2.1.2. Algoritmos de Optimización

Un POG se considera resuelto cuando se encuentra el óptimo global, sea máximo o mínimo, dependiendo del problema. Sin embargo, encontrar este óptimo puede ser complicado debido a factores como múltiples óptimos locales (ver Figura 2.1.6) o un gran número de parámetros. Para abordar estos desafíos, existen varios algoritmos de optimización global que buscan encontrar el conjunto óptimo de soluciones para un problema específico [29]. Dentro de la literatura, es posible encontrar distintas clasificaciones para estos algoritmos [20, 22, 26, 29–32]. De forma general se puede distinguir entre dos clases principales de algoritmos: **Algoritmos de optimización exactos** y **Algoritmos de optimización aproximados** [20, 22, 32, 33].

2.1.2.1. Algoritmos de Optimización Exactos

La teoría del cálculo permite resolver algunos problemas de optimización sin restricciones de manera analítica, encontrando los ceros del gradiente y verificando que la matriz Hessiana sea positiva definida en esos puntos [20]. Esto se logra asumiendo que la función f asociada es clase C^2 , es decir, que tenga derivadas de segundo orden continuamente diferenciables. Sin embargo, cumplir con estas condiciones puede ser un desafío en muchos casos. Otros métodos analíticos buscan determinar el óptimo global de una función f , considerando características

específicas como convexidad [20, 34].

Dentro de las técnicas numéricas de optimización, los algoritmos de optimización exactos son aquellos que siguen un comportamiento determinista, lo que significa que siempre obtienen la misma solución cuando se ejecutan desde el mismo punto de inicio. Estos algoritmos iterativos ajustan la dirección y el rango de búsqueda en cada iteración. Se clasifican en dos categorías: aquellos basados en el gradiente de la función objetivo y aquellos basados en la evaluación de la función objetivo [20, 35].

Los algoritmos exactos basados en el gradiente de la función objetivo buscan la solución óptima mediante la resolución de un conjunto de ecuaciones no lineales que igualan a cero el gradiente de la función objetivo. Alternativamente, se puede partir de una solución inicial x y generar una secuencia de soluciones mejoradas iterativamente [35]. Algunos de los algoritmos más destacados en esta categoría son el descenso del gradiente, el método de Newton-Raphson y los métodos Quasi-Newton [20]. Estos algoritmos son particularmente eficientes en la búsqueda local de óptimos en entornos como los mostrados en las Figura 2.1.5 y Figura 2.1.7(a).

Los algoritmos basados en la evaluación de la función objetivo, se enfocan en evaluar de forma iterativa en la función objetivo a cada una de las posibles soluciones dentro del espacio de búsqueda, esto con el fin de tratar de encontrar el valor del óptimo global. Este tipo de algoritmos sufren de lo que se denomina maldición de la dimensión, lo cual significa que debido al tamaño que pueda tener del espacio de búsqueda de un problema, el algoritmo será ineficiente debido a la gran cantidad de evaluaciones que debe realizar [35]. El algoritmo de Nelder-Mead es un ejemplo de este tipo de algoritmos basados en la evaluación de la función objetivo.

En general, los algoritmos de optimización exactos son eficientes en búsqueda local y aseguran encontrar la solución óptima global [22]. Sin embargo, su aplicación en optimización global es limitada debido a la complejidad de los problemas y los tiempos de cómputo prolongados necesarios [22, 26]. Esto los hace menos viables para problemas con múltiples parámetros, restricciones y funciones objetivo.

2.1.2.2. Algoritmos de Optimización Aproximados

A causa de las limitaciones de los algoritmos, métodos y técnicas de optimización exactos, los algoritmos de optimización aproximados (también llamados estocásticos o probabilísticos) han ganado relevancia en las últimas décadas. Estos algoritmos, que se remontan a la década de los años 50s [36], no garantizan encontrar el óptimo global (en la mayoría de los casos), pero pueden encontrar una solución óptima en un tiempo significativamente menor que los algoritmos exactos [22]. Aunque no garantizan la solución óptima global, no implica que la solución obtenida sea incorrecta; simplemente no es la óptima global. En muchos casos, encontrar una solución suficientemente buena es mejor que no tener una solución o esperar cientos de años para encontrarla.

Dentro de los algoritmos de optimización aproximados, los algoritmos metaheurísticos han adquirido gran relevancia por su capacidad para resolver y encontrar soluciones óptimas en una amplia variedad de problemas de optimización, sin necesidad de un conocimiento profundo en los detalles y definición de tales problemas [33, 37]. En las siguientes secciones, se profundizará en este tipo de algoritmos, que tienen una estrecha relación con la presente investigación. Antes de continuar, se presentará el Teorema *No free lunch*, uno de los teoremas más influyentes en el campo de los algoritmos de optimización.

2.1.2.3. The No Free Lunch Theorem

Ante la existencia de diversos algoritmos de optimización, surge la pregunta de cuál es el más adecuado para un problema específico. Wolpert y Macready investigaron esta cuestión y formularon el teorema *No free lunch* [38], que establece que, para cualquier par de algoritmos A_1 y A_2 , si A_1 supera a A_2 en ciertos problemas, entonces A_2 superará a A_1 en otros [39].

$$\sum_f P(x_m|f, A_1) = \sum_f P(x_m|f, A_2) \quad (2.1.10)$$

donde $P(x_m|f, A_1)$ es la probabilidad que el algoritmo A_1 identifique la solución óptima para una función objetivo f arbitraria y $P(x_m|f, A_2)$ es la probabilidad

análoga para el algoritmo A_2 .

El teorema *No free lunch* establece que el rendimiento promedio de cualquier algoritmo de optimización es igual para todos los problemas posibles. Esto implica que si un algoritmo es mejor en ciertos problemas, necesariamente tendrá un desempeño inferior en otros. En consecuencia, no es posible desarrollar un algoritmo de optimización universal que pueda resolver cualquier problema de optimización. Las características de un problema de optimización pueden influir en el desempeño de un algoritmo, lo que hace que sea importante considerar el conocimiento específico del problema y evitar comparar algoritmos sobre una muestra reducida de problemas [39].

2.1.3. Metaheurísticas

Las metaheurísticas son algoritmos de optimización aproximados que pueden resolver problemas de optimización de diversas clases sin necesidad de ser adaptados a cada tipo de problema o conocer su estructura detallada. Estos algoritmos utilizan técnicas de aproximación con componentes estocásticos, llamadas heurísticas, para explorar el espacio de búsqueda [22]. Las heurísticas ayudan al algoritmo a tomar decisiones para encontrar soluciones óptimas, utilizando información actual del proceso de optimización [29]. El término “metaheurísticas” incluye el prefijo griego “meta”, indicando que estos algoritmos son un procedimiento de nivel superior para generar heurísticas. Los algoritmos metaheurísticos tienen características como implementación sencilla, robustez en su aplicación a problemas de optimización variados, no requieren cálculo de derivadas y, debido a su naturaleza estocástica, tienen mejor capacidad para evitar mínimos locales. Estas características han llamado la atención de varios investigadores y han sido implementadas en diversas aplicaciones [40–43].

La mayoría de los algoritmos metaheurísticos se basan en conceptos naturales. Algunos se inspiran en la evolución biológica, el comportamiento animal, procesos físicos y químicos, y otros fenómenos naturales [44].

Este tipo de algoritmos, en general, siguen un mecanismo que permite encontrar soluciones óptimas. La Figura 2.1.8 ilustra este proceso. La búsqueda de solucio-

nes óptimas comienza con la generación aleatoria de un conjunto inicial de posibles soluciones dentro del espacio de búsqueda (círculos verdes). Este conjunto se conoce por diferentes nombres en la literatura, como enjambre, población, colonia o grupo. Cada elemento de este conjunto se puede denominar de manera distinta, como individuo, cromosoma o partícula. Luego, se aplican operadores o métodos para combinar los elementos del conjunto inicial. A partir de esto, se genera un nuevo conjunto de soluciones modificadas (círculos morados). A continuación, se crea un nuevo conjunto a partir del inicial y modificado (círculos verdes y morados). Finalmente, se selecciona una nueva posible solución óptima bajo cierto criterio del nuevo conjunto de soluciones. Este proceso se repite de forma iterativa hasta que se alcanza un criterio de parada, como un número máximo de iteraciones o un valor de la función objetivo [45].

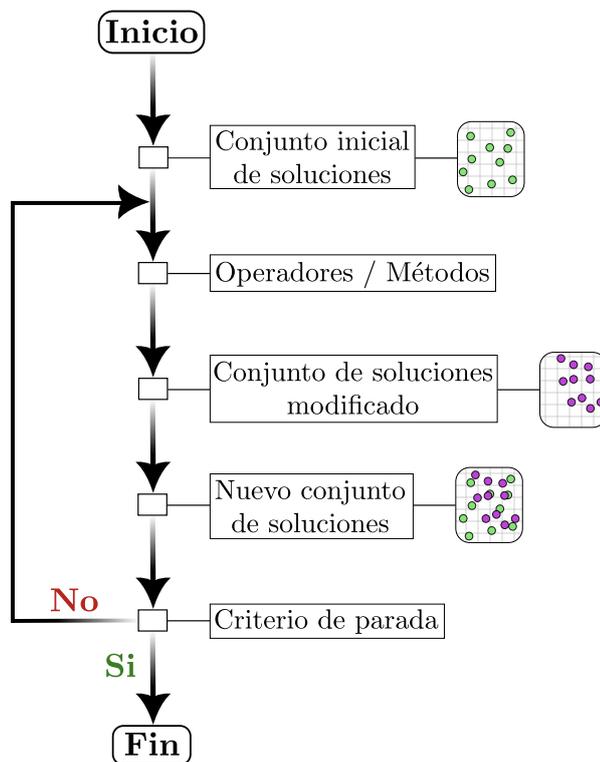


Figura 2.1.8: Mecanismo general de un algoritmo metaheurístico para la búsqueda de una solución óptima.

El proceso de búsqueda en algoritmos de optimización puede dividirse en dos fases: exploración y explotación. La exploración busca identificar áreas prometedoras dentro del espacio de búsqueda que puedan contener soluciones óptimas de alta calidad, mientras que la explotación se enfoca en la búsqueda local den-

tro de estas áreas. La naturaleza estocástica de estos algoritmos beneficia la fase de exploración [46]. El equilibrio entre ambas fases es crucial, ya que afecta significativamente el rendimiento de los algoritmos. Este balance implica encontrar soluciones de alta calidad sin invertir recursos en regiones ya exploradas o que no muestren indicios de tener soluciones óptimas [46].

Dentro del amplio espectro de metaheurísticas, cada una se distingue por su enfoque único en lograr el equilibrio entre exploración y explotación. Estas metaheurísticas pueden clasificarse según características como la trayectoria de búsqueda, el uso de memoria histórica, la fuente de inspiración y el uso de poblaciones de soluciones. La literatura ofrece diversas clasificaciones basadas en estas características [1, 45, 47–50]. La clasificación más común, propuesta por Behesht et al. [1], utiliza un diagrama de Euler para categorizar algunos algoritmos metaheurísticos conocidos según características como el tipo de búsqueda (ver Figura 2.1.9), la fuente de inspiración, el tipo de población y la trayectoria de búsqueda.

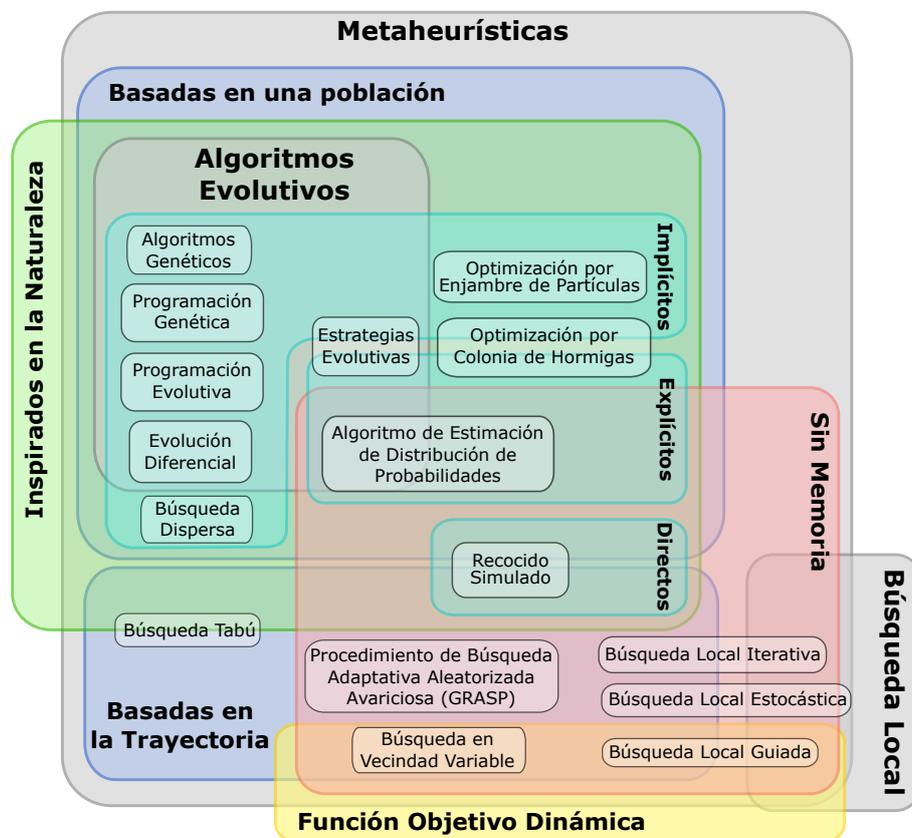


Figura 2.1.9: Clasificación de algunas de las metaheurísticas más conocidas (Adaptado y traducido al español de Behesht et al. [1].).

En el marco de esta investigación, se centrará en un tipo específico de algoritmos metaheurísticos, conocidos como Algoritmos Bio-Inspirados. Estos algoritmos, que se encuentran dentro de la categoría de metaheurísticas inspiradas en la naturaleza, serán descritos en las secciones siguientes [51–54].

2.2. Algoritmos Bio-Inspirados

Algunos de los procesos más eficientes se encuentran en la naturaleza. Un ejemplo es la coordinación hábil de un grupo de aves al buscar alimento. De esta forma, la naturaleza representa una de las mayores inspiraciones a partir de la cual se realiza el diseño de un amplio espectro de algoritmos de optimización [52]. Este tipo de algoritmos de optimización son conocidos como Algoritmos Biológicamente-Inspirados o de forma compacta como Algoritmos Bio-Inspirados.

Dentro de la literatura, los algoritmos bio-inspirados se clasifican según su fuente de inspiración natural [52, 53, 55–57]. La Figura 2.2.1 muestra una clasificación sencilla de estos algoritmos, destacando su relación con la naturaleza. En este trabajo, se enfocan principalmente en dos categorías: Algoritmos Evolutivos, basados en la evolución biológica, y Algoritmos Basados en Inteligencia de Enjambre, que se inspiran en comportamientos sociales colectivos [55]. En las siguientes secciones, se describen estas dos categorías y se presentan algunos de sus algoritmos más representativos.

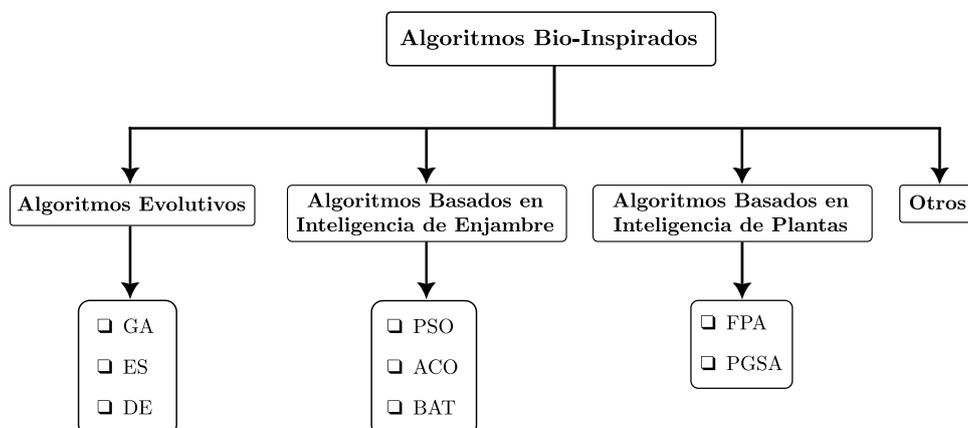


Figura 2.2.1: Clasificación de los Algoritmos Bio-Inspirados.

2.2.1. Algoritmos Evolutivos

El campo de estudio dentro de la Inteligencia Computacional (IC) conocido como Computo Evolutivo (CE) incluye a los AE [24]. Estos algoritmos son una clase de metaheurísticas diseñados para resolver problemas de optimización, inspirados en la Teoría de la Evolución de Darwin y conceptos biológicos.

De acuerdo con la Teoría de la Evolución de Darwin, la evolución de los seres vivos se basa en los siguientes principios:

- **Variaciones:** Existen variaciones en las características individuales entre padres y su descendencia. Cada individuo presenta rasgos únicos que lo diferencian de sus progenitores y hermanos.
- **Herencia:** Estas características se transmiten de generación en generación a través de la herencia genética. Los descendientes heredan una combinación de los rasgos de sus padres.
- **Selección natural:** En una población, los individuos más adaptados al entorno en el que se encuentran tienen mayores probabilidades de sobrevivir y reproducirse (presión selectiva). Este proceso de selección natural favorece la perpetuación de las características más ventajosas para la supervivencia en un determinado medio.

Darwin concluyó que la competencia entre individuos fomenta la transmisión de las mejores variaciones a través de las generaciones, lo que llevó a la evolución natural de organismos complejos adaptados a diferentes entornos [58, 59]. Este éxito en la generación de organismos complejos se ha extrapolado a la optimización, donde se aplican principios de evolución natural. De manera similar, los algoritmos AE utilizan operadores evolutivos para resolver problemas de optimización difíciles.

La mayoría de los AE siguen un patrón de búsqueda guiado por variaciones estocásticas generadas a partir de operadores evolutivos. El proceso comienza generando una población aleatoria de soluciones dentro del dominio del problema de

optimización. Luego, cada solución candidata se evalúa mediante la función objetivo para determinar su calidad. Las soluciones candidatas interactúan y comparan información mediante operadores de cruce y mutación. El cruce combina dos o más soluciones candidatas para generar nuevas soluciones. La mutación aplica variaciones a una solución candidata para generar una nueva posible solución. Estos operadores generan una nueva población de soluciones (descendencia), que luego compite con la población anterior según su calidad. Este proceso se repite iterativamente hasta que se cumpla un criterio de parada, como encontrar una solución suficientemente buena o ejecutar un número establecido de iteraciones (generaciones) [58]. En el Algoritmo 2.2.1 se muestra en forma de pseudocódigo el esquema general de búsqueda que sigue un algoritmo evolutivo.

Algoritmo 2.2.1: Esquema general de un Algoritmo Evolutivo en pseudocódigo.	
1:	Inicializar de forma aleatoria una población de posibles soluciones
2:	Evaluar cada solución en la función objetivo
3:	mientras <i>no se cumpla el criterio de parada</i> hacer
4:	Seleccionar padres
5:	Recombinar pares de padres
6:	Mutar la descendencia resultante
7:	Evaluar la nueva población
8:	Seleccionar individuos para la siguiente generación
9:	fin

La construcción de un algoritmo evolutivo (incluso otros ABI) implica definir varios componentes clave. Estos incluyen la representación de soluciones, la inicialización de soluciones, la selección, el cruce, la mutación y el reemplazo. Además, es necesario establecer un criterio de parada para indicar cuando el algoritmo debe detener su ejecución. A continuación, se describirán estos componentes con mayor detalle. Posteriormente, se presentará uno de los algoritmos evolutivos más representativos relacionados con el presente trabajo de tesis.

2.2.1.1. Representación de soluciones

Uno de los primeros pasos en el diseño de un algoritmo evolutivo es especificar la representación matemática de las posibles soluciones al problema. Esta repre-

sentación permite almacenar las soluciones en una computadora y manipularlas mediante el propio algoritmo evolutivo. Una solución candidata a un problema, expresada como un vector de parámetros, se denomina fenotipo. La representación de esta solución utilizada por el algoritmo evolutivo se conoce como genotipo [60]. En la Figura 2.2.2, se ilustran los espacios a los que pertenecen el fenotipo y el genotipo asociados a una solución candidata. Además, se muestra que un mapeo del espacio fenotipo al espacio genotipo se define mediante un proceso de codificación, mientras que la decodificación se realiza de forma inversa.

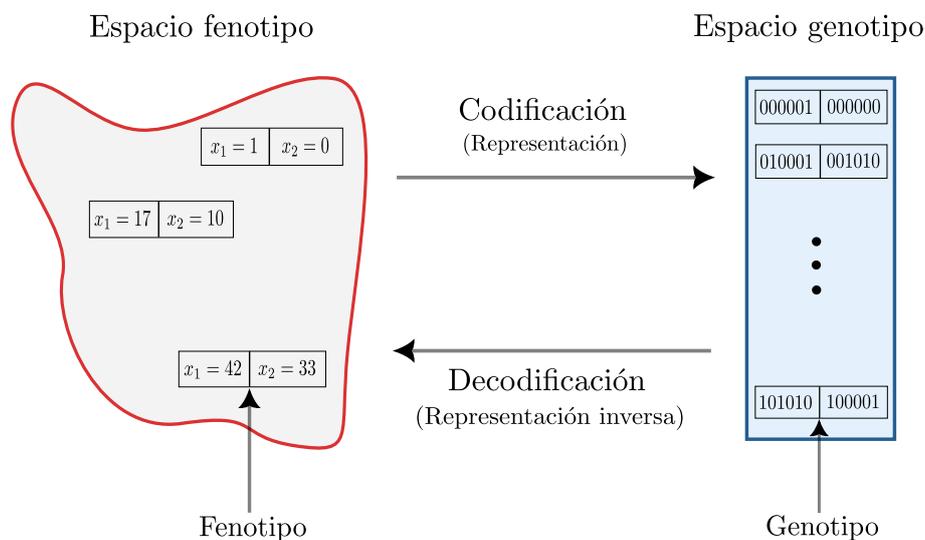


Figura 2.2.2: Espacio fenotipo y genotipo para la representación de una solución.

La representación de las soluciones tiene una gran importancia ya que determina la naturaleza del espacio de búsqueda del algoritmo evolutivo. Además, la forma de la representación impacta directamente en el diseño que tendrán los operadores de cruce y mutación debido a que de acuerdo a la representación, estos operadores pueden variar [61]. A continuación se da una revisión de las representaciones más comunes dentro de los algoritmos evolutivos.

Representación binaria

En los AE, la representación binaria fue una de las primeras formas de representar soluciones. En esta representación, el genotipo se compone de una sucesión de bits, que deben ser configurados adecuadamente para cada problema. Esto incluye determinar la longitud de la cadena de bits y cómo reproducir el fenotipo

asociado. Es fundamental establecer un mapeo entre el genotipo y el fenotipo para asegurar que la decodificación represente una solución válida dentro del contexto del problema original y que toda solución pueda ser codificada en una cadena de bits para ser manipulada por el algoritmo evolutivo [61]. Por ejemplo, en la Figura 2.2.3, se muestra cómo una posible solución ($\mathbf{x} = (58, 30, 32, 12)$) para un problema de optimización con cuatro parámetros se asocia con el genotipo: 111010011110100000001100.

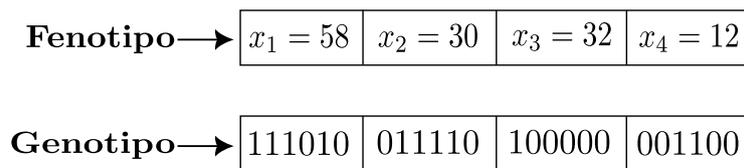


Figura 2.2.3: Representación binaria: fenotipo y genotipo.

En problemas de optimización en los cuales los parámetros de la función objetivo son del tipo booleano, la representación binaria es natural. Sin embargo, puede ser utilizada para representar parámetros con valores enteros y reales [62].

A pesar de la simplicidad de la codificación binaria, se enfrenta a un problema conocido como *Hamming Cliffs*. Esto implica que grandes cambios en el genotipo pueden generar pequeños cambios en el valor entero resultante. Para superar esta limitación, se busca localidad, es decir, que pequeños (grandes) cambios en el genotipo produzcan pequeños (grandes) cambios en el fenotipo y en el valor objetivo [61]. Una forma de abordar este problema es utilizar la *Codificación Gray*. En esta codificación, un cambio en un solo entero requiere solo un cambio en un bit del genotipo binario (ver Tabla 2.2.1), lo que garantiza que soluciones adyacentes en el espacio de búsqueda tengan valores enteros adyacentes en el espacio codificado [60, 61].

Tabla 2.2.1: Codificación binaria y Gray para un valor decimal.

Decimal	Binaria	Gray
0	0000	000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110

Entre los algoritmos evolutivos, la representación binaria es muy utilizada particularmente por los Algoritmos Genéticos [63].

Representación entera

En la representación entera, las soluciones a un problema se representan mediante vectores de parámetros con valores enteros. Cada parámetro puede tener un conjunto infinito de valores enteros o estar limitado a un conjunto finito de valores [58]. En contraste con la representación binaria, no es necesario realizar un proceso de codificación y decodificación entre el fenotipo y genotipo, lo que puede acelerar el proceso de optimización. La Figura 2.2.4 muestra un ejemplo de cómo se representa un fenotipo y genotipo de una posible solución ($\mathbf{x} = (58, 30, 32, 12)$) para un problema de optimización con cuatro parámetros. En este caso, el fenotipo y genotipo son idénticos, lo que permite al algoritmo evolutivo trabajar directamente en el contexto del problema sin necesidad de conversiones adicionales.

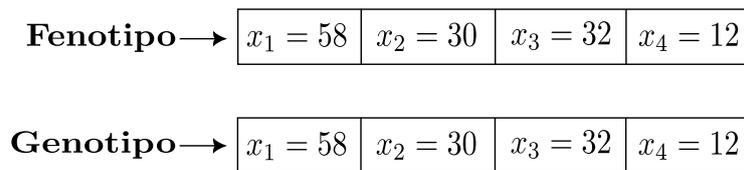


Figura 2.2.4: Representación entera: fenotipo y genotipo.

En términos generales, la representación entera resulta ideal para problemas de optimización en los cuales el dominio del problema de optimización contempla solo el uso de valores enteros.

Representación real o de punto flotante

En problemas de optimización de la vida real, es común que las variables tomen un conjunto continuo de valores, en lugar de un conjunto discreto. La forma más natural de representar una posible solución es mediante un vector de parámetros con valores reales. Esta representación permite especificar valores de cantidades físicas como longitud, ancho, altura o peso de un componente dentro de un umbral determinado. Debido a las limitaciones de precisión de las computadoras para representar valores reales, este tipo de representación también se conoce como

representación de punto flotante [58]. Por ejemplo, al utilizar un algoritmo evolutivo para evolucionar los pesos de las conexiones entre nodos en una red neuronal artificial, se puede emplear esta representación. La Figura 2.2.5 ilustra el fenotipo y genotipo asociados a la representación de punto flotante de una posible solución ($\mathbf{x} = (58, 30, 32, 12)$) a un problema dependiente de cuatro parámetros. En este caso, el genotipo consiste en cuatro valores de punto flotante, lo que permite mejorar la precisión del algoritmo evolutivo en la búsqueda de soluciones óptimas.

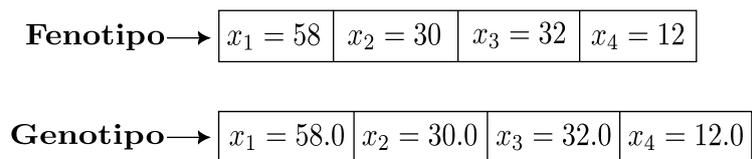


Figura 2.2.5: Representación de punto flotante: fenotipo y genotipo.

La representación de punto flotante es ideal para optimizar parámetros que necesitan alta precisión. Sin embargo, si el problema no requiere una alta resolución del espacio de parámetros, se puede utilizar la representación binaria, asignando un número razonable de bits a cada parámetro [64]. La representación de punto flotante tiene una ventaja significativa sobre la representación binaria, ya que no hay pérdida de precisión debido a la discretización o codificación binaria. Además, ofrece una amplia variedad de operadores que pueden ser utilizados [60].

En las siguientes secciones, se abordarán otros componentes y operadores clave para definir un algoritmo evolutivo. Como este estudio se centra en la optimización global en dominios continuos, los componentes y operadores se describirán bajo la suposición de un dominio o espacio de búsqueda continuo.

2.2.1.2. Inicialización de soluciones

Los algoritmos evolutivos operan sobre una población de individuos, donde cada individuo representa una posible solución codificada a un problema de optimización. La inicialización de estos algoritmos consiste en generar una población inicial diversa, que será sometida a un proceso de evolución [62]. Para asegurar una adecuada exploración del espacio de búsqueda, la población inicial debe ser lo suficientemente grande y variada, de modo que los individuos presenten diferentes valores de ajuste o evaluación sobre la función objetivo [60, 64]. El tamaño

óptimo de la población depende de las características del espacio de búsqueda y del costo computacional asociado a la evaluación de los individuos a lo largo de varias generaciones. Si bien no existe una fórmula precisa para determinar el número de individuos a utilizar, es común emplear poblaciones que van desde cientos hasta miles de individuos [60, 64].

Para mejorar la capacidad de búsqueda global del algoritmo evolutivo, la inicialización de la población debe ser aleatoria [62]. Una forma común de hacerlo es seleccionar valores aleatorios dentro de un rango válido para cada parámetro, especialmente en representaciones reales o de punto flotante [61, 64]. Una vez que la población inicial está generada, se puede calcular su ajuste o evaluación ante la función objetivo.

2.2.1.3. Selección

La selección de individuos más adaptados dentro de una población es importante en el proceso evolutivo, ya que permite identificar a aquellos con mayor probabilidad de sobrevivir y reproducirse. Los algoritmos evolutivos, al intentar imitar la evolución natural, deben utilizar estrategias de selección adecuadas [60]. La mayoría de estas estrategias se basan en selecciones probabilísticas, donde los individuos de mayor calidad tienen más probabilidades de reproducirse [58]. Sin embargo, también es importante considerar la posibilidad de seleccionar individuos de baja calidad para evitar la estancamiento en un mínimo local [61].

La estrategia de selección determina la presión de selección, que indica el grado en que las mejores soluciones son seleccionadas. Una presión de selección baja puede prolongar el proceso de optimización, mientras que una alta presión puede llevar a una convergencia prematura. Un balance entre la presión de selección y la exploración del espacio de búsqueda es clave para un proceso de optimización efectivo. Existen varias estrategias de selección, que pueden clasificarse en dos grandes categorías: selección proporcional al ajuste y selección ordinal [61, 64]. Estas estrategias deben ser cuidadosamente seleccionadas y ajustadas para lograr un proceso de optimización eficiente y evitar la convergencia prematura.

Selección proporcional al ajuste

En esta estrategia de selección la probabilidad de que un individuo de la población sea seleccionado está relacionada a su valor de ajuste $f(\mathbf{x}_i)$ y el ajuste del resto de individuos [64]. Esta probabilidad de selección puede ser calculada de la siguiente manera:

$$p_i = \frac{f(\mathbf{x}_i)}{\sum_{i=1}^N f(\mathbf{x}_i)}, \quad (2.2.1)$$

donde p_i es la probabilidad de selección del individuo o posible solución \mathbf{x}_i y N es el número total de individuos.

Este proceso de selección puede ser representado mediante una ruleta, donde cada sección corresponde a un individuo de la población y su tamaño está directamente relacionado con la probabilidad de selección p_i [64]. Las mejores soluciones tienen secciones más grandes, mientras que las malas soluciones tienen secciones más pequeñas. La figura 2.2.6 muestra una ruleta con cinco individuos ($\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5$). Debido a la probabilidad de selección, el individuo \mathbf{x}_1 tiene más probabilidades de ser seleccionado. La selección se realiza mediante un número aleatorio dentro del intervalo $[0, 1]$ dentro de la ruleta.

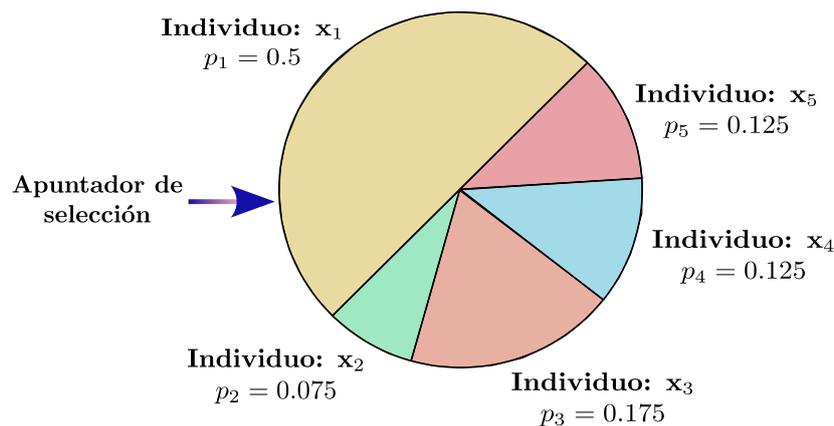


Figura 2.2.6: Selección proporcional al ajuste en donde cada sección de la ruleta corresponde a la probabilidad de selección de cada individuo de la población. El individuo \mathbf{x}_1 es seleccionado.

La selección proporcional al ajuste puede parecer intuitiva, pero puede fallar en dos situaciones específicas. En la primera, si todos los individuos tienen probabi-

lidades de selección similares, la evolución de la población se comporta como una búsqueda aleatoria. En la segunda, si un individuo tiene una probabilidad de selección muy alta, el proceso de optimización tiende a converger prematuramente, lo que puede ser perjudicial para la búsqueda de soluciones óptimas [61, 64].

Selección ordinal

La selección ordinal incluye métodos como: selección basada en ranking, selección por truncamiento y selección por torneo. [58, 61, 64].

- **Selección basada en ranking.** En este tipo de selección ordinal, los individuos son clasificados de mejor a peor, y sobre esta clasificación se asigna la probabilidad de selección.
- **Selección por truncamiento.** Los individuos de la población son ordenados de mejor a peor de acuerdo a su valor de ajuste. Una vez ordenados, los primeros n individuos son seleccionados para ser utilizados en el operador de cruce.
- **Selección por torneo.** Consiste en llevar a cabo torneos entre pequeños subconjuntos de individuos de la población. En cada torneo, se seleccionan aleatoriamente k individuos de la población, siendo k el tamaño del torneo. De este grupo, se elige al individuo con el mejor valor de aptitud para participar en el cruce.

2.2.1.4. Cruce

El cruce, también conocido como recombinación, es el proceso por el cual se intercambia información entre dos o más individuos padres para generar una nueva solución o descendencia. Este proceso asegura que la descendencia adquiera características de sus padres mediante la recombinación de componentes seleccionados [64]. La probabilidad de cruce $P_c \in [0, 1]$ controla si un individuo padre forma parte de la siguiente generación, permitiendo que individuos de diversas generaciones coexistan. Si $P_c = 1$, la siguiente generación se conforma de la descendencia producida durante el cruce; si $P_c = 0$, la descendencia es igual a la

generación actual. En caso de $0 \leq P_c \leq 1$, algunos padres pueden sobrevivir a la siguiente generación [60].

Para la representación real o de punto flotante, existen varios operadores de cruce. Algunos de estos operadores son adaptados de la representación binaria y se realizan directamente sobre los vectores de parámetros de los individuos padres, en lugar de sobre cadenas de bits. Ejemplos de operadores de cruce provenientes de la representación binaria y adoptados por la representación real son el cruce de un punto y el cruce multi-punto. Además, se pueden utilizar el cruce uniforme y el cruce aritmético para generar una nueva descendencia bajo la representación real [64]. A continuación, se proporciona una breve revisión de estos operadores de cruce utilizados en la representación real.

Cruce de un punto y multi-punto

El cruce de un punto implica la selección aleatoria de un punto de cruce entre dos individuos padre, a partir del cual se intercambia la información de sus vectores de parámetros. En la figura Figura 2.2.7(a) se muestra un ejemplo de cruce con un solo punto, donde el punto de cruce está ubicado en el segundo parámetro de los individuos padre $\mathbf{x}_1 = (1.1, 1.2, 1.3, 1.4, 1.5, 1.6)$ y $\mathbf{x}_2 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6)$, generando una nueva descendencia. Por otro lado, el cruce multi-punto se logra al seleccionar n puntos de cruce aleatorios ($n > 1$), intercambiando la información confinada entre cada par de puntos de cruce entre los individuos padre. En la figura Figura 2.2.7(b) se ilustra un ejemplo de cruce multi-punto con $n = 2$, donde los individuos padre $\mathbf{x}_1 = (1.1, 1.2, 1.3, 1.4, 1.5, 1.6)$ y $\mathbf{x}_2 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6)$ intercambian la información confinada por los puntos de cruce ubicados en el segundo y cuarto parámetro, generando una nueva descendencia.

Cruce uniforme

El cruce uniforme consiste en el intercambio de información entre individuos padre en n posiciones seleccionadas de forma aleatoria. A diferencia del cruce de un solo punto o multi-punto, el intercambio de información se realiza directamente

sobre el parámetro seleccionado y no sobre la información posterior al punto seleccionado o sobre la información confinada entre dos puntos seleccionados. En la Figura 2.2.8 se presenta un ejemplo de cruce uniforme, en donde la selección de aleatoria de los parámetros: x_1, x_3, x_5 y x_6 en intercambiada de forma directa entre los individuos padre $\mathbf{x}_1 = (1.1, 1.2, 1.3, 1.4, 1.5, 1.6)$ y $\mathbf{x}_2 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6)$, para producir nueva descendencia.

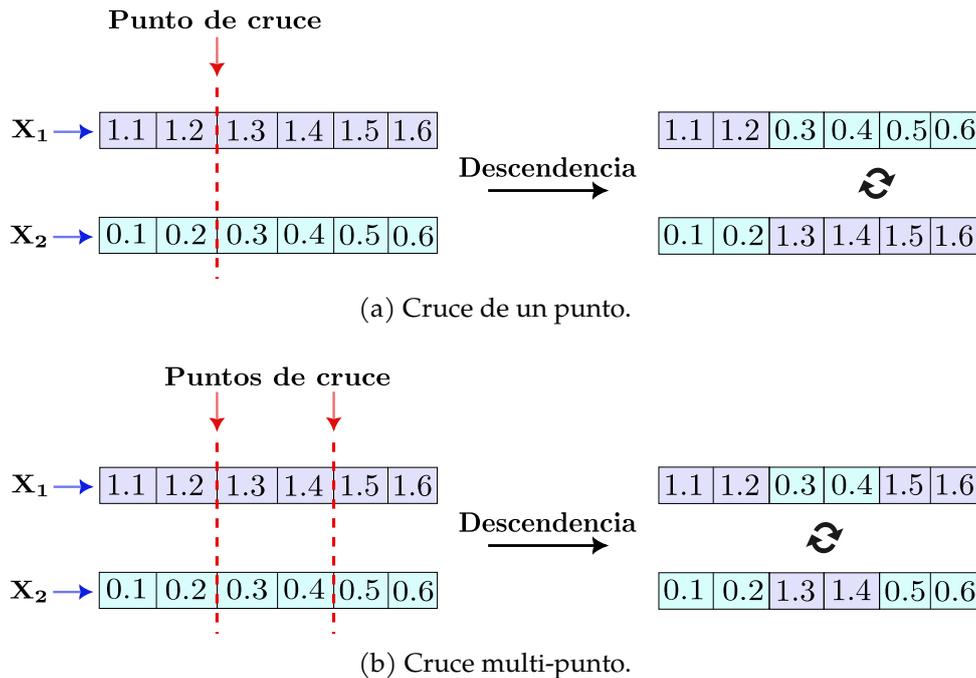


Figura 2.2.7: Ejemplos de operadores de cruce provenientes de la representación binaria y adoptados por la representación real.

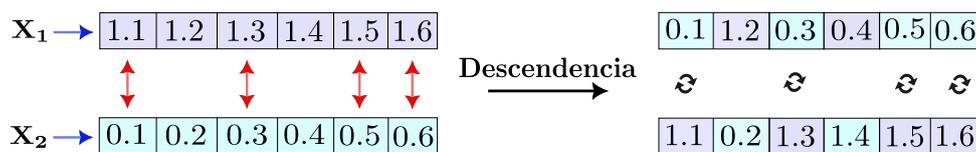


Figura 2.2.8: Ejemplo de cruce uniforme.

Cruce aritmético

Dentro del cruce aritmético, se genera una nueva descendencia mediante el cálculo del promedio en posiciones específicas de los vectores de parámetros de dos individuos padre. El cruce aritmético puede ser simple, completo o de una sola posición, dependiendo de la selección aleatoria del punto de cruce [58]. En el cruce aritmético simple, se selecciona un punto de cruce y se calculan los promedios

de las posiciones restantes de los vectores de parámetros. Por ejemplo, en la Figura 2.2.9(a), se muestra un cruce aritmético simple donde se selecciona el cuarto parámetro de los individuos padre x_1 y x_2 . En el cruce aritmético de una sola posición, se selecciona solo una posición dentro de los vectores de parámetros y se calcula el promedio por pares de los parámetros de los individuos padre en esa posición. La Figura 2.2.9(b) muestra un ejemplo de cruce aritmético de una sola posición. Finalmente, el cruce aritmético completo implica tomar los promedios de ambos padres a lo largo de todos sus parámetros. Este es uno de los operadores de cruce más utilizados y se muestra un ejemplo en la Figura 2.2.9(c) para los individuos padre x_1 y x_2 .

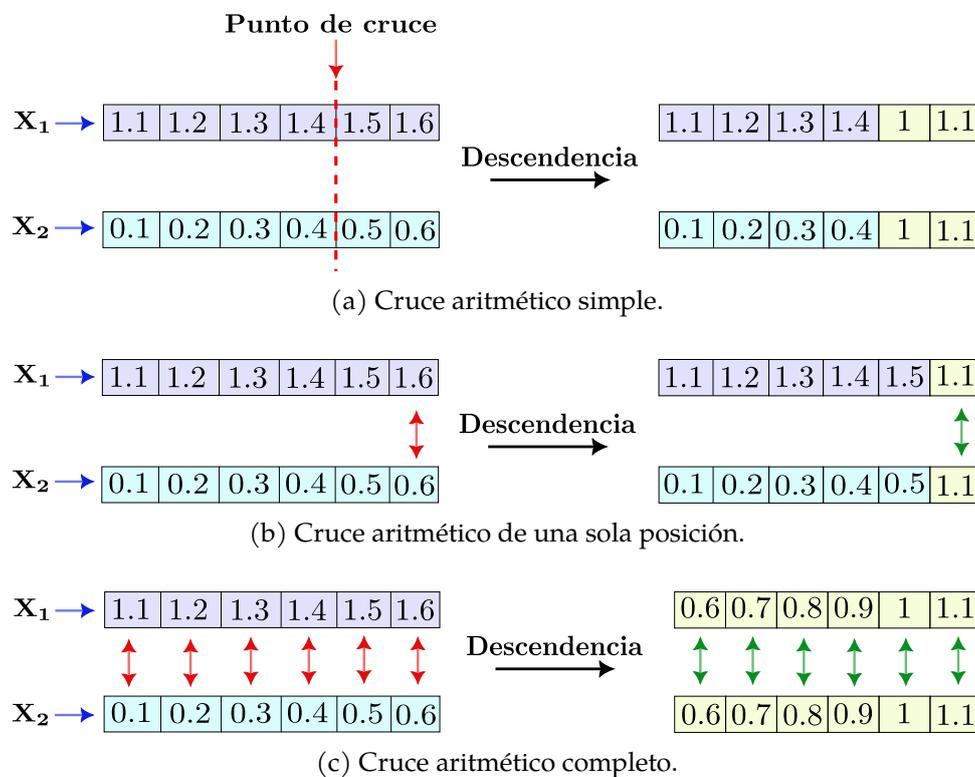


Figura 2.2.9: Ejemplos de cruce aritmético: simple, de una sola posición y completo.

2.2.1.5. Mutación

La mutación es un proceso que opera a nivel individual, realizando pequeñas modificaciones aleatorias en el vector de parámetros de una posible solución. Estas modificaciones son útiles para escapar de mínimos locales y preservar la diversidad de la población, algo que el operador de cruce no puede garantizar [64, 65]. En

representaciones con valores reales, existen varias formas de crear nuevos descendientes o modificar soluciones provenientes de un cruce. Dos métodos destacados son la mutación uniforme, que cambia los parámetros por valores aleatorios dentro de sus límites, y la mutación no uniforme o Gaussiana [58]. A continuación, se describen estos operadores de mutación en mayor detalle.

Mutación uniforme

El operador de mutación uniforme modifica los valores de los parámetros de una posible solución $\mathbf{x} = (x_1, x_2, \dots, x_n)$ mediante valores aleatorios uniformemente generados dentro de su dominio o entre sus constricciones de frontera. Aunque es uno de los métodos más directos y fáciles de implementar, su uso requiere una probabilidad de mutación p_m para decidir si se lleva a cabo o no la modificación de cierto parámetro [58, 64].

Mutación no uniforme o Gaussiana

La mutación Gaussiana es una de las técnicas más utilizadas en la representación real de individuos en algoritmos evolutivos [65]. Este operador introduce pequeños cambios al sumar un vector de valores aleatorios generados a partir de una distribución Gaussiana con media cero y desviación estándar definida por el usuario al individuo a mutar. La distribución Gaussiana (ver Ecuación (2.2.2)) tiene la propiedad de que la probabilidad de generar un número aleatorio de cualquier tamaño es una función que decrece rápidamente en función de la desviación estándar σ . Esto significa que la mayoría de los cambios serán pequeños, pero existe la posibilidad de obtener grandes cambios debido a que la distribución nunca alcanza un valor exacto de cero. El parámetro σ se conoce como el tamaño de mutación y determina la extensión de las perturbaciones para un valor x_i dado al aplicar el operador de mutación. Valores grandes de σ pueden ser útiles al principio del proceso de optimización, ya que permiten una exploración más amplia del espacio de búsqueda. Por el contrario, valores pequeños de σ se recomiendan en las últimas etapas para mejorar la precisión en la búsqueda del óptimo [58, 65].

$$p(\Delta x_i) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(\Delta x_i - \xi)^2}{2\sigma^2}} \quad (2.2.2)$$

El operador de mutación no uniforme produce un vector de parámetros mutado de la siguiente forma:

$$\mathbf{m} = \mathbf{x} + \mathcal{N}(0, \sigma), \quad (2.2.3)$$

donde \mathbf{m} es el vector de parámetros mutado, \mathbf{x} es el individuo a ser mutado y $\mathcal{N}(0, \sigma)$ es un vector del mismo tamaño de \mathbf{x} con valores aleatorios pertenecientes a una distribución Gaussiana con media cero y desviación estándar σ .

Una alternativa a la distribución Gaussiana es hacer uso de la distribución de Cauchy. Con esta distribución las probabilidades de obtener valores más grandes son más altas en comparación de la distribución Gaussiana, teniendo en cuenta la misma desviación estándar [58].

2.2.1.6. Reemplazo

El reemplazo, también conocido como selección de supervivientes, es un mecanismo que, similar al proceso de selección de individuos padres, evalúa la calidad de los individuos de una población para conformar una nueva población μ [58]. Después de la creación de descendencia, los mecanismos de reemplazo se utilizan para seleccionar a los individuos que compondrán la próxima generación, a partir de μ padres y una descendencia λ [61]. Estos mecanismos pueden variar según el valor de ajuste o la edad de los individuos. A continuación, se presentan algunas estrategias más comunes para llevar a cabo el proceso de reemplazo clasificadas según estos criterios.

Reemplazo basado en edad

En este tipo de estrategias de reemplazo, el valor de ajuste de los individuos no influye en la selección de los individuos que sobrevivirán en futuras generaciones o iteraciones. Entre estas estrategias de reemplazo se tiene al reemplazo generacional y el reemplazo aleatorio.

- **Reemplazo generacional.** El método de reemplazo generacional es uno de los más simples y mayormente adoptados dentro de la estructura canónica

los algoritmos genéticos [61]. En cada generación la población de padres μ es reemplazada por la descendencia λ . Con lo cual, el tamaño de la población de padres que sobreviven para cada una de las siguientes generaciones se mantiene constante, es decir, $\mu = \lambda$ [64, 65].

- **Reemplazo aleatorio.** La nueva población es generada a partir de la selección aleatoria de individuos entre la población de padres μ y su descendencia λ . El tamaño de la nueva población debe corresponder al tamaño de la población de individuos padre μ [61].

Reemplazo basado en el valor de ajuste

En estas estrategias de reemplazo, el valor de ajuste obtenido por los individuos padre, así como el obtenido por su descendencia, juega un rol importante al momento de seleccionar a los individuos que sobrevivirán dentro de futuras generaciones.

- **Reemplazo de los peores individuos.** En esta estrategia de reemplazo, los peores individuos de la población λ son seleccionados para ser reemplazados.
- **Reemplazo por torneo.** Involucra a un individuo padre y su descendencia. El que tenga el mejor valor de ajuste es seleccionado para permanecer en la población [61, 64].
- **Elitismo.** Se mantienen a los mejores n individuos de la población obtenidos en la generación anterior. Métodos como el elitismo mejoran considerablemente el rendimiento de los algoritmos evolutivos cuando son aplicados en funciones objetivo convexas, sin embargo, el rendimiento puede verse afectado en funciones objetivo multimodales debido a que el elitismo incrementa la presión de selección en los mejores individuos de la población [65].
- **Reemplazo (μ, λ) -ES.** Mediante una selección truncada de los mejores μ individuos en la descendencia λ es como se crea la nueva población. Para este reemplazo se debe cumplir que $\lambda \geq \mu$ [65].
- **Reemplazo $(\mu + \lambda)$ -ES.** La nueva población es generada a partir de un proceso de selección truncada entre la unión de los individuos que conforman la

descendencia λ y a la población de individuos padre μ . Este tipo de reemplazo puede ser considerado como elitista, ya que mantiene al mejor individuo de la población [65]. Los reemplazos (μ, λ) -ES y $(\mu + \lambda)$ -ES fueron presentados originalmente dentro de las estrategias evolutivas.

2.2.1.7. Criterio de parada

En los algoritmos evolutivos, los operadores de evolución, como el cruce y la mutación, se aplican iterativamente sobre la población de individuos hasta que se cumpla un criterio de parada (ver Algoritmo 2.2.1). Los criterios de parada se clasifican en dos categorías: aquellos basados en el tiempo de ejecución y aquellos basados en la convergencia [24]. A continuación, se presentan los principales criterios de parada de cada tipo.

Criterios de parada basados en el tiempo de ejecución

Una opción para detener la ejecución de un algoritmo evolutivo sin considerar la calidad de la solución encontrada, consiste en limitar su tiempo de ejecución. En este sentido, se tienen los siguientes criterios de parada:

- **Número máximo de iteraciones (generaciones).** Si el algoritmo supera un número máximo de iteraciones permitidas a ejecutar, entonces el algoritmo detiene su ejecución.
- **Número máximo de evaluaciones.** Similar al criterio de parada mencionado anteriormente, si el algoritmo supera un número máximo permitido de evaluaciones sobre la función objetivo, entonces el algoritmo detiene su ejecución.

Criterios de parada basados en la convergencia

Al evaluar la calidad de la solución o la convergencia de la población, se pueden considerar otros criterios de parada más allá de los basados en el tiempo de ejecución. La convergencia se define como un estado de estancamiento en la población [24]. Algunos criterios de parada basados en convergencia que se pueden utilizar son:

- **Terminar cuando no se observan mejoras en un número consecutivo de iteraciones.** Si en un número determinado de iteraciones el valor de ajuste de la mejor solución encontrada no mejora, entonces el algoritmo es detenido.
- **Terminar si no hay cambio en la población.** Si en un número determinado de iteraciones el cambio promedio en la información de cada uno de los parámetros de los individuos de la población es pequeño, entonces el algoritmo evolutivo puede ser detenido.
- **Terminar si se ha encontrado un solución con una calidad aceptable.** Si \mathbf{x}^* representa la solución global de la función objetivo, y si el mejor individuo \mathbf{x}_i es tal que $f(\mathbf{x}_i) \leq |f(\mathbf{x}) - \epsilon|$, entonces se tiene una solución con una calidad aceptable. En este caso ϵ representa un umbral de error. Si ϵ es muy grande, se tendrán malas soluciones. Si ϵ es muy pequeño, puede provocar que la ejecución del algoritmo se prolongue indefinidamente si no se restringe su tiempo de ejecución.

2.2.2. Evolución Diferencial

Uno de los algoritmos evolutivos más destacados es el Algoritmo de Evolución Diferencial (DE) (DE, por sus siglas en inglés), propuesto por Storn y Price en 1995 para abordar problemas de optimización global en espacios continuos [66]. Aunque no se basa en un proceso biológico específico, utiliza los mecanismos comunes de los algoritmos evolutivos [61]. Debido a su relevancia en este trabajo, se describe con más detalle a continuación.

El algoritmo DE sigue las etapas generales de un algoritmo evolutivo, pero con una diferencia clave en el orden de los operadores de cruce y mutación. El funcionamiento general del algoritmo DE se describe de la siguiente manera: se genera una población μ de NP soluciones candidatas, representadas por vectores de N dimensiones con valores reales dentro del espacio de búsqueda, y luego se aplica un operador de mutación para crear un vector mutado $\mathbf{v}_{i,G}$, el cual se cruza con la información de la solución candidata $\mathbf{x}_{i,G}$ ($i \in [1, NP]$) en la generación actual G , para producir un vector prueba $\mathbf{u}_{i,G}$. Si el valor de ajuste del vector de prueba es mejor que el de la solución candidata, entonces el vector de prueba reemplaza a la

solución candidata [60, 67]. Este proceso se repite iterativamente hasta satisfacer un criterio de parada. A continuación se detallan los operadores de mutación y cruce implementados en este algoritmo.

2.2.2.1. Mutación

El operador de mutación del algoritmo DE tiene el objetivo de permitir la diversidad en la población y desplazar a las soluciones candidatas hacia el óptimo global del problema [57]. En este operador, para cada solución candidata $\mathbf{x}_{i,G}$ de la población, se seleccionan de forma aleatoria a otras tres soluciones candidatas de la población $\mathbf{x}_{r1,G}$, $\mathbf{x}_{r2,G}$ y $\mathbf{x}_{r3,G}$, diferentes entre sí y diferentes a $\mathbf{x}_{i,G}$. Luego, a partir de la suma entre el vector $\mathbf{x}_{r1,G}$ y la diferencia vectorial entre $\mathbf{x}_{r2,G}$ y $\mathbf{x}_{r3,G}$ multiplicada por un factor de escala $F \in [0.4, 0.9]$, se crea un vector mutado para la solución candidata $\mathbf{x}_{i,G}$. Esta forma de generar un vector mutado para una posible solución se puede expresar de la siguiente forma:

$$\mathbf{v}_{i,G} = \mathbf{x}_{r1,G} + F(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}). \quad (2.2.4)$$

La Figura 2.2.10 ilustra el procedimiento de mutación para generar el vector $\mathbf{v}_{i,G}$. En este esquema bidimensional, tres soluciones candidatas aleatorias se representan con marcas amarillas, naranjas y azules. La marca magenta denota la solución candidata. El vector azul, resultado de la diferencia vectorial entre $\mathbf{x}_{r2,G}$ y $\mathbf{x}_{r3,G}$ escalada por F , se suma a $\mathbf{x}_{r1,G}$ para producir el vector mutado (en verde), cuya posición se indica con una marca roja.

2.2.2.2. Cruce

Después de obtener el vector mutado $\mathbf{v}_{i,G}$, la siguiente etapa dentro del algoritmo DE es la construcción de un vector de prueba $\mathbf{u}_{i,G}$ a través del cruce de información de la solución candidata $\mathbf{x}_{i,G}$ y el vector mutado $\mathbf{v}_{i,G}$. Cada variable del vector de prueba se obtiene por medio de la selección del valor de una variable $j \in [1, N]$ ya sea del vector objetivo o del vector mutado. Esta selección se realiza de la siguiente manera [61, 68]:

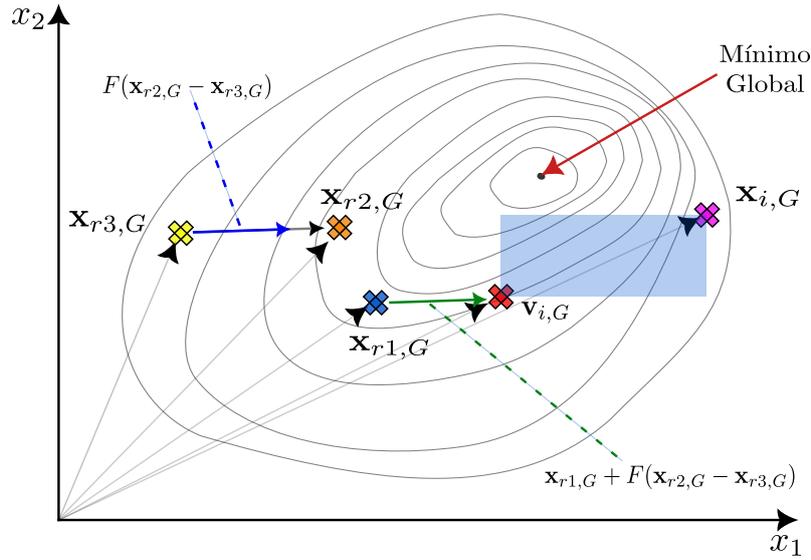


Figura 2.2.10: Procedimiento para obtener el vector mutado $\mathbf{v}_{i,G}$ a través del operador de mutación.

$$\mathbf{u}_{i,G}^j = \begin{cases} \mathbf{v}_{i,G}^j, & \text{si } (r_j \leq C_r) \text{ o } (j = i_{\text{rand}}) \\ \mathbf{x}_{i,G}^j, & \text{si } (r_j > C_r) \text{ y } (j \neq i_{\text{rand}}) \end{cases}, \quad (2.2.5)$$

donde r_j es un valor aleatorio uniformemente distribuido en el intervalo $[0, 1]$ y i_{rand} un valor entero en $[1, N]$. El valor de i_{rand} es utilizado para garantizar que el vector de prueba tenga al menos una variable del vector mutado. C_r es el valor asociado a la probabilidad de cruce, y toma valores entre 0 y 1.

La probabilidad de cruce C_r al igual que el factor de escala F y el número de individuos que conformarán a la población inicial NP , son los únicos parámetros que requieren ser ajustados dentro de la versión canónica del algoritmo de evolución diferencial. Cabe resaltar que este tipo de cruce es conocido como cruce binomial.

En la Figura 2.2.10 la región iluminada de color azul claro representa el área en la cual el vector de prueba $\mathbf{u}_{i,G}$ puede ser generado a partir de la información del vector mutado $\mathbf{v}_{i,G}$ y la solución candidata $\mathbf{x}_{i,G}$.

2.2.2.3. Reemplazo

La siguiente etapa del algoritmo de evolución diferencial implica la selección de supervivientes para la próxima generación. En este proceso, se utiliza un esquema

de competición uno a uno entre la solución candidata $\mathbf{x}_{i,G}$ y el vector de prueba $\mathbf{u}_{i,G}$. El vector con el mejor valor de ajuste sobrevive y es seleccionado para permanecer en la población en la próxima generación.

Teniendo en cuenta todos los componentes y etapas descritas anteriormente, el proceso completo del algoritmo DE se muestra en el pseudocódigo del Algoritmo 2.2.2.

Algoritmo 2.2.2: Algoritmo de Evolución Diferencial Canónico.	
1:	Inicializar parámetros del algoritmo
2:	$F =$ Factor de escala $\in [0.4, 0.9]$
3:	$C_r =$ Probabilidad de cruce $\in [0, 1]$
4:	$NP =$ Numero de individuos en la población
5:	$G = 0$
6:	$\mu_G \leftarrow$ Inicializar de forma aleatoria una población con NP soluciones candidatas
7:	mientras no se cumple el criterio de parada hacer
8:	para cada solución candidata $\mathbf{x}_{i,G} \in \mu_G, i = 1, 2, \dots, NP$ hacer
9:	Seleccionar de forma aleatoria $\mathbf{x}_{r1,G}, \mathbf{x}_{r2,G}$ y $\mathbf{x}_{r3,G} \in \mu_G$, tales que $r1 \neq r2 \neq r3 \neq i$
10:	$\mathbf{v}_{i,G} \leftarrow$ Generar el vector mutado usando la Ecuación (2.2.4)
11:	$\mathbf{u}_{i,G} \leftarrow$ Cruzar $\mathbf{x}_{i,G}$ y $\mathbf{x}_{m,G}$ para generar el vector de prueba
12:	Evaluar $\mathbf{u}_{i,G}$ y $\mathbf{x}_{i,G}$
13:	si $\mathbf{u}_{i,G}$ tiene mejor valor de ajuste que $\mathbf{x}_{i,G}$ entonces
14:	$\mathbf{x}_{i,G+1} = \mathbf{u}_{i,G}$
15:	en otro caso
16:	$\mathbf{x}_{i,G+1} = \mathbf{x}_{i,G}$
17:	fin
18:	fin
19:	fin
20:	$G = G + 1$
21:	fin

2.2.2.4. Variantes del Algoritmo de Evolución Diferencial

Existen diversas variaciones del operador de mutación y del esquema de cruce en el algoritmo de evolución diferencial, lo que da lugar a diferentes variantes del algoritmo. Estas variaciones se reflejan en la selección del vector base, el número de diferencias vectoriales utilizadas y la forma de realizar el cruce [58, 61]. Para identificar estas variaciones, se utiliza la notación $DE/a/b/c$, donde a especifica la

selección del vector base (aleatorio, mejor individuo o solución, o un vector entre la solución candidata y la mejor solución), b denota el número de diferencias vectoriales (normalmente una o dos), y c la forma de cruce (binomial o exponencial) [58, 61]. La variante utilizada en las secciones anteriores se puede representar como $DE/rand/1/bin$, que selecciona el vector base $\mathbf{x}_{r1,G}$ aleatoriamente, utiliza una diferencia vectorial ($\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}$) y aplica el cruce binomial (ver Ecuación (2.2.5)). Esta variante es la forma canónica del algoritmo de evolución diferencial y una de las más utilizadas [61, 66].

El algoritmo DE ofrece diferentes variantes que no especifican explícitamente el operador de cruce. Sin embargo, se asume generalmente el uso del cruce binomial. Dos ejemplos de estas variantes son $DE/best/1$, que utiliza el mejor individuo de la población como vector base, y $DE/rand/2$, que combina dos diferencias vectoriales para aumentar la diversidad. Estas variantes se implementan según las ecuaciones Ecuaciones (2.2.6) a (2.2.7) y Ecuaciones (2.2.6) a (2.2.10) respectivamente.

- $DE/rand/1$:

$$\mathbf{v}_{i,G} = \mathbf{x}_{r1,G} + F \cdot (\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}) \quad (2.2.6)$$

- $DE/rand/2$:

$$\mathbf{v}_{i,G} = \mathbf{x}_{r1,G} + F \cdot (\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}) + F \cdot (\mathbf{x}_{r4,G} - \mathbf{x}_{r5,G}) \quad (2.2.7)$$

- $DE/best/1$:

$$\mathbf{v}_{i,G} = \mathbf{x}_{best,G} + F \cdot (\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}) \quad (2.2.8)$$

- $DE/best/2$:

$$\mathbf{v}_{i,G} = \mathbf{x}_{best,G} + F \cdot (\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}) + F \cdot (\mathbf{x}_{r3,G} - \mathbf{x}_{r4,G}) \quad (2.2.9)$$

- $DE/current-to-best/1$:

$$\mathbf{v}_{i,G} = \mathbf{x}_{i,G} + F \cdot (\mathbf{x}_{best,G} - \mathbf{x}_{i,G}) + F \cdot (\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}) \quad (2.2.10)$$

donde los índices r_1, \dots, r_5 son seleccionados de forma aleatoria dentro del rango $[1, NP]$ tales que sean diferentes entre sí y distintos al índice i . $\mathbf{x}_{best,G}$ es la mejor solución dentro de la población en la generación G , y F es el factor de escala.

Además de la selección del operador de mutación (o estrategia de mutación de acuerdo a la notación $DE/a/b$), dentro de la literatura se cuenta con otro tipo de variantes las cuales tratan de mejorar el rendimiento de la forma canónica del algoritmo DE. En estas variantes se busca principalmente: mejorar el control de los parámetros F y C_r , realizar hibridaciones del algoritmo DE con otros algoritmos de optimización, mejorar las estrategias de mutación existentes, así como crear nuevas.

2.2.3. Algoritmos Basados en Inteligencia de Enjambre

La Inteligencia de Enjambre (IE) y los AE son paradigmas dentro de la IC que se basan en modelos algorítmicos de agentes o individuos que interactúan entre sí en un entorno local. Estos modelos permiten a los individuos desarrollar estrategias para resolver problemas complejos a través de la interacción [24]. En este contexto, el grupo de individuos se conoce como un enjambre. La IE se inspira en sistemas biológicos como bandadas de aves, bancos de peces, abejas, arañas y hormigas, que a pesar de tener estructuras simples, logran comportamientos complejos a través de la colaboración [24, 69]. Este comportamiento se desarrolla a través de interacciones entre individuos a lo largo del tiempo, permitiendo a los individuos aprender de su propia experiencia y de la de los demás, lo que les permite resolver conjuntamente tareas complejas que serían imposibles para cualquier individuo de la población de manera individual [61].

La cooperación dentro de la inteligencia de enjambre es importante, ya que los individuos pueden resolver tareas complejas a pesar de tener información limitada sobre su entorno [61]. Esto se logra a través de interacciones directas, como contacto físico y estímulos auditivos, visuales o químicos, y indirectas, que se producen a través de cambios en el entorno.

El objetivo de los modelos de inteligencia de enjambres es simular el comportamiento de individuos simples y sus interacciones con su entorno y otros indivi-

duos cercanos para generar comportamientos complejos que puedan ser utilizados para resolver problemas, y en particular problemas de optimización.

El algoritmo de inteligencia de enjambre sigue un patrón de etapas fundamentales. La primera etapa consiste en inicializar aleatoriamente una población de individuos o agentes en el espacio de búsqueda. En la segunda etapa, se evalúa el ajuste de cada individuo. Luego, se verifica si se ha cumplido un criterio de parada. Si se cumple, el algoritmo termina y se regresa la mejor solución encontrada. De lo contrario, los individuos se actualizan y se mueven a una nueva posición [70]. La inteligencia de enjambre es una clase popular de algoritmos de optimización debido a su eficacia en resolver problemas de la vida real [69, 71]. En el Algoritmo 2.2.3 se muestra en forma de pseudocódigo el esquema general que sigue un algoritmo de inteligencia de enjambre.

Algoritmo 2.2.3: Esquema general de un Algoritmo de Inteligencia de Enjambre en pseudocódigo.

```

1: Inicializar de forma aleatoria una población de posibles soluciones
2: mientras no se cumpla el criterio de parada hacer
3:   Evaluar cada solución en la función objetivo
4:   Actualizar individuos
5:   Mover individuos
6: fin

```

Uno de los algoritmos de optimización más destacados dentro de la inteligencia de enjambre, y especialmente relevante en el contexto de esta investigación, se describe a continuación.

2.2.4. Optimización por Enjambre de Partículas

El Algoritmo de Optimización por Enjambre de Partículas (PSO) (PSO, por sus siglas en inglés) fue introducido por primera vez en 1995 por Kennedy y Eberhart [72]. Destaca entre los algoritmos de inteligencia de enjambre por su simplicidad y capacidad de resolver diversos problemas de optimización [73]. Basado en una población de individuos, PSO imita el comportamiento social de bandadas de aves o bancos de peces, emulando habilidades como el movimiento sincronizado, el

cambio repentino de direcciones, la dispersión y el reagrupamiento. El objetivo principal del desarrollo del PSO ha sido representar modelos de interacción social y optimizar espacios continuos [61, 74].

PSO utiliza un conjunto de NP soluciones candidatas generadas aleatoriamente dentro del espacio de búsqueda, conocidas como enjambre. Cada solución candidata, o partícula, se representa mediante un vector de N componentes $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iN})$, $i \in [1, NP]$. El proceso de búsqueda se ve influenciado por dos tipos de aprendizaje: el aprendizaje cognitivo, que permite que cada partícula aprenda de su propia experiencia y almacene la mejor posición encontrada por sí misma $\mathbf{x}_{ipbest,G}$ dentro de la iteración G , y el aprendizaje social, que permite que cada partícula aprenda de los demás miembros del enjambre y almacene la mejor posición encontrada por cualquiera de los miembros del enjambre $\mathbf{X}_{gbest,G}$ [69]. Cada partícula tiene asociadas dos propiedades: su posición actual $\mathbf{x}_{i,G}$ y su velocidad actual $\mathbf{v}_{i,G}$, y la actualización de estas propiedades dentro del espacio de búsqueda se realiza mediante la modificación de estas propiedades, con el fin de encontrar el óptimo global.

La velocidad desempeña un papel importante en PSO, ya que determina el cambio en la magnitud y dirección de las partículas y afecta directamente a la capacidad de convergencia. En este contexto, la velocidad se refiere al cambio en la posición respecto a una iteración [69]. Desde su creación, el PSO ha experimentado variaciones en la forma de actualizar la velocidad de las partículas [72, 74, 75], aunque una forma estándar de hacerlo, se realiza de la siguiente manera [75]:

$$\mathbf{v}_{i,G+1} = \mathbf{v}_{i,G} + c_1 r_1 (\mathbf{x}_{ipbest,G} - \mathbf{x}_{i,G}) + c_2 r_2 (\mathbf{X}_{gbest,G} - \mathbf{x}_{i,G}), \quad (2.2.11)$$

donde $\mathbf{v}_{i,G+1}$ es la velocidad actualizada correspondiente a la i -ésima partícula $\mathbf{x}_{i,G}$, $\mathbf{v}_{i,G}$ es la velocidad actual de la partícula, $\mathbf{x}_{ipbest,G}$ representa la mejor posición encontrada por $\mathbf{x}_{i,G}$, $\mathbf{X}_{gbest,G}$ es la mejor posición encontrada por todo el enjambre, c_1 y c_2 son dos constantes positivas conocidas parámetros de escalamiento cognitivo y social respectivamente, r_1 y r_2 son dos valores aleatorios uniformemente distribuidos en el intervalo $[0, 1]$. En esta ecuación, el segundo término es considerado como la parte cognitiva, y representa la pensamiento individual de la

partícula. El tercer término de la ecuación, es la parte social, el cual representa la colaboración entre todas las partículas del enjambre [68, 76].

La forma de actualizar la posición de una partícula, se puede ver como la adición de un vector de perturbación a la posición actual de la partícula dentro del espacio de búsqueda [74]. Este vector de perturbación está determinado por la nueva velocidad de la partícula $\mathbf{v}_{i,G+1}$. De forma iterativa la posición de cada partícula se actualizará de la siguiente forma [75]:

$$\mathbf{x}_{i,G+1} = \mathbf{x}_{i,G} + \mathbf{v}_{i,G+1}, \quad (2.2.12)$$

donde $\mathbf{x}_{i,G+1}$ representa la nueva posición de la i -ésima partícula, $\mathbf{x}_{i,G}$ la posición actual de la partícula y $\mathbf{v}_{i,G+1}$ la nueva velocidad de la partícula.

La Figura 2.2.11 ilustra el proceso de actualización de la posición de una partícula en el algoritmo PSO mediante un esquema en dos dimensiones. La partícula, inicialmente en $\mathbf{x}_{i,G}$ (marca color azul), con velocidad $\mathbf{v}_{i,G}$ (vector color negro), actualiza su posición sumando los vectores resultantes del segundo y tercer término de la Ecuación (2.2.11), representados por los vectores verde y rojo respectivamente. La suma de estos vectores (vector punteado color azul) determina la nueva posición de la partícula.

De acuerdo a lo presentado anteriormente, el proceso completo del algoritmo de optimización por enjambre de partículas muestra mediante el pseudocódigo del Algoritmo 2.2.4. El algoritmo de optimización por enjambre de partículas (PSO) comienza generando un enjambre de partículas de forma aleatoria μ_G , cada una con una posición, velocidad y mejor posición encontrada. En la primera iteración, se establece la mejor posición de cada partícula como su posición actual y se inicializan las velocidades a cero. Luego, se determina la partícula con la mejor posición dentro del enjambre. A continuación, se actualizan las velocidades y posiciones de las partículas mediante las Ecuaciones (2.2.11) a (2.2.12). Si la nueva posición tiene un ajuste mejor que la mejor posición encontrada hasta el momento, se reemplaza. Si es mejor que la mejor posición del enjambre, se reemplaza también. Este proceso se repite hasta que se cumpla un criterio de parada.

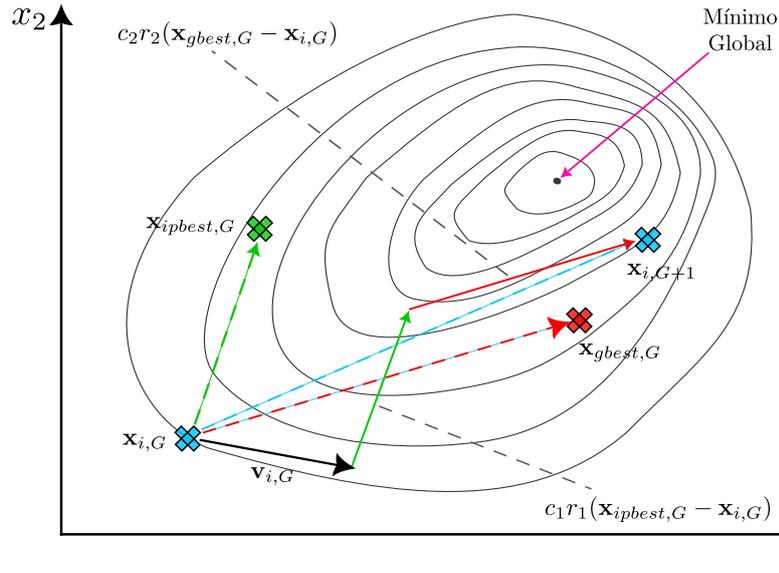


Figura 2.2.11: Esquema del proceso de actualización de la posición de una partícula en dos dimensiones.

Algoritmo 2.2.4: Algoritmo de Optimización por Enjambre de Partículas Canónico.

- 1: Inicializar parámetros del algoritmo
- 2: c_1 = Parámetro de escala cognitivo
- 3: c_2 = Parámetro de escala social
- 4: NP = Numero de individuos en el enjambre
- 5: $G = 0$
- 6: $\mu_G \leftarrow$ Inicializar un enjambre con NP partículas de forma aleatoria
- 7: $v_{i,G} \leftarrow$ Inicializar las velocidades de las partículas
- 8: $x_{pbest,G} = \mu_G \leftarrow$ Inicializar las mejores posiciones de las partículas
- 9: $x_{gbest,G} \leftarrow$ Determinar a la mejor posición de la población
- 10: **mientras** no se cumpla el criterio de parada **hacer**
- 11: **para cada** partícula $x_{i,G} \in \mu_G, i = 1, 2, \dots, NP$ **hacer**
- 12: $v_{i,G+1} \leftarrow$ Actualizar la velocidad usando la Ecuación (2.2.11)
- 13: $x_{i,G+1} \leftarrow$ Actualizar la posición usando la Ecuación (2.2.12)
- 14: Evaluar $x_{i,G+1}, x_{ipbest,G}$ y $x_{gbest,G}$
- 15: **si** $x_{p,G+1}$ tiene mejor valor de ajuste que $x_{ipbest,G}$ **entonces**
- 16: $x_{ipbest,G} = x_{p,G+1}$
- 17: **fin**
- 18: **si** $x_{p,G+1}$ tiene mejor valor de ajuste que $x_{gbest,G}$ **entonces**
- 19: $x_{gbest,G} = x_{p,G+1}$
- 20: **fin**
- 21: **fin**
- 22: $G = G + 1$
- 23: **fin**

2.2.4.1. Configuración de parámetros

PSO tiene tres parámetros configurables: el tamaño de la población NP , el parámetro de escala cognitivo c_1 y el parámetro de escala social c_2 . No hay una configuración óptima universal, ya que cada parámetro puede depender del problema de optimización. Sin embargo, se ofrecen recomendaciones en la literatura. El tamaño de la población suele variar según la complejidad del problema y el número de variables, recomendándose entre 20 y 50 partículas en general [69]. Los parámetros c_1 y c_2 controlan la velocidad de las partículas. Si $c_1 = c_2 > 0$, las partículas se atraen hacia el promedio entre la mejor posición individual y la mejor posición del enjambre. Para problemas multimodales, $c_1 > c_2$ es recomendable mientras que $c_2 > c_1$ para problemas unimodales. Valores pequeños de c_1 y c_2 generan trayectorias suaves, mientras que valores grandes producen movimientos abruptos [69].

2.2.4.2. Estructura de red social y variaciones

La estructura de la red social en PSO se define mediante vecindarios que se superponen y la influencia entre las partículas dentro de ellos. Las partículas dentro de un vecindario se comunican intercambiando información sobre el éxito de cada partícula en el vecindario. El rendimiento del algoritmo puede verse afectado por la estructura de la red social, que depende de factores como la conectividad entre nodos, el agrupamiento y la distancia entre ellos [24]. Una estructura de red altamente conectada puede generar una mejor convergencia, pero es más propensa a quedar atrapada en mínimos locales. El algoritmo PSO canónico utiliza una estructura de red llamada estrella, en la que cada partícula se comunica con todas las demás y se atrae hacia la mejor solución encontrada. Otros tipos de estructuras de red incluyen anillo, pirámide, rueda, agrupamientos y Von Neumann. Los detalles de estas estructuras se pueden encontrar en [24].

A partir de la estructura canónica del algoritmo PSO, se han realizado variaciones del algoritmo PSO para mejorar su rendimiento. Estas variaciones se centran en controlar la velocidad de las partículas, la estructura de red social, la obtención de los parámetros de escala c_1 y c_2 , y la actualización de la mejor posición de cada

partícula y del enjambre. Las variaciones más destacadas incluyen el algoritmo PSO con factor de inercia y con coeficiente de constricción. A continuación, se presentan brevemente estas variaciones [24, 61, 62].

Shi y Eberhart [76] introdujeron el parámetro de inercia o fricción ω , con el propósito de mejorar el balance entre la búsqueda global y la búsqueda local [61, 68]. En esta variante, el parámetro de inercia impacta sobre la manera de actualizar la velocidad una partícula, concretamente, la forma de actualizar la velocidad de la partícula se realiza de la siguiente manera:

$$\mathbf{v}_{i,G+1} = \omega \mathbf{v}_{i,G} + c_1 r_1 (\mathbf{x}_{ipbest,G} - \mathbf{x}_{i,G}) + c_2 r_2 (\mathbf{x}_{gbest,G} - \mathbf{x}_{i,G}). \quad (2.2.13)$$

Valores altos de ω permiten una mayor exploración inicial, mientras que valores bajos fomentan la explotación en las etapas finales. Comúnmente, se controlan estos valores mediante un mecanismo que combina exploración y explotación en diferentes etapas del proceso de optimización [24, 68]. Esta versión del PSO es ampliamente utilizada en diversas aplicaciones debido a su capacidad para adaptarse a diferentes problemas de optimización [73].

Otra variante que también trata de generar un balance entre la exploración y explotación fue propuesta por Clerc en [77]. En esta variante se introduce al factor de constricción χ , el cual impone una constricción a la velocidad. Con este factor la forma de actualizar la velocidad de una partícula se realiza de la siguiente forma:

$$\mathbf{v}_{i,G+1} = \chi [\mathbf{v}_{i,G} + c_1 r_1 (\mathbf{x}_{ipbest,G} - \mathbf{x}_{i,G}) + c_2 r_2 (\mathbf{x}_{gbest,G} - \mathbf{x}_{i,G})], \quad (2.2.14)$$

con:

$$\chi = \frac{2\kappa}{|2 - \phi - \sqrt{\phi(\phi - 4)}|}, \quad (2.2.15)$$

donde $\phi = c_1 + c_2$, $\phi > 4$ y $\kappa \in [0, 1]$. La selección de estos valores ha sido realizada con el fin de que el algoritmo PSO converja a una pequeña región del espacio de búsqueda [61]. Una selección común de valores del parámetro de constricción

es $\chi = 0.7298$. Lo cual resulta en valores de $c_1 = c_2 = 2.05$ [61]. El parámetro κ en la Ecuación (2.2.15), controla las habilidades de exploración y explotación del enjambre. Valores de $\kappa \approx 0$, generan una convergencia rápida en entornos de búsqueda local. Con valores de $\kappa \approx 1$, se obtiene una convergencia lenta que involucra una mayor exploración [24].

2.3. El Modelo Estándar

El Modelo Estándar de la Física de Partículas (ME) propuesto por Glashow, Salam y Weinberg [78], es una teoría cuántica de campos que describe a las partículas elementales y sus interacciones. En el ME, se distinguen dos categorías principales de partículas: los fermiones, que constituyen la materia, y los bosones, que actúan como mediadores de las fuerzas fundamentales.

Los fermiones están constituidos por partículas elementales llamadas quarks y leptones. Cada una de estas partículas se divide a su vez en dos subsectores: el sector “up” y “down” para los quarks, y el sector de los leptones cargados y los neutrinos para los leptones. En cada uno de estos últimos sectores, los fermiones se estructuran en tres generaciones o familias, donde cada familia comparte el mismo número cuántico, a excepción de su masa, que varía entre ellas [82]. La primer familia está formada con partículas más ligeras, mientras que la segunda y tercer familia tienen partículas más pesadas. En la Figura 2.3.1 se muestran las partículas elementales del ME en forma de una tabla periódica.

Es importante destacar que el término “sabor” en la FAE es utilizado para denominar a cada tipo de quark o lepton [83, 84], y no tiene relación con el sabor en el sentido culinario o sensorial. Con lo cual, se pueden identificar 12 tipos diferentes de sabores: el sector de quarks tienen los sabores u, c, t, d, s y b ; el sector de leptones los correspondientes sabores $e, \mu, \tau, \nu_e, \nu_\mu, \nu_\tau$. Las propiedades tanto de los quarks y leptones se describen respectivamente en las Tablas 2.3.1 y 2.3.3.

Tabla 2.3.1: Propiedades de los Quarks.

Quark	Masa	Carga (Q)	Spin	Familia
Up (u)	2.3 MeV/c ²	$\frac{2}{3}$	$\frac{1}{2}$	I
Down (d)	4.8 MeV/c ²	$-\frac{1}{3}$	$\frac{1}{2}$	I
Charm (c)	1.275 GeV/c ²	$\frac{2}{3}$	$\frac{1}{2}$	II
Strange (s)	95 MeV/c ²	$-\frac{1}{3}$	$\frac{1}{2}$	II
Top (t)	173.07 GeV/c ²	$\frac{2}{3}$	$\frac{1}{2}$	III
Bottom (b)	4.18 GeV/c ²	$-\frac{1}{3}$	$\frac{1}{2}$	III

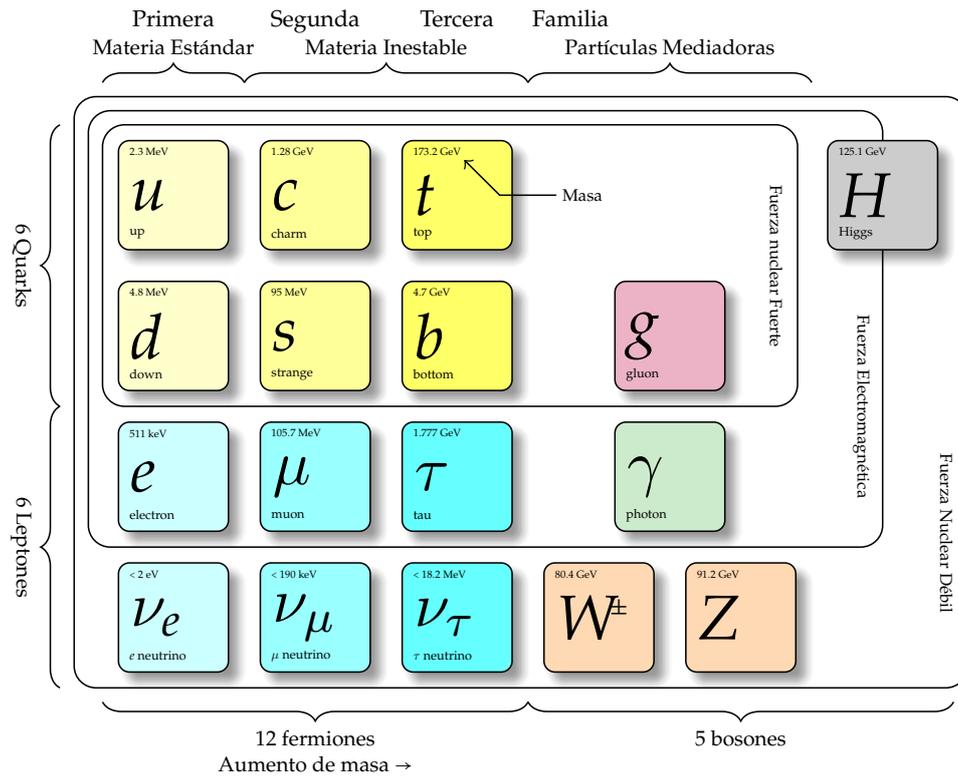


Figura 2.3.1: Tabla de partículas elementales del modelo estándar.

Tabla 2.3.2: Propiedades de los Leptones.

Lepton	Masa	Carga (Q)	Spin	Familia
Electrón (e^-)	$0.511 \text{ MeV}/c^2$	-1	$\frac{1}{2}$	I
Electrón Neutrino (ν_e)	$< 2.2 \text{ MeV}/c^2$	0	$\frac{1}{2}$	I
Muón (μ^-)	$105.7 \text{ MeV}/c^2$	-1	$\frac{1}{2}$	II
Muón Neutrino (ν_μ)	$< 0.17 \text{ GeV}/c^2$	0	$\frac{1}{2}$	II
Tau (τ^-)	$1.777 \text{ GeV}/c^2$	-1	$\frac{1}{2}$	III
Tau Neutrino (ν_τ)	$< 15.5 \text{ GeV}/c^2$	0	$\frac{1}{2}$	III

Por su parte, los bosones se encargan de mediar las interacciones (fuerzas) entre partículas, esto significa que, las partículas interactúan a través del intercambio de un bosón. Cada bosón se encarga de mediar una interacción en particular: El fotón (γ) media la interacción electromagnética entre partículas cargadas eléctricamente; Ocho gluones (g) encargados de mediar las interacciones nucleares fuertes, que mantienen a los hadrones unidos; y los bosones W^\pm y Z median a las interacción nuclear débil, la cual genera la desintegración radioactiva. Además, se tiene al bosón de Higgs (H), el cual es responsable de dotar de masa a otras partículas

elementales. Los bosones de acuerdo a su spin se clasifican en dos tipos: bosones vectoriales (Gauge) y bosones escalares (Higgs). Las propiedades asociadas a los bosones se muestran en la Tabla 2.3.3.

Tabla 2.3.3: Propiedades de los Bosones.

Bosón	Masa	Carga (Q)	Interacción
Bosones vectoriales (Gauge), spin = 1			
Fotón (γ)	0	0	Electromagnética
8 Gluones (g)	0	0	Nuclear Fuerte
$W^{\pm 1}$	$80.4\text{GeV}/c^2$	± 1	Nuclear débil
Z	$91.2\text{GeV}/c^2$	0	Nuclear débil
Bosones escalares (Higgs), spin = 0			
Higgs (H)	$\sim 125\text{ GeV}/c^2$	0	Generación de masa

2.3.1. El problema del sabor

El ME de la FAE es una teoría muy exitosa que describe con gran precisión las interacciones fundamentales de la naturaleza. Sin embargo, a pesar de su éxito, el ME deja algunos problemas sin resolver, especialmente en lo que respecta a la estructura y propiedades de las partículas elementales.

Uno de estos problemas es el “Problema del Sabor”, que se refiere a la falta de una explicación satisfactoria para el patrón observado de masas y mezclas de los fermiones (quarks y leptones). Las masas de los fermiones en el ME se generan a través del mecanismo de Higgs, pero los valores específicos de estas masas, así como la mezcla entre los diferentes sabores de fermiones, no están predichos por la teoría y deben determinarse experimentalmente. El Problema del Sabor surge porque el ME no proporciona una comprensión profunda de por qué las masas y mezclas de los fermiones toman los valores observados.

Las investigaciones sobre la generación de masa de los fermiones y de reproducir los elementos de la matriz V_{CKM} conocida como la matriz de mezcla, constituyen actualmente una parte importante de la física de partículas. Desde los primeros

años de la década de los 70s, poco después del establecimiento del Modelo Estándar, se han desarrollado diferentes enfoques para abordar este problema, que se pueden clasificar en: Mecanismos radiactivos [2, 3], Texturas[4–6], Simetrías entre familias[7, 8], y Mecanismos de Seesaw[9–11]. Estos enfoques están relacionados entre sí.

La presente investigación se centra en el enfoque fenomenológico de texturas. Antes de profundizar en este enfoque, es importante presentar algunos conceptos teóricos relacionados, comenzando con el sector electrodébil del modelo estándar. En este sector, la dinámica de masa de los quarks, que son de especial interés en este estudio, tiene lugar.

2.3.2. Sector electrodébil

El ME es una teoría unificada que describe tres de las cuatro interacciones fundamentales: la interacción electrodébil y la interacción fuerte. Este modelo se basa en una teoría de norma con un grupo de simetría $SU_C(3) \times SU_L(2) \times U_Y(1)$, donde la interacción fuerte está descrita por $SU_C(3)$ y la interacción electrodébil por $SU(2)_L \times U(1)_Y$.

En el sector electrodébil, las partículas fermiónicas interactúan mediante el intercambio de bosones cargados (corrientes cargadas) o neutros (corrientes neutras). La interacción débil distingue entre los estados de helicidad izquierda (dobletes bajo $SU(2)_L$) y derecha (singletes) de las partículas.

Para describir la física del ME, se utiliza un lagrangiano que abarca las interacciones fuertes y electrodébiles. Esta sección, se centrará únicamente en el sector electrodébil del modelo estándar.

El lagrangiano del sector electrodébil, denotado como $\mathcal{L}_{SU(2) \times U(1)}$, se describe por:

$$\mathcal{L}_{SU(2) \times U(1)} = \mathcal{L}_{norma} + \mathcal{L}_{\Phi} + \mathcal{L}_F + \mathcal{L}_{Yukawa}, \quad (2.3.1)$$

donde \mathcal{L}_{norma} describe los términos cinéticos y de interacción de los campos de norma, \mathcal{L}_{Φ} describe el sector escalar, \mathcal{L}_F describe el sector de fermiones y \mathcal{L}_{Yukawa}

describe las interacciones de Yukawa con el campo escalar. Este último término es muy importante, ya que en él se describen las masas de los fermiones a través del mecanismo de Higgs, y por lo tanto merece un análisis más profundo.

2.3.2.1. Sector de Yukawa

El Sector de Yukawa está descrito por el siguiente lagrangiano:

$$-\mathcal{L}_{Yukawa} = \sum_{m,n=1}^3 (Y_{mn}^u \bar{Q}'_{Lm} \bar{\Phi}_n u'_{Rn} + Y_{mn}^d \bar{Q}'_{Lm} \bar{\Phi}_n d'_{Rn} + Y_{mn}^l \bar{L}'_{Lm} \bar{\Phi}_n l'_{Rn}) + h.c. , \quad (2.3.2)$$

donde $Y^{u,d}$ son matrices de dimensión 3×3 las cuales describen los acoplamientos de Yukawa entre el doblete de Higgs y los sabores m y n de quarks y leptones. $\Phi = (\phi^\pm, \phi^0)^T$ es un doblete que transforma de la misma forma que $\bar{\Phi}$ bajo $SU(2)$ definido por $\bar{\Phi} = i\sigma_2 \Phi^*$. "h.c." denota el conjugado hermítico para incluir los términos de Yukawa complejos conjugados.

Se puede observar que este lagrangiano se puede dividir en dos partes:

$$\mathcal{L}_{Yukawa} = \mathcal{L}_Y^{Quarks} + \mathcal{L}_Y^{Leptones} , \quad (2.3.3)$$

donde \mathcal{L}_Y^{Quarks} y $\mathcal{L}_Y^{Leptones}$ representan los sectores de leptones y quarks respectivamente.

En este trabajo, se analiza únicamente el sector de quarks, dado su relevancia para la investigación actual.

2.3.2.2. Sector de quarks de Yukawa

De la Ecuación (2.3.2), se tiene que el sector de quarks del lagrangiano de Yukawa tiene la siguiente forma:

$$-\mathcal{L}_Y^q = \bar{Q}'_{Lm} (Y_{mn}^d \bar{\Phi}_n d'_{Rn} + Y_{mn}^u \bar{\Phi}_n u'_{Rn}) + h.c. , \quad (2.3.4)$$

donde $Q'_L = \begin{pmatrix} u' \\ d' \end{pmatrix}$ por lo que se puede reescribir el lagrangiano como:

$$-\mathcal{L}_Y^q = \sum_{m,n=1}^3 (Y_{mn}^u \bar{u}'_{Lm} \phi^{0*} u'_{Rn} + Y_{mn}^d \bar{u}'_{Lm} \phi^- u'_{Rn} + Y_{mn}^d \bar{u}'_{Lm} \phi^+ d'_{Rn} + Y_{mn}^d \bar{d}'_{Lm} \phi^0 d'_{Rn}) + h.c. \quad (2.3.5)$$

Se considera únicamente el término del campo escalar neutro de Higgs ya que es necesario para la generación de masas de las partículas:

$$\mathcal{L}_Y^{0q} = \sum_{m,n=1}^3 (Y_{mn}^u \bar{u}'_{Lm} \phi^{0*} u'_{Rn} + Y_{mn}^d \bar{d}'_{Lm} \phi^0 d'_{Rn}) + h.c. \quad (2.3.6)$$

Al expandir el campo escalar neutro alrededor de su estado de mínima energía:

$$\phi^0 = \frac{v}{\sqrt{2}} + H, \quad (2.3.7)$$

y sustituyendo en la Ecuación (2.3.6), se tiene:

$$\mathcal{L}_Y^{0q} = \sum_{m,n=1}^3 \left(\frac{v^*}{\sqrt{2}} Y_{mn}^u \bar{u}'_{Lm} u'_{Rn} + Y_{mn}^u \bar{u}'_{Lm} H^* u'_{Rn} + \frac{v}{\sqrt{2}} Y_{mn}^d \bar{d}'_{Lm} d'_{Rn} + Y_{mn}^d \bar{d}'_{Lm} H d'_{Rn} \right) + h.c. \quad (2.3.8)$$

Incorporando el término correspondiente del hermítico conjugado, el lagrangiano \mathcal{L}_Y^{0q} finalmente es:

$$\mathcal{L}_Y^{0q} = \sum_{m,n=1}^3 \left(\frac{v}{\sqrt{2}} Y_{mn}^u \bar{u}'_m u'_n + Y_{mn}^u \bar{u}'_m H^* u'_n + \frac{v}{\sqrt{2}} Y_{mn}^d \bar{d}'_m d'_n + Y_{mn}^d \bar{d}'_m H d'_n \right) + h.c. \quad (2.3.9)$$

Se pueden reconocer dos partes en el lagrangiano:

$$\mathcal{L}_Y^{0q} = \mathcal{L}_m^{0q} + \mathcal{L}_I^{0q}, \quad (2.3.10)$$

donde \mathcal{L}_m^{0q} y \mathcal{L}_I^{0q} corresponden al lagrangiano de masa y al lagrangiano de interacción respectivamente.

El Lagrangiano de masa \mathcal{L}_m^{0q} se utiliza para definir la masa de los quarks y viene dado como:

$$\mathcal{L}_m^{0q} = \sum_{m,n=1}^3 \left(\bar{u}'_{Lm} M_{mn}^u u'_{Rn} + \bar{d}'_{Lm} M_{mn}^d d'_{Rn} \right), \quad (2.3.11)$$

donde se definen las matrices:

$$M_{mn}^{u,d} = \frac{v}{\sqrt{2}} Y_{mn}^{u,d}. \quad (2.3.12)$$

En el ME la matriz de masa M y la matriz de Yukawa Y están alineadas, es decir, comparten los mismos eigenvectores. Hasta este punto se ha utilizado una base de sabor, pero es más conveniente utilizar una base de masa. Para lograrlo, es necesario diagonalizar la matriz M_{mn} . Esto se logra encontrando los eigenvalores y eigenvectores de M_{mn} . Sin pérdida de generalidad, se puede asumir que $Y_{mn}^{u,d}$ es hermítica, lo cual implica que M_{mn} también lo es, según la Ecuación (2.3.12).

Del álgebra lineal se establece que para cualquier matriz hermítica M , siempre es posible encontrar una matriz unitaria V , tal que VMV^\dagger es diagonal y real. Esto significa que se puede obtener una matriz diagonal de M de la forma:

$$M_{diag} = VMV^\dagger, \quad (2.3.13)$$

donde V es una matriz unitaria, es decir, $V^\dagger V = I$. En otras palabras, las columnas de V forman un conjunto ortonormal de vectores propios de M , y los elementos diagonales de M_{diag} son los valores propios correspondientes.

Reescribiendo el lagrangiano de masas:

$$\mathcal{L}_m^{0q} = \sum_{m,n=1}^3 \left[(\bar{u}'_{Lm} V_u^\dagger) (V_u M_{mn}^u V_u^\dagger) (V_u u'_{Rn}) + (\bar{d}'_{Lm} V_d^\dagger) (V_d M_{mn}^d V_d^\dagger) (V_d d'_{Rn}) \right]. \quad (2.3.14)$$

Para encontrar los eigenvalores de $M^{u,d}$, que representan las masas de los quarks, y sus eigenvectores asociados que describen los campos físicos, se utilizan las siguientes transformaciones unitarias que diagonalizan las matrices de masa.

$$u = V_u u', \quad d = V_d d'. \quad (2.3.15)$$

Por lo que, los lagrangianos de masa para quarks tipo u y d , se escriben respectivamente como:

$$\mathcal{L}_m^u = \bar{u}' M^u u', \quad (2.3.16)$$

$$\mathcal{L}_m^d = \bar{d}' M^d d'. \quad (2.3.17)$$

Es así que al tomar la matriz como unitaria, se asegura que sea real y diagonal, es decir:

$$V_{u,d} M^{u,d} V_{u,d}^\dagger = M_{diag}^{u,d} = \begin{pmatrix} m_{u,d} & 0 & 0 \\ 0 & m_{c,s} & 0 \\ 0 & 0 & m_{t,b} \end{pmatrix}. \quad (2.3.18)$$

Se definen los nuevos campos físicos (\bar{u}, \bar{d}) y (u, d) como:

$$\bar{u} = (\bar{u}, \bar{c}, \bar{t}), \quad u = \begin{pmatrix} u \\ c \\ t \end{pmatrix}, \quad (2.3.19)$$

$$\bar{d} = (\bar{d}, \bar{s}, \bar{b}), \quad d = \begin{pmatrix} d \\ s \\ b \end{pmatrix}. \quad (2.3.20)$$

Por lo que finalmente, los lagrangianos de masa tienen la siguiente estructura:

$$\mathcal{L}_m^u = \bar{u} m_u u + \bar{c} m_c c + \bar{t} m_t t, \quad (2.3.21)$$

$$\mathcal{L}_m^d = \bar{d} m_d d + \bar{s} m_s s + \bar{b} m_b b, \quad (2.3.22)$$

donde las antipartículas de cada quark se denotan por $\bar{u}, \bar{c}, \bar{t}, \bar{d}, \bar{s}, \bar{b}$ respectivamente.

De lo anterior, se desprende al lagrangiano de masas dado por:

$$\mathcal{L}_{masa} = -\bar{d}_L \bar{M}^d d_R - \bar{u}_L \bar{M}^u u_R + h.c. \quad (2.3.23)$$

La matriz de mezcla V_{CKM} se construye a partir de las matrices de rotación V_q a través de la siguiente expresión:

$$V_{CKM} = V_u^\dagger V_d \quad (2.3.24)$$

Dado que las masas de los quarks y los elementos de la matriz V_{CKM} son los únicos parámetros experimentales medibles, las matrices V_u y V_d no son observables físicos ni predichas por la teoría, lo que implica que tampoco lo es la matriz V_{CKM} .

Los acoplamientos del bosón de Higgs con los distintos sabores de quarks, vienen descritos por el lagrangiano de interacción, el cual se define como:

$$\mathcal{L}_Y^{0q} = \sum_{m,n=1}^3 (Y_{mn}^u \bar{u}'_m H^* u'_n + Y_{mn}^d \bar{d}'_m H^* d'_n) + h.c. \quad (2.3.25)$$

Escribiendo los lagrangianos en términos de los campos físicos de los quarks dados en las Ecuaciones 2.3.19 y 2.3.20, se tiene:

$$\mathcal{L}_1^u = \frac{\sqrt{2}}{v} \bar{u} M_u u H \quad (2.3.26)$$

$$\mathcal{L}_1^d = \frac{\sqrt{2}}{v} \bar{d} M_d d H \quad (2.3.27)$$

Los cuales, finalmente se convierten en:

$$\mathcal{L}_1^u = \frac{\sqrt{2}}{v} \bar{u} m_u u H + \frac{\sqrt{2}}{v} \bar{c} m_c c H + \frac{\sqrt{2}}{v} \bar{t} m_t t H \quad (2.3.28)$$

$$\mathcal{L}_1^d = \frac{\sqrt{2}}{v} \bar{d} m_d d H + \frac{\sqrt{2}}{v} \bar{s} m_s s H + \frac{\sqrt{2}}{v} \bar{b} m_b b H \quad (2.3.29)$$

De este lagrangiano se muestra que el acoplamiento es proporcional a las masas de los diferentes tipos de quarks. Esto significa que el Higgs se acopla más fuertemente a quarks más pesados. Además, el lagrangiano conserva el sabor, es decir, no hay acoplamientos entre quarks de diferentes generaciones inducidos por el escalar de Higgs.

2.3.3. La matriz de mezcla V_{ckm}

En 1963, Nicola Cabibbo propuso que los quarks débiles no eran los mismos que los quarks de sabor, sino que estaban relacionados por una matriz de rotación. Esto explicaba la supresión de ciertos procesos débiles. Posteriormente, en 1973, Makoto Kobayashi y Toshihide Maskawa extendieron el trabajo de Cabibbo a tres generaciones de quarks, proponiendo la matriz 3×3 ahora conocida como matriz V_{CKM} .

La matriz V_{CKM} (ver Ecuación (2.3.24)) describe cómo los eigenvectores de masa de los quarks no coinciden con los eigenvectores de interacción débil. Los elementos de matriz (ver Ecuación (2.3.30)) representan las probabilidades de transición entre quarks de diferentes sabores mediante la interacción débil. La matriz V_{CKM} es unitaria y se parametriza con tres ángulos de mezcla y una fase compleja.

$$V_{CKM} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \quad (2.3.30)$$

Desde su introducción, la matriz V_{CKM} ha sido probada extensamente por experimentos de física de partículas. Sus predicciones han sido confirmadas con gran precisión, convirtiéndola en una pieza fundamental del Modelo Estándar. El trabajo de Cabibbo, Kobayashi y Maskawa fue reconocido con el Premio Nobel de Física en 2008, por haber descubierto el origen del rompimiento de la simetría y predicho la existencia de 6 sabores de quarks.

2.3.3.1. Parametrización Chau-Keng

Como se mencionó anteriormente, la matriz V_{ckm} es unitaria y puede parametrizarse

zarse en términos de tres ángulos y una fase. La parametrización más utilizada, y la que se empleará en este trabajo, es la parametrización de Chau-Keng, definida por los ángulos θ_{12} , θ_{13} , θ_{23} y la fase δ_{13} , a través de la siguiente multiplicación de matrices [85, 86]:

$$\begin{aligned}
 V_{CKM} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & C_{23} & S_{23} \\ 0 & -S_{23} & C_{23} \end{pmatrix} \begin{pmatrix} C_{13} & 0 & S_{13}e^{-i\delta_{13}} \\ 0 & 1 & 0 \\ -S_{13}e^{i\delta_{13}} & 0 & C_{13} \end{pmatrix} \begin{pmatrix} C_{12} & S_{12} & 0 \\ -S_{12} & C_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
 &= \begin{pmatrix} C_{12}C_{13} & S_{12}C_{13} & S_{13}e^{-i\delta_{13}} \\ -S_{12}C_{23} - C_{12}S_{23}S_{13}e^{i\delta_{13}} & C_{12}C_{23} - S_{12}S_{23}S_{13}e^{i\delta_{13}} & S_{23}C_{13} \\ S_{12}S_{23} - C_{12}C_{23}S_{13}e^{i\delta_{13}} & C_{12}S_{23} - S_{12}C_{23}S_{13}e^{i\delta_{13}} & C_{23}C_{13} \end{pmatrix}, \tag{2.3.31}
 \end{aligned}$$

donde se ha utilizado la notación $C_{ij} = \text{Cos}(\theta_{ij})$ y $S_{ij} = \text{Sen}(\theta_{ij})$, además, los índices (i, j) , relacionan las familias de los quarks.

Los elementos θ_{13} , θ_{12} , θ_{23} se obtienen al elegir las tres magnitudes de los elementos $|(V_{CKM})_{us}|$, $|(V_{CKM})_{ub}|$ y $|(V_{CKM})_{cb}|$ y la fase δ_{13} se obtiene del invariante de Jarlskog (J) a través de las siguientes relaciones[86]:

$$\sin \theta_{13} = |(V_{CKM})_{ub}|, \tag{2.3.32}$$

$$\sin \theta_{12} = \frac{|(V_{CKM})_{us}|}{\sqrt{1 - |(V_{CKM})_{ub}|^2}}, \tag{2.3.33}$$

$$\sin \theta_{12} = \frac{|(V_{CKM})_{us}|}{\sqrt{1 - |(V_{CKM})_{ub}|^2}}, \tag{2.3.34}$$

$$\sin \theta_{23} = \frac{|(V_{CKM})_{cb}|}{\sqrt{1 - |(V_{CKM})_{ub}|^2}}, \tag{2.3.35}$$

$$\sin \delta_{13} = \frac{J}{\sqrt{1 - (\sin \theta_{12})^2} \sin \theta_{12} [1 - (\sin \theta_{13})^2] \sin \theta_{13} \sqrt{1 - (\sin \theta_{23})^2} \sin \theta_{23}}, \tag{2.3.36}$$

donde $J = \text{Im}(V_{us} V_{cb} V_{ub}^*, V_{cs}^*)$.

2.3.3.2. Valores experimentales

La matriz V_{CKM} describe con precisión la interacción entre quarks. De acuerdo a los datos experimentales más actuales [87], se han determinado los valores numéricos de los nueve elementos que componen esta matriz. A continuación, se presentan los valores centrales y los errores asociados a cada uno de estos elementos:

$$|V_{CKM}| = \begin{pmatrix} 0.97401 \pm 0.00011 & 0.22650 \pm 0.00048 & 0.00361^{+0.00011}_{-0.00009} \\ 0.22636 \pm 0.00048 & 0.97320 \pm 0.00011 & 0.04053^{+0.00083}_{-0.00061} \\ 0.00854^{+0.00023}_{-0.00016} & 0.03978^{+0.00082}_{-0.00060} & 0.999172^{+0.000024}_{-0.000035} \end{pmatrix}. \quad (2.3.37)$$

Mientras que para las masas de los quarks en escala de M_z se tiene [88]:

$$\begin{aligned} m_u &= 1.23 \pm 0.21 \text{ MeV}, & m_d &= 2.67 \pm 0.19 \text{ MeV}, & m_s &= 53.16 \pm 4.61 \text{ MeV} \\ m_c &= 0.620 \pm 0.017 \text{ GeV}, & m_b &= 2.839 \pm 0.026 \text{ GeV}, & m_t &= 168.26 \pm 0.75 \text{ GeV} \end{aligned} \quad (2.3.38)$$

En la Ecuación (2.3.38) se observa una jerarquía de masas donde se establece que las masas de los quarks siguen la relación $|m_d| < |m_s| < |m_b|$ y $|m_u| < |m_c| < |m_t|$. Esta secuencia de masas crecientes dentro de cada familia se conoce como la jerarquía de masas en el modelo estándar de la física de partículas.

2.4. Formalismo de texturas

El Modelo Estándar Electro débil (MEED) no proporciona información sobre la estructura de las matrices de masa de quarks, lo que tiene un impacto significativo en el patrón de mezcla de sabor de quarks. Se espera que una teoría más fundamental permita determinar la estructura concreta de las matrices de masa de quarks M_u y M_d , lo que a su vez permitiría calcular de manera precisa las seis masas de quarks, los tres ángulos de mezcla de sabor y la fase de violación CP .

El formalismo de texturas es una propuesta teórica dentro de la física de partículas

que busca explicar el origen de las masas de los quarks y la estructura de la matriz de mezcla de Cabibbo-Kobayashi-Maskawa V_{CKM} . Este formalismo fue introducido por Harald Fritzsch en 1978 y se basa en la idea de que las matrices de masa de los quarks tienen una estructura específica llamada “texturas” que podría ayudar a determinar la estructura concreta de las matrices de masa de quarks y, por lo tanto, a calcular de manera precisa las propiedades de los quarks y la matriz de mezcla de Cabibbo-Kobayashi-Maskawa.

2.4.1. Modelos de matrices de masa tipo texturas

Desde un punto de vista fenomenológico, es posible analizar si hay una simetría subyacente considerando una matriz de Yukawa bajo la fenomenología existente y restringir sus entradas con información experimental. Esto inicialmente se propuso por Harald Fritzsch con una textura de masas con el menor número de parámetros libres que puedan atender a límites experimentales, colocando ceros en la textura ya sea en la diagonal ó fuera de la diagonal de manera simétrica, considerando en los dos casos como un cero.

Las matrices de masa de quarks M_u y M_d pueden considerarse hermíticas en general, lo que permite asumir texturas hermíticas de la forma:

$$M_q = \begin{pmatrix} E_q & D_q & F_q \\ D_q^* & C_q & B_q \\ F_q^* & B_q^* & A_q \end{pmatrix} \quad q = u, d \quad (2.4.1)$$

La física queda invariante bajo una transformación unitaria $M_q \rightarrow SM_qS^\dagger$, donde S es una matriz unitaria arbitraria [89]. Esta libertad permite una configuración adicional para las matrices de masa de quarks como:

$$M_q = \begin{pmatrix} 0 & D_q & 0 \\ D_q^* & 0 & B_q \\ 0 & B_q^* & A_q \end{pmatrix} \quad q = u, d, \quad (2.4.2)$$

donde el elemento A_q es real, mientras que B_q y D_q son complejos y se expresan en forma polar como $Z_q = |Z_q|e^{i\phi_{Z_q}}$, donde $|Z_q|$ es la magnitud y $i\phi_{Z_q}$ es su fase

angular ($Z = B, D$).

La estructura de matriz de masa mostrada en la Ecuación (2.4.2) representa la propuesta original de Fritzsche, es decir una matriz de masa simétrica con seis ceros de textura [15]. Cabe destacar que en este contexto, una matriz de masa con “ n ” ceros de textura en particular, se denota de tal manera que la suma del número de ceros en posiciones diagonales y un par de ceros simétricamente colocados fuera de la diagonal, contados como uno, es “ n ”.

2.4.2. Viabilidad de los modelos de texturas

El proceso de evaluar la viabilidad de una combinación específica de texturas con ceros involucra de forma general, los siguientes pasos:

1. Derivar la matriz V_{CKM} correspondiente: Se calcula la matriz de Cabibbo-Kobayashi-Maskawa (V_{CKM}) en función de la combinación de textura de ceros que se está analizando.
2. Comparar la matriz V_{CKM} derivada con datos experimentales: Se compara la compatibilidad de la matriz V_{CKM} derivada con los valores experimentales más actuales de la matriz V_{CKM} .
3. Determinar la viabilidad de la combinación de texturas con ceros: Basándose en la comparación entre las matrices V_{CKM} derivada y obtenida experimentalmente, se llega a una conclusión sobre la viabilidad de la combinación de la textura con ceros bajo investigación.

Siguiendo esta metodología, se puede evaluar sistemáticamente la factibilidad de varias combinaciones o modelos de texturas con ceros y su compatibilidad con las observaciones experimentales más recientes.

Análisis realizados sobre diferentes modelos de texturas con seis ceros (como el presentado en la Ecuación (2.4.2)) indican que este tipo de modelos han sido excluidos de acuerdo a los datos experimentales más recientes [90]. Esto también ocurre para el caso de modelos de texturas con cinco ceros, los cuales también han sido excluidos por los datos experimentales más recientes [91].

Considerando el contexto previo y el éxito demostrado por el modelo de texturas de cuatro ceros de Fritzsich (ver Ecuación (2.4.3)) en estudios anteriores [89, 92], esta investigación se centra en evaluar su viabilidad y las posibles implicaciones fenomenológicas a la luz de los datos experimentales más recientes. Este análisis, que se presentará en los siguientes capítulos, permitirá evaluar la solidez y relevancia del modelo de texturas de Fritzsich con cuatro ceros, y proporcionará una base sólida para futuras investigaciones en esta área.

$$M_u = \begin{pmatrix} 0 & D_u & 0 \\ D_u^* & C_u & B_u \\ 0 & B_u^* & A_u \end{pmatrix}, \quad M_d = \begin{pmatrix} 0 & D_d & 0 \\ D_d^* & C_d & B_d \\ 0 & B_d^* & A_d \end{pmatrix} \quad (2.4.3)$$

Capítulo 3

Estado del Arte

El enfoque algorítmico bio-inspirado propuesto para abordar el problema de optimización del criterio χ^2 dentro del análisis numérico del modelo de texturas de cuatro ceros, involucra una serie de conceptos de diversas áreas de estudio. Por tal motivo, el estado del arte revisa en primer lugar, trabajos previos relacionados al análisis numérico de la estructura tipo texturas de cuatro ceros. Después, se discute sobre trabajos previos dentro de la FAE en los cuales se han implementado ABI, ya sea en el contexto del análisis de modelos de textura o en otras aplicaciones. Finalmente, se analizan propuestas para mejorar el rendimiento del algoritmo DE en problemas de optimización con parámetros reales con un solo objetivo y con restricciones de frontera, tal como el problema de optimización del criterio χ^2 .

3.1. Análisis numérico de la estructura tipo texturas de cuatro ceros

Determinar la viabilidad de los modelos de texturas de acuerdo a los datos experimentales más recientes de observables físicas como la magnitud de los elementos $|V_{us}|$, $|V_{ub}|$ y $|V_{cb}|$ de la matriz V_{CKM} , así como los valores de las masas de los quarks, es un tema activo dentro del formalismo de texturas. Analizar de forma numérica el espacio de parámetros de los elementos de las matrices de masa de los quarks M_u y M_d , y la capacidad de predicción de los elementos de la matriz V_{CKM} , permite encontrar nuevas características del modelo, o poder excluirlo. De forma particular, a continuación se revisan los trabajos más recientes y relevantes sobre el análisis numérico del modelo de texturas de cuatro ceros.

Establecer la forma concreta una matriz de masa de los quarks, la cual sea consistente de acuerdo a los datos experimentales, permite determinar los valores de

cantidades como las masas de los quarks, los ángulos de mezcla y la fase de violación CP. En [93] se propone de forma analítica una estructura de matriz de masas para el sector de quarks M_q considerando cuatro ceros en la matriz. A partir de los elementos D_q, B_q, C_q y A_q de esta matriz de masa, los cuales se definen por medio del parámetro $a_q = 1 - \delta_q$, se obtienen las expresiones analíticas de las matrices de transformación O_q y de los elementos de la matriz V_{ckm} . Como parte del análisis numérico, y con el fin de revisar la compatibilidad entre las expresiones analíticas de los elementos de la matriz V_{ckm} y los datos experimentales [94], se utiliza un criterio de χ^2 . A partir de este análisis, se encontraron los rangos para los parámetros de la diagonal de la matriz de masa de quarks. Los resultados obtenidos reportan ser congruentes respecto a trabajos previos, y se motiva a utilizar el parámetro a_q en futuros análisis numéricos.

En [95] se analiza la viabilidad de todos los modelos de textura de matrices de masa de quarks. Entre estos modelos de textura, se estudian modelos generales y simétricos. Para probar que cierto modelo de textura fuera compatible con los valores experimentales de las masas de los quarks, y de los parámetros de mezcla $\lambda, A, \bar{\rho}$ y $\bar{\eta}$ [96, 97], se realizó un análisis numérico de χ^2 , en el cual, para cada modelo se estableció de manera similar una función con la siguiente forma:

$$\chi^2(x) = \sum_j \chi_j^2(x) \quad \text{con} \quad \chi_j^2(x) = \left(\frac{P_j(x) - \mathcal{O}_j}{\sigma_j} \right)^2, \quad (3.1.1)$$

donde x representa el conjunto de parámetros del modelo, $P_j(x)$ es la predicción del modelo para la observable \mathcal{O}_j con valores centrales $\bar{\mathcal{O}}_j$ y errores σ_j .

En este análisis numérico, se identifica que un modelo de texturas es compatible con los datos experimentales si en el valor mínimo de χ^2 la contribución de cada observable a χ^2 es a lo máximo 25, lo cual implica que $\chi^2 \leq 250$. Los autores propusieron además, un método numérico para conocer el alcance predictivo de un modelo compatible con los datos experimentales. Este método se basa en la idea de conocer que tanto la predicción de la observable \mathcal{O}_i puede desviarse de su media variando x tal que todas $P(x)_j$ con $j \neq i$ se mantengan cerca de $\bar{\mathcal{O}}_i$. Los resultados del análisis concluyeron que, diversos modelos de texturas pueden ser compatibles con los datos experimentales, sin embargo algunos de ellos, no

muestran un gran alcance predictivo.

En [89] se explora de forma numérica el espacio de parámetros de las matrices de masa de quarks M_u y M_d con cuatro ceros. El análisis realizado para explorar dicho espacio de parámetros, utilizó los valores experimentales de la matriz V_{ckm} y las masas de los quarks en escala de M_z [97, 98]. Los elementos $|V_{us}|$, $|V_{ub}|$, $|V_{cb}|$ y la observable $\text{Sin}(2\beta)$ fueron seleccionadas como entradas. Cabe resaltar que los autores no reportan el criterio y/o algoritmos utilizados para comprobar la compatibilidad con los datos experimentales seleccionados. Los resultados obtenidos de este análisis, permitieron encontrar una región no explorada anteriormente del espacio de parámetros. La integración de esta nueva región, dio paso a la propuesta de nuevas estructuras para las matrices de masa de quarks M_u y M_d .

Para ampliar el entendimiento de la relación entre la textura de cuatro ceros y el patrón jerárquico observado de forma experimental en la matriz V_{ckm} , en [88] se propusieron nuevas aproximaciones analíticas para los elementos de las matrices de transformación O_q . Con estas aproximaciones, se obtuvieron las correspondientes expresiones analíticas de cada uno de los elementos de la matriz V_{ckm} en términos de un nuevo parámetro r_q . La introducción de este parámetro permitió obtener expresiones analíticas más simples para los elementos de la matriz V_{ckm} en términos de las masas de los quarks, dos fases y el parámetro r_q . Con ayuda de estas expresiones y de los valores obtenidos de forma experimental para la matriz V_{ckm} , así como los valores centrales de las masas de los quarks [99], se llevó a cabo un análisis numérico a partir del cual se obtienen: los valores de cada uno de los nueve elementos de la matriz V_{ckm} , el valor del invariante de Jarlskog y las correspondientes texturas de masas de quarks. A partir estos resultados analíticos, se encontró que bajo las expresiones analíticas aproximadas para los elementos la matriz V_{ckm} , existe hasta un 6 % de error, en comparación de las expresiones genéricas de la matriz V_{ckm} , al ser evaluadas con los mismos valores para los parámetros A_u , A_d , ϕ_1 y ϕ_2 . Con estos resultados se pudo concluir que los elementos $(2, 2)$, $(2, 3)$ y $(3, 3)$ de la matriz de masa de quarks deben tener una jerarquía relativamente más débil que su correspondiente jerarquía de masas de los quarks.

En la Tabla 3.1.1 se muestran las características principales de acuerdo al objetivo

de la presente investigación, de los trabajos relacionados al análisis de la textura de cuatro ceros [88, 89, 93].

Tabla 3.1.1: Trabajos relacionados al análisis de la textura de cuatro ceros del estado del arte y algunas de sus características.

Autor	Enfoque de estudio	Criterio de Ajuste	Método de optimización
O. Félix Beltrán et al., 2013 [93]	Obtener correlaciones entre una textura de cuatro ceros y los valores experimentales provenientes de la matriz V_{CKM} mediante un ajuste de χ^2 .	χ^2	No especificado (Simplex)
P.O. Ludl y W. Grimus, 2015 [95]	Identificar la compatibilidad de distintos modelos de textura con datos experimentales, y conocer su alcance predictivo.	χ^2	Nelder-Mead
Xing y Zhao, 2015 [89]	Encontrar nuevas regiones de valores permitidos para los parámetros de una textura de cuatro ceros.	No mencionado	No mencionado
H. Fritzscht et al., 2021 [88]	Demostrar que los elementos (2,2), (2,3) y (3,3) de M_q tienen una jerarquía débil de textura, inclusive más débil que la correspondiente jerarquía de masa en quarks.	$\frac{ V _{aprox} - V _{exacto}}{ V _{exacto}}$	No mencionado
Propuesta, 2024	Identificar la compatibilidad del modelo de texturas de cuatro ceros con los datos experimentales actuales y conocer su alcance predictivo.	χ^2	HE-DEPSO

3.2. Algoritmos bio-inspirados dentro de la física de altas energías

En la Física de Altas Energías (FAE) los Algoritmos Bio-Inspirados (ABI) han sido poco explorados, a pesar de esto, han mostrado ser capaces de brindar buenos resultados en contextos experimentales y fenomenológicos. Una de sus aplicaciones ha sido la optimización de hiperparámetros de modelos de aprendizaje de máquina, los cuales, son utilizados en el análisis de grandes cantidades de datos provenientes de experimentos de FAE. Otras aplicaciones recaen en la optimización del diseño de aceleradores y en la discriminación de modelos provenientes de teorías supersimétricas. A continuación se revisan los trabajos más recientes y relevantes, concernientes a la aplicación de los ABI dentro de la FAE.

En [16], por ejemplo, se propone el uso de algoritmos evolutivos con el fin de encontrar la mejor configuración de hiperparámetros para un modelo de árboles de decisión, el cual tiene el objetivo de realizar el análisis de grandes cantidades de datos derivados de experimentos realizados en FAE. De forma concreta, los algoritmos que se proponen utilizar son GA y PSO. El rendimiento de estos algoritmos evolutivos es comparado contra otras técnicas de optimización como “*Gradient Descent*” (GD), “*Grid Search*” (GS), y “*Random Guessing*” (RNG). Las tareas que ayudaron a cuantificar el rendimiento de estos algoritmos, consistieron en la optimización de la función de Rosenbrock y la implementación de estos algoritmos como optimizadores de hiperparámetros del método de árboles de decisión. En ambas tareas se observó un rendimiento sobresaliente por parte de los AE, y en especial del algoritmo PSO.

Una aplicación de estos algoritmos dentro de la fenomenología de la FAE se muestra en [18]. En este trabajo se hace uso de un GA con el fin de abordar el problema de discriminar entre modelos de ruptura de supersimetría. La discriminación entre modelos se realiza mediante la evaluación de la precisión requerida en ciertas mediciones de masa para poder distinguir entre dos modelos. Se construye una función de ajuste denominada como: distancia relativa. Esta función, determina la diferencia entre dos espectros de masa. Es por medio de esta función que se obtiene el valor correspondiente a la precisión requerida para las mediciones de masa. Este valor a la vez permite establecer la distinción entre modelos teóricos.

La optimización del diseño de aceleradores de partículas es otra de las aplicaciones que han tenido los algoritmos de optimización metaheurísticos. En [17] se muestra una variedad de casos en los que, de forma particular, se hace uso de GA para solucionar distintos problemas en física de aceleradores, principalmente relacionados al diseño de estos aceleradores. Por su parte en [100, 101], con el objetivo de optimizar el diseño de la apertura dinámica del anillo en los colisionadores SuperKEKB y CEPC, se propone el uso del algoritmo DE.

En la Tabla 3.2.1 se presentan de forma resumida los trabajos más relevantes y recientes, relacionados a la aplicación de algoritmos bio-inspirados dentro de la física de altas energías.

Tabla 3.2.1: Trabajos relacionados a las técnicas de optimización bio-inspiradas y sus aplicaciones dentro del área de física de altas energías.

Autor	Algoritmo	Aplicación
B.C Allanach et al., 2004 [18]	GA	Discriminación de forma experimental de modelos de súper simetría de ruptura.
Hofler et al., 2013 [17]	GA	Optimización del diseño de aceleradores de partículas.
Zhang et al., 2016 [100]	DE	Optimización del diseño de aceleradores de partículas .
Wu et al., 2020 [101]	DE	Optimización del diseño de aceleradores de partículas .
Laurits Tani et al., 2021 [16]	GA PSO	Optimización de hiperparámetros en modelos de aprendizaje automático.
Propuesta, 2024	HE-DEPSO	Optimización de χ^2 dentro del formalismo de texturas.

3.3. Mejoras en el algoritmo de Evolución Diferencial

El algoritmo DE es uno de los algoritmos evolutivos que más ha destacado en tiempos recientes debido a su sencillez, poder y eficiencia [102, 103], sin embargo, al igual que otros AE, es propenso a sufrir convergencia prematura y estancamiento en mínimos locales [102, 104]. Las mejoras implementadas para solucionar estos problemas se pueden clasificar de la siguiente forma:

- 1) Cambiar la estrategia de mutación. La fase de mutación del algoritmo de DE tiene una importancia, ya que permite la integración de nuevos individuos en la población existente, influyendo así en su rendimiento. Algoritmos como “*Self-Adaptive Differential Evolution*” (JADE) [105] y “*Composite DE*” (CoDE) [106] han presentado nuevas estrategias de mutación e implementado diferentes estrategias de mutación, respectivamente, con el fin de mejorar la eficacia del algoritmo DE.
- 2) Control de parámetros. El algoritmo DE es sensible a la configuración de los de sus principales parámetros: el factor de escala F y la probabilidad de cruce Cr [102, 107]. El control auto-adaptativo de estos parámetros ha demostrado mejorar

significativamente el rendimiento del algoritmo DE. En este sentido, los algoritmos SHADE [108] y “*Self-Adaptive DE*” (SaDE) [109] representan dos variantes bastante conocidas.

3) Hibridación con otros algoritmos de optimización. Una forma de mejorar el rendimiento del algoritmo DE, consiste en tomar ventaja de las fortalezas de operadores de otros algoritmos e incorporarlos dentro de la estructura del algoritmo DE mediante un proceso de hibridación [103, 110]. La hibridación con otros algoritmos de IC como: “*Artificial Neural Networks*” (ANN), “*Ant Colony Optimization*” (ACO) y PSO, han marcado una tendencia dentro de la última década [103].

Como se mencionó anteriormente, una de las variantes más representativas concernientes a la mejora del control de los parámetros F y Cr del algoritmo DE, fue propuesta en el algoritmo SHADE [108]. De forma precisa, en cada iteración del algoritmo SHADE los parámetros F y Cr asociados a vectores de prueba con un mejor ajuste que su vector padre, son almacenados en dos conjuntos: S_F y S_{Cr} respectivamente. Dos archivos o memorias M_F y M_{Cr} con un tamaño H definido, y cuyos contenidos $M_{Cr,k}$ y $M_{F,k}$ ($k = 1, \dots, H$) son inicializados con un valor de 0.5, sirven para generar los parámetros F_i y Cr_i del i -ésimo individuo $X_{i,t}$ en cada iteración t . La forma de generar los valores de estos parámetros requiere la selección uniforme de un índice $r_i \in [1, H]$ y es llevada a cabo de acuerdo a las siguientes expresiones:

$$Cr_i = \text{randn}_i(M_{Cr,r_i}, 0.1), \quad (3.3.1)$$

$$F_i = \text{randc}_i(M_{F,r_i}, 0.1), \quad (3.3.2)$$

donde M_{Cr,r_i} es el elemento seleccionado de M_{Cr} para generar el valor de Cr_i , mientras que M_{F,r_i} es el elemento seleccionado de M_F para generar el valor de F_i . $\text{randn}_i(M_{Cr,r_i}, 0.1)$ representa una distribución normal con media M_{Cr,r_i} y desviación estándar 0.1, mientras que $\text{randc}_i(M_{F,r_i}, 0.1)$ representa una distribución de Cauchy con posición de parámetro M_{F,r_i} y factor de escala 0.1. Al generar el valor del parámetro Cr_i , se debe verificar que este se encuentre dentro del rango $[0, 1]$, en caso contrario, si $Cr_i < 0$ entonces Cr_i es truncado a 0 y si $Cr_i > 1$ entonces

Cr_i es truncado a 1. Para el parámetro F_i , si $F_i > 1$ entonces F_i es truncado a 1, en otro caso, si $F_i < 0$ entonces F_i es regenerado hasta que $F_i > 0$.

Al final de cada iteración, los contenidos de las memorias M_{Cr} y M_F son actualizados de la siguiente forma:

$$M_{Cr,k,t+1} = \begin{cases} mean_{WL}(S_{Cr}), & \text{si } S_{Cr} \neq \emptyset \\ M_{Cr,k,t}, & \text{otro caso} \end{cases}, \quad (3.3.3)$$

$$M_{F,k,t+1} = \begin{cases} mean_{WL}(S_F), & \text{si } S_F \neq \emptyset \\ M_{F,k,t}, & \text{otro caso} \end{cases}, \quad (3.3.4)$$

donde $mean_{WL}(S)$ es la media ponderada de Lehmer definida por medio de la Ecuación (3.3.5), en la cual S hace referencia a S_{Cr} o S_F .

$$mean_{WL} = \frac{\sum_{k=1}^{|S|} w_k \cdot S_k^2}{\sum_{k=1}^{|S|} w_k \cdot S_k} \quad (3.3.5)$$

$$w_k = \frac{\Delta f_k}{\sum_{l=1}^{|S|} \Delta f_l} \quad (3.3.6)$$

$$\Delta f_k = |f(X_{p,k,t}) - f(X_{k,t})| \quad (3.3.7)$$

En la Ecuación (3.3.3) y Ecuación (3.3.4), $k \in [1, H]$ determina la posición en la memoria a actualizar. En la iteración t , el elemento k -ésimo en la memoria es actualizado. Al principio del proceso de optimización $k = 1$, y es incrementado cada vez que un nuevo elemento es agregado a la memoria. Si $k > H$, k es asignado igual a 1.

Siguiendo la tendencia de mejorar rendimiento del algoritmo DE mediante hibridaciones, el Algoritmo de Evolución Diferencial basado en Optimización por Enjambre de Partículas con mutación auto-adaptativa (DEPSO) [19] es un método

actual, el cual incorpora características del algoritmo PSO dentro de la estructura del algoritmo DE para la solución de problemas numéricos de optimización.

En este algoritmo, el top $\alpha \cdot NP$ ($\alpha \in [0.1, 0.9]$) de las mejores partículas en la iteración actual, es utilizado para conformar un sub-enjambre élite (P), mientras que el resto, es utilizado para formar un sub-enjambre no élite sub-enjambre no élite (Q).

El algoritmo DEPSO sigue un esquema similar a la versión estándar del algoritmo DE, es decir, utiliza los operadores de mutación, cruce y selección dentro del proceso evolutivo del algoritmo. Dentro de la mutación, se implementa una estrategia de mutación la cual, de forma auto-adaptativa, selecciona entre dos esquemas de mutación para generar en cada iteración un vector mutado $V_{i,t+1}$ para la i -ésima partícula $X_{i,t}$ de la siguiente forma:

$$V_{i,t+1} = \begin{cases} X_{r1,t}^P + F_i \cdot (X_{r2,t}^P - X_{r3,t}^Q) & \text{si } rand[0, 1] < SP_t \\ \omega \cdot X_{i,t} + c_1 \cdot r_1 \cdot (X_{gbest,t} - X_{i,t}) \\ \quad + c_2 \cdot r_2 \cdot (X_{pbest,t} - X_{i,t}) & \text{otro caso} \end{cases}, \quad (3.3.8)$$

donde $rand(0, 1)$ es un número aleatorio seleccionado de forma uniforme dentro del intervalo $[0, 1]$, y SP_G representa la probabilidad de seleccionar entre alguno de los dos esquemas de mutación. En la Ecuación (3.3.8), el caso en el cual $rand[0, 1] < SP_t$ representa la forma en la cual un novedoso esquema de mutación denominado DE/e-rand/1 genera un vector mutado $V_{i,t+1}$. En este esquema son requeridas dos soluciones óptimas del sub-enjambre élite ($X_{r1,G}^P$ y $X_{r2,G}^P$) y una solución del sub-enjambre no élite ($X_{r3,G}^Q$), además se utiliza un factor de escala F_i , el cual actúa a nivel de cada partícula. El caso restante representa la forma en la cual el esquema de mutación del algoritmo PSO estándar es utilizado para genera un vector mutado $V_{i,t+1}$.

La probabilidad de selección SP_G de la Ecuación (3.3.8) cambia de forma adaptativa dentro de la evolución del algoritmo de la siguiente manera:

$$SP_t = \frac{1}{1 + e^{1-(t_{max}/t+1)^\tau}}, \quad (3.3.9)$$

donde t_{max} es el número máximo de iteraciones y t la iteración actual. τ es una constante positiva.

Dentro del operador de cruce se genera un vector de prueba mediante la combinación de la información de la partícula actual $X_{i,t}$ y el vector mutado $V_{i,t+1}$, siguiendo el cruce binomial (ver Ecuación (2.2.5)). La selección de la partícula que sobrevive a la siguiente generación se lleva a cabo mediante la competición entre la partícula actual $X_{i,t}$ y el vector mutado $V_{i,t+1}$, el que tenga mejor valor de ajuste de acuerdo a la función objetivo, es seleccionado para permanecer dentro de la siguiente iteración.

Cuando una partícula permanece estancada sobre un número máximo de iteraciones (por ejemplo 5), los valores de su correspondiente factor de escala F_i y probabilidad de cruce Cr_i son reiniciados con el objetivo de incrementar la diversidad. La forma en la cual se reinician los valores de estos parámetros, es la siguiente:

$$F_{i,t+1} = \begin{cases} F_{i,t} & \text{si } NS_i < NS_{max} \\ F_l + rand[0, 1] \cdot (F_u - F_l) & \text{otro caso} \end{cases}, \quad (3.3.10)$$

$$Cr_{i,t+1} = \begin{cases} Cr_{i,t} & \text{si } NS_i < NS_{max} \\ Cr_l + rand[0, 1] \cdot (Cr_u - Cr_l) & \text{otro caso} \end{cases}, \quad (3.3.11)$$

donde F_i y Cr_i son el factor de escala y la probabilidad de cruce respectivamente, de la partícula $X_{i,G}$ en la iteración t . $F_l = 0.1$ y $F_u = 0.8$ son las fronteras inferior y superior respectivamente del factor de escala. $Cr_l = 0.3$ y $Cr_u = 1.0$ son las fronteras inferior y superior respectivamente de la probabilidad de cruce. $rand[0, 1]$ es un número aleatorio seleccionado de forma aleatoria dentro del intervalo $[0, 1]$. NS_i es un contador de estancamiento para cada partícula, y NS_{max} es el número máximo de iteraciones con estancamiento.

Para evitar estancamiento, el algoritmo DEPSO selecciona de forma aleatoria subdimensiones de los individuos dentro de la población no élite (Q) en los cuales $NS_i < NS_{max}$, para reiniciarlas de la siguiente manera:

$$X_{i,t}^j = \begin{cases} X_{min}^j + rand(0, 1) \cdot (X_{max}^j - X_{min}^j) & \text{si } rand(0, 1)_j \leq \gamma, \quad j = 1, 2, \dots, N \\ X_{i,t}^j & \text{otro caso} \end{cases}, \quad (3.3.12)$$

donde $rand(0, 1)_j$ es un número aleatorio seleccionado de forma aleatoria dentro del intervalo $[0, 1]$. γ es un valor de ajuste de probabilidad. X_{min}^j y X_{max}^j son los límites inferior y superior respectivamente, de la variable j .

De forma general, el algoritmo DEPSO logra ser superior a otras variantes del algoritmo DE en distintos problemas de optimización. El rendimiento de este algoritmo se debe a que mantiene un buen equilibrio entre exploración y explotación, gracias al uso de la estrategia de mutación auto-adaptativa, en la cual, el esquema “DE/e-rand/1” tiene mejores habilidades de exploración, mientras que, el esquema de mutación del algoritmo PSO consigue mejores habilidades de convergencia. Con esta estrategia, la población consigue mantener una buena diversidad en las primeras etapas del proceso evolutivo, y una convergencia más rápida hacia las últimas etapas del mismo.

En la Tabla 3.3.1 se muestran diversos algoritmos relacionados con la mejora en el rendimiento del algoritmo DE.

3.4. Discusión

En este capítulo se presentaron los trabajos más relevantes respecto al análisis numérico del modelo de texturas con cuatro ceros, al igual que el estado del arte de las aplicaciones de los ABI dentro de la FAE, y los principales trabajos respecto a la mejora en el rendimiento del algoritmo DE. De la revisión de la literatura presentada en este capítulo, se puede observar que, aunque existan distintos enfoques de estudio del modelo de texturas de cuatro ceros como se presenta en la

Tabla 3.3.1: Resumen de los trabajos más importantes del estado del arte relacionados con la mejora en el rendimiento del algoritmo DE.

Autor	Algoritmo	Mejora
Qin et al., 2009 [109]	SaDE	Nueva estrategia de mutación, Control de parámetros
Zhang y Sanderson, 2009 [105]	JADE	Nueva estrategia de mutación
Wang et al., 2011 [106]	CoDE	Nueva estrategia de mutación
Tanabe y Fukunaga, 2013 [108]	SHADE	Control de parámetros
Kumar y Singh, 2018 [51]	Hibridación entre ANN y DE	Hibridación del algoritmo DE con otros algoritmos de IC
Zhang et al., 2018 [111]	coACO	Hibridación del algoritmo DE con otros algoritmos de IC
Song et al., 2018 [112]	GPSO	Hibridación del algoritmo DE con otros algoritmos de IC
Wang et al., 2019 [19]	DEPSO	Hibridación del algoritmo DE con otros algoritmos de IC
Propuesta, 2024	HE-DEPSO	Nueva estrategia de mutación, Control de parámetros

Tabla 3.1.1, la validación del modelo de texturas con cuatro ceros de acuerdo a los últimos datos experimentales sigue siendo un aspecto importante dentro de la FAE. Además, se encontró que dentro del análisis numérico realizado en estos trabajos, es común el uso del criterio χ^2 como parte del ajuste entre las expresiones teóricas y los datos experimentales, sin embargo, en la mayoría de estos trabajos no se menciona de forma concisa el método o algoritmo utilizado para optimizar dicho criterio de ajuste. En el caso de [95], si bien se reporta el método utilizado, el autor advierte que el algoritmo puede depender de la selección de un punto de inicio óptimo, quedar estancado fácilmente en mínimos locales, e inclusive puede requerir de un largo periodo de cómputo. En vista de esto, se ve una oportunidad nicho en incorporar estrategias de optimización distintas como los ABI dentro del análisis numérico del modelo de texturas de cuatro ceros.

Dentro de la segunda sección de este capítulo se mostraron trabajos del estado del arte de las aplicaciones de los ABI dentro de la FAE. De forma general, es posible observar que el número de trabajos es pequeño, lo cual demuestra la poca explo-

ración de estos algoritmos en esta área de la física, y se identifica la oportunidad de expandir el uso de estas técnicas. Por otro lado, se encontró que la gran parte de las aplicaciones recaen en la optimización del diseño de aceleradores (ver Tabla 3.2.1), y que en tiempos más recientes, el uso de los algoritmos DE y PSO se ha visto más presente. Esto principalmente a su facilidad de implementación y robustez.

Finalmente, en la última sección de este capítulo se presentaron distintos algoritmos que han tratado de mejorar el rendimiento del algoritmo DE, principalmente en problemas de optimización con parámetros reales con un solo objetivo y con restricciones de frontera, tal como los representa el problema de optimización del criterio χ^2 . De la literatura revisada, se encontró que las variantes más representativas del algoritmo DE han mejorado el control de parámetros y la estrategia de mutación, tal es el caso de los algoritmos SHADE y JADE. Recientemente, se observa una mayor tendencia hacia la hibridación con otros algoritmos de IC, entre los cuales destaca la hibridación con el algoritmo PSO.

En el siguiente capítulo, se presenta un nuevo algoritmo llamado HE-DEPSO, el cual se encuentra inspirado en los principios del algoritmo DEPSO y SHADE.

Capítulo 4

Algoritmo propuesto HE-DEPSO

En ese capítulo se describe de forma detallada el algoritmo propuesto denominado Algoritmo de Evolución Diferencial basado en Optimización por Enjambre de Partículas con Información Élite Histórica (HE-DEPSO).

4.1. Motivación

A pesar de la existencia de diversas versiones avanzadas del algoritmo DE propuestas dentro de los últimos años, el rendimiento del algoritmo DE sigue presentando dificultades en problemas de optimización con múltiples mínimos locales. Para hacer frente a estos problemas, el diseño de operadores de mutación efectivos y el control de parámetros representan dos puntos clave para mejorar el rendimiento del algoritmo DE. Con este fin, en el algoritmo propuesto HE-DEPSO se desarrolla un operador de mutación híbrido adaptativo el cual toma como base la estrategia de mutación auto-adaptativa propuesta dentro del algoritmo DEPSO [19] e incorpora el control de parámetros del algoritmo SHADE [108]. Dentro de este operador de mutación híbrido adaptativo, se propone una nueva estrategia de mutación llamada “DE/current-to-EHE/1”, la cual utiliza la información histórica de los individuos elite de la población.

4.2. Operador de mutación híbrido adaptativo

Con el fin de lograr un mejor balance entre exploración y explotación, el operador de mutación del algoritmo HE-DEPSO adopta un mecanismo dual en el cual se selecciona de forma adaptativa entre dos estrategias de mutación. En primer lugar, se presenta una nueva estrategia de mutación denominada “DE/current-to-EHE/1”, la cual se encuentra orientada a mejorar el balance entre las habilidades

de exploración y explotación dentro de las primeras etapas del proceso evolutivo. Por otro lado, con el propósito de mejorar la capacidad de explotación dentro de las etapas más avanzadas del proceso evolutivo, se incorpora de forma similar a la estrategia de mutación auto-adaptativa del algoritmo DEPSO (ver Ecuación (3.3.8)), el esquema de mutación del algoritmo PSO (ver Ecuación (2.2.13)).

A diferencia del esquema “DE/e-rand/1” utilizado dentro de la estrategia de mutación auto-adaptativa del algoritmo DEPSO, en la cual, solo se utiliza la información de los individuos de la iteración actual, la estrategia “DE/current-to-EHE/1”, utiliza la información histórica de los individuos elite y obsoletos del proceso evolutivo para mejorar la capacidad de exploración del algoritmo. Esta estrategia es descrita a continuación.

4.2.1. Estrategia “DE/current-to-EHE/1”

Dentro de los algoritmos evolutivos, los mejores individuos de la población también conocidos como individuos elite conservan información evolutiva valiosa para guiar a la población hacia regiones prometedoras [19, 113–115]. Sin embargo, gran parte de las propuestas realizadas utilizan la información de los individuos elite de la iteración actual, olvidando por completo la información de los individuos elite anteriores. Esta ausencia de información dentro de iteraciones posteriores, puede acotar la capacidad que tienen nuevos individuos de explorar el espacio de búsqueda. Debido a esto, se recurre a la información histórica de evolución para mejorar la capacidad de exploración de los nuevos individuos.

Antes de ser aplicado el operador de mutación, todos los individuos en la población actual P_t son reordenados de forma ascendente de acuerdo a su valor de ajuste. Una vez reordenada la población, dos particiones de individuos son generadas, la primer partición denotada por E_t , está conformada por los top $pb\% \in [0.1, 1]$ mejores individuos o individuos elite. La segunda partición, denotada por NE_t agrupa a todos los individuos no elite, es decir, $(NP - NP \cdot pb\%)$. Es importante notar que $E_t \cup NE_t = P_t$.

A diferencia de otras estrategias de mutación actuales [19, 105, 116], en las cuales solo se utilizan individuos elite correspondientes a la iteración actual, en esta es-

trategia se hace uso del historial de evolución mediante la incorporación de dos archivos externos de tamaño NP . El primer archivo denotado por **HE**, almacena en cada iteración a los individuos elite pertenecientes a la partición E_t . Si el tamaño del archivo **HE** es mayor que NP entonces se reajusta su tamaño. El segundo archivo utilizado, se denota por **HL** y almacena a los individuos obsoletos (individuos descartados dentro del proceso de selección) y es actualizado en cada generación. De forma similar al archivo **HE**, si el tamaño del archivo **HL** es mayor que NP entonces se reajusta su tamaño. Con la información de los individuos pertenecientes a E_t y **HE** se puede conformar un grupo de candidatos guía $E_t \cup \text{HE}$ para mutar individuos en la población. Por otra parte, los individuos de NE_t y **HL** conforman un grupo de individuos $\text{NE}_t \cup \text{HL}$ cuya información contribuye a la exploración del espacio de búsqueda. La forma en la cual se genera un vector mutado mediante la estrategia “DE/current-to-EHE/1”, es de la siguiente manera:

$$V_{i,t} = X_{i,t} + F_i \cdot (X_{E_r,t} - X_{i,t}) + F_i \cdot (X_{P_r,t} - X_{HL_r,t}), \quad (4.2.1)$$

donde $X_{i,t}$ es el i -ésimo individuo de la iteración t , $X_{E_r,t}$ es un individuo seleccionado de forma aleatoria entre los individuos del grupo $E_t \cup \text{HE}$, $X_{P_r,t}$ es un individuo seleccionado de forma aleatoria dentro de la población actual P_t , $X_{HL_r,t}$ es un individuo seleccionado de forma aleatoria de los individuos que conforman el grupo $\text{NE}_t \cup \text{HL}$, y F_i es el factor de escala correspondiente al i -ésimo individuo.

Siguiendo a la Ecuación (4.2.1), es posible observar que la estrategia “DE/current-to-EHE/1” puede ayudar al algoritmo HE-DEPSO a mantener una buena capacidad de exploración, y dirigir a los individuos mutados hacia regiones prometedoras sin propiciar el estancamiento en mínimos locales. Esto se puede explicar debido a las siguientes razones:

- El uso de un individuo seleccionado de forma aleatoria dentro del grupo $E_t \cup \text{HE}$, es decir $X_{E_r,t}$, contribuye a guiar a los individuos mutados hacia regiones más prometedoras. Sin embargo, debido a la presencia de la información histórica de los individuos elite (**HE**), se evita que los individuos mutados se dirijan de manera boráz hacia las mejores regiones encontradas,

permitiendo así, mantener una buena diversidad en la población y aumentar las posibilidades de dirigirse a regiones óptimas.

- La participación de dos individuos, $X_{P_r,t}$ y $X_{L_r,t}$, seleccionados de forma aleatoria de P_t y **HL** respectivamente, promueve la diversidad de los individuos mutados. Por consecuencia, la diversidad de búsqueda del algoritmo HE-DEPSO es mejorada considerablemente, lo cual es beneficioso para escapar de posibles mínimos locales.

Al comienzo de la ejecución del algoritmo HE-DEPSO, los archivos **HE** y **HL** se definen vacíos, y a través del proceso evolutivo, almacenan a los individuos elite y obsoletos respectivamente. Al analizar el efecto del parámetro $pb\%$, el cual determina el porcentaje de individuos dentro de la partición E_t , se encontró que valores cercanos a 0.9 promueven la participación de un mayor número de individuos elite, provocando una mayor diversidad en la población. Sin embargo, este incremento en la diversidad puede afectar a la capacidad de convergencia del algoritmo HE-DEPSO. Por otro lado, valores cercanos a 0.1 pueden restringir el número de individuos elite a considerar, lo cual mejora la convergencia del algoritmo, pero puede provocar el estancamiento en mínimos locales.

Debido a lo anterior, dentro del algoritmo HE-DEPSO se opta por actualizar en cada iteración el valor de $pb\%$ siguiendo un decrecimiento lineal dado de la siguiente manera:

$$pb\% = p_{max} - \frac{t \cdot (p_{max} - p_{min})}{t_{max}}, \quad (4.2.2)$$

donde t hace referencia a la iteración actual, t_{max} es el número máximo de iteraciones, p_{min} y p_{max} son los valores mínimos y máximos del intervalo asignado para el porcentaje de individuos dentro de la partición E_t . En este trabajo se ha optado por utilizar los valores $p_{max} = 0.4$ y $p_{min} = 0.1$. De esta forma, se disminuye la sensibilidad del parámetro $pb\%$ y a la vez, se logra mantener un buen balance entre explotación y exploración.

4.2.2. Selección de estrategia de mutación para el operador de mutación híbrido adaptativo

El operador de mutación híbrido adaptativo implementado en el algoritmo HE-DEPSO, utiliza dos estrategias de mutación las cuales tienen el objetivo de mejorar las habilidades de exploración y explotación del algoritmo HE-DEPSO en distintas etapas del proceso de evolutivo. De forma concreta, para cada individuo $X_{i,t}$ en la población, este operador genera un vector mutado $V_{i,t}$ de la siguiente forma:

$$V_{i,t} = \begin{cases} X_{i,t} + F_i \cdot (X_{Er,t} - X_{i,t}) + F_i \cdot (X_{Pr,t} - X_{HLr,t}) & \text{si } rand[0,1] < \alpha_t \\ \omega \cdot X_{i,t} + c_1 \cdot r_1 \cdot (X_{gbest,t} - X_{i,t}) \\ + c_2 \cdot r_2 \cdot (X_{pbest,t} - X_{i,t}) & \text{otro caso} \end{cases}, \quad (4.2.3)$$

donde $rand[0,1]$ es un número aleatorio seleccionado de forma uniforme dentro del intervalo $[0,1]$, y α_t representa la probabilidad de seleccionar entre la estrategia de mutación “DE/current-to-EHE/1” y la estrategia de mutación adoptada del algoritmo PSO (ver Ecuación (2.2.13)).

La probabilidad de selección α_t , es actualizada en cada iteración de acuerdo a la Ecuación (3.3.9) tomando un valor de $\tau = 1.8$. En la Figura 4.2.1 se muestra la curva de selección de estrategia de mutación que sigue el operador de mutación híbrido adaptativo del algoritmo HE-DEPSO. En la Figura 4.2.1a se ilustra en color rojo, la curva de selección de probabilidad descrita por la Ecuación (3.3.9) con un valor de $\tau = 1.8$, para un número máximo de iteraciones t_{max} . Por otra parte, en la Figura 4.2.1b se muestra a modo de ejemplo, una posible distribución de ocurrencias entre las cuales es seleccionada la estrategia de mutación “DE/current-to-EHE/1” (cuadrados en color azul) o la estrategia adoptada del algoritmo PSO (círculos en color magenta)). De acuerdo con estas gráficas, se puede observar que dentro de las primeras etapas de evolución del algoritmo HE-DEPSO, la probabilidad de selección α_t tiende hacia valores cercanos a 1, lo cual aumenta la probabilidad de que la estrategia de mutación “DE/current-to-EHE/1” sea seleccionada, y se aproveche el equilibrio entre exploración y explotación que proporciona esta estrategia.

Por otra parte, hacia las etapas más avanzadas de la evolución del algoritmo, la estrategia de mutación adoptada del algoritmo PSO es seleccionada con mayor frecuencia, esto implica que, durante las etapas más avanzadas existe una mayor probabilidad de que el algoritmo HE-DEPSO incremente su capacidad de explotación debido a un uso más frecuente de la información de las mejores posiciones de cada individuo, así como la información de la posición del mejor individuo de la población.

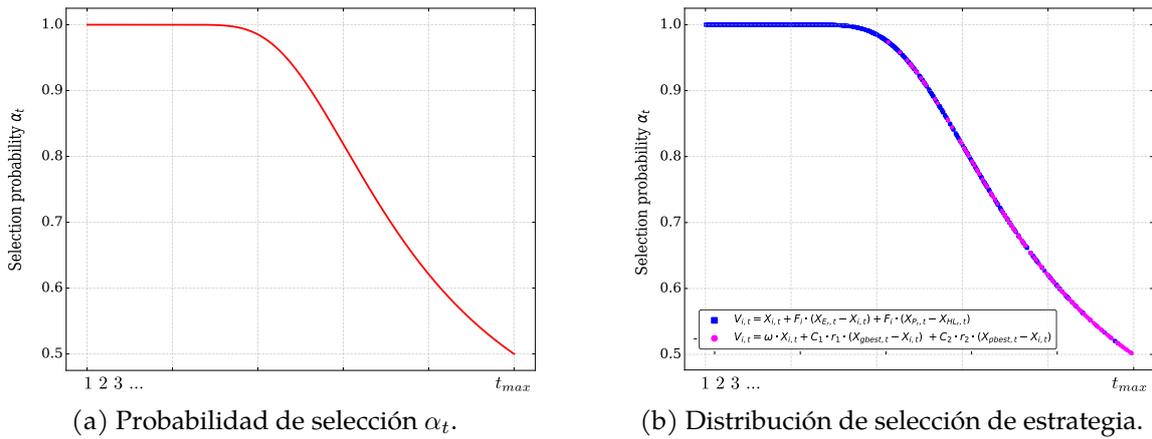


Figura 4.2.1: Curva de selección de estrategia.

4.3. El algoritmo HE-DEPSO completo

El algoritmo HE-DEPSO sigue de forma general la estructura usual del algoritmo DE, en la cual, se tienen las etapas de mutación, cruce y selección. Después de generar un vector mutado $V_{i,t}$ para el i -ésimo individuo de la población de acuerdo a lo descrito en la sección anterior, se procede a generar un vector de prueba $U_{i,t}$ para el individuo $X_{i,t}$ utilizando el cruce binomial (Ecuación (2.2.5)). Con el fin de reducir la sensibilidad del control de los parámetros F y C_r , el algoritmo HE-DEPSO toma ventaja del control de parámetros adaptativo a nivel individuo del algoritmo SHADE [108], para generar la configuración de parámetros de F y C_r para cada individuo. Finalmente, en la etapa de selección se identifican a los individuos que conformarán a la población para la siguiente iteración, esto se realiza de acuerdo a lo comentado dentro de la Subsección 2.2.2.3. Basado en la descripción anterior, el pseudocódigo del algoritmo HE-DEPSO es reportado en

el Algoritmo 4.3.1.

Algoritmo 4.3.1: HE-DEPSO

```

Entrada: Tamaño de la población:  $NP$ , Número máximo de iteraciones:  $t_{max}$ ,
Número de dimensiones:  $D$ 
// Inicialización
1: Numero de iteración:  $t = 0$ 
2: Inicializar aleatoriamente la población:  $P_0 = \{X_{t,0}\}, i = 1, 2, \dots, NP$ 
3: Evaluar el valor de ajuste de la población inicial:  $\{f_{i,0}\}$ 
4: Inicializar todos los elementos en  $M_F$  y  $M_{Cr}$  a 0.8 y 0.5 respectivamente
5: Inicializar  $HE = \emptyset, HL = \emptyset$ 
6: Inicializar  $p_{min} = 0.1, p_{max} = 0.25, k = 0, c1 = c2 = 2$ 
7: Inicializar pbest y gbest
// Ciclo principal
8: mientras  $t < t_{max}$  hacer
9:    $S_{Cr} = \emptyset; S_F = \emptyset$ 
10:  Actualizar pb %,  $\alpha_t$  y  $\omega$ 
11:  Ordenar individuos en  $P_t$ 
12:  Generar  $E_t$  y  $NE_t$  a partir de  $\{100pb\%$  y  $\{NP - (100pb\%)\}$  respectivamente
13:  Añadir directamente  $E_t$  a  $HE$  (Reajustar el tamaño de  $HE$  si es necesario)
14:  Seleccionar de forma aleatoria un par de valores de  $M_F$  y  $M_{Cr}$  para generar  $F$  y  $Cr$  para todos
    los individuos
15:  para  $i = 1 : NP$  hacer
16:    Seleccionar de forma aleatoria un índice de  $[1, NP]$  para obtener el asociado  $F_{i,t}$  y  $Cr_{i,t}$  de
       $X_{i,t}$ 
17:    Seleccionar de forma aleatoria un individuo  $X_{Er,t}$  de  $E_t \cup HE$ 
18:    Seleccionar de forma aleatoria un individuo  $X_{Pr,t}$  de  $P_t$ 
19:    Seleccionar de forma aleatoria un individuo  $X_{HLr,t}$  de  $NE_t \cup HL$ 
20:    Generar un número aleatorio de forma uniforme  $rand_i$  dentro del intervalo  $[0, 1]$ 
21:    Generar el vector mutado  $V_{i,t}$  con:
22:    si  $rand_i < \alpha_t$  entonces
23:       $V_{i,t} = X_{i,t} + F_i \cdot (X_{Er,t} - X_{i,t}) + F_i \cdot (X_{Pr,t} - X_{HLr,t})$ 
24:    en otro caso
25:       $V_{i,t} = \omega \cdot X_{i,t} + c1 \cdot r1 \cdot (X_{gbest,t} - X_{i,t}) + c2 \cdot r2 \cdot (X_{pbest,t} - X_{i,t})$ 
26:    fin
27:    Generar el vector de prueba  $U_{i,t}$  usando el cruce binomial y calcular su valor de ajuste
28:    si  $f(U_{i,t}) < f(X_{i,t})$  entonces
29:       $X_{i,t+1} = U_{i,t}$ 
30:       $F_{i,t} \rightarrow S_F; Cr_{i,t} \rightarrow S_{Cr}$ 
31:    en otro caso
32:      Añadir directamente  $X_{i,t}$  a  $HL$  (Reajustar el tamaño de  $HL$  si es necesario)
33:       $X_{i,t+1} = X_{i,t}$ 
34:    fin
35:    si  $f(U_{i,t}) < f(X_{pbest,i})$  entonces
36:       $X_{pbest,i} = U_{i,t}$ 
37:    fin
38:    si  $f(U_{i,t}) < f(X_{gbest,i})$  entonces
39:       $X_{gbest,i} = U_{i,t}$ 
40:    fin
41:  fin
42:  Actualizar  $M_{Cr}$  y  $M_F$ 
43:   $k = k + 1$  (Si  $k > NP \rightarrow k = 1$ )
44:   $t = t + 1$ 
45: fin
Salida :  $gbest$  y  $f(gbest)$ 

```

4.4. Resultados experimentales y análisis

En esta sección se verifica el rendimiento del algoritmo propuesto HE-DEPSO al resolver diferentes problemas de optimización. En primer lugar, se presenta el con-

junto de problemas de prueba utilizado, posteriormente, se dan a conocer los algoritmos seleccionados para llevar a cabo comparaciones, así como la configuración de sus parámetros. Después, el rendimiento del algoritmo HE-DEPSO es comparado contra los algoritmos seleccionados dentro del conjunto de problemas de prueba. Finalmente, se presenta una discusión de los resultados obtenidos.

4.4.1. Funciones de prueba

Para validar el rendimiento del algoritmo HE-DEPSO al resolver distintos problemas de optimización, se utilizó el conjunto de funciones de prueba de un solo objetivo con restricciones de frontera CEC 2017 [117]. Este conjunto se encuentra conformado por un total de 29 funciones de prueba, de las cuales se conoce su óptimo global. Se tienen dos funciones unimodales (F_1 y F_3), siete funciones multimodales simples ($F_4 - F_{10}$), diez funciones híbridas ($F_{11} - F_{20}$) y diez funciones compuestas ($F_{21} - F_{30}$). En todos estos problemas de optimización, se busca encontrar el mínimo global (F_{min}) dentro del espacio de búsqueda delimitado en cada dimensión (D) por el intervalo $[-100, 100]$. En la Tabla 4.4.1 se presenta de forma breve, información acerca de este conjunto de funciones de prueba. Una descripción más detallada de estas funciones de prueba se presenta dentro del Apéndice A.

4.4.2. Algoritmos y configuración de parámetros

Para comparar el algoritmo HE-DEPSO, se seleccionaron seis algoritmos, incluidos los algoritmos PSO y DE, así como tres variantes representativas y avanzadas del algoritmo DE, denominadas: CoDE [106], SHADE [108] y DEPSO [19]. Entre estos algoritmos, CoDE y DEPSO incorporan modificaciones principalmente dentro del operador de mutación del algoritmo DE, mientras que SHADE modifica el control de parámetros F y Cr . La comparación del rendimiento del algoritmo HE-DEPSO con los algoritmos anteriormente mencionados se realizó en dos dimensiones diferentes, $D = 10$ y $D = 30$, sobre el conjunto de funciones de prueba CEC 2017.

Con el propósito de garantizar una comparación equitativa, la configuración de

Tipo de Función	Índice	Nombre de la función	Óptimo (F_{min})
Funciones	1	Shifted and Rotated Bent Cigar	100
Unimodales	3	Shifted and Rotated Zakharov	300
	4	Shifted and Rotated Rosenbrock	400
	5	Shifted and Rotated Rastrigin	500
Funciones	6	Shifted and Rotated Expanded Schaffer F6	600
Multimodales	7	Shifted and Rotated Lunacek Bi-Rastrigin	700
Simples	8	Shifted and Rotated Non-Continuous Rastrigin	800
	9	Shifted and Rotated Levy	900
	10	Shifted and Rotated Schwefel	1000
	11	Hybrid Problem 1 (N = 3)	1100
	12	Hybrid Problem 2 (N = 3)	1200
	13	Hybrid Problem 3 (N = 3)	1300
Funciones	14	Hybrid Problem 4 (N = 4)	1400
Híbridas	15	Hybrid Problem 5 (N = 4)	1500
	16	Hybrid Problem 6 (N = 4)	1600
	17	Hybrid Problem 7 (N = 5)	1700
	18	Hybrid Problem 8 (N = 5)	1800
	19	Hybrid Problem 9 (N = 5)	1900
	20	Hybrid Problem 10 (N = 6)	2000
	21	Composition Problem 1 (N = 3)	2100
	22	Composition Problem 2 (N = 3)	2200
	23	Composition Problem 3 (N = 4)	2300
Funciones	24	Composition Problem 4 (N = 4)	2400
Compuestas	25	Composition Problem 5 (N = 5)	2500
	26	Composition Problem 6 (N = 5)	2600
	27	Composition Problem 7 (N = 6)	2700
	28	Composition Problem 8 (N = 6)	2800
	29	Composition Problem 9 (N = 9)	2900
	30	Composition Problem 10 (N = 3)	3000

Tabla 4.4.1: Funciones de prueba CEC 2017.

los parámetros en común de todos los algoritmos fue asignada de la misma manera: el número máximo de iteraciones t_{max} es configurado a 1000, el tamaño de la población NP es configurado a 100, y el número de repeticiones independientes realizadas es 31. La configuración de los parámetros de cada algoritmo siguiendo los valores sugeridos por sus respectivos autores se presenta en la Tabla 4.4.2. Vale la pena mencionar que el algoritmo propuesto HE-DEPSO fue implementado en el lenguaje de programación Python en la versión 3.11 y que todos los algoritmos fueron ejecutados en la misma computadora con 8 GB de RAM y un procesador

de seis núcleos a 3.6 GHz.

Tabla 4.4.2: Configuración de parámetros de todos los algoritmos.

Algoritmo	Parámetros
HE-DEPSO	$p_{min} = 0.1, p_{max} = 0.4, \tau = 1.8, c1 = c2 = 2, \omega \in [0.4, 0.9],$ $H = NP, M_F(1 : H) = 0.5, M_{Cr}(1 : H) = 0.8$
DEPSO	$c1 = c2 = 2, \omega \in [0.4, 0.9], Cr[0.3, 1.0], F[0.1, 0.8],$ $NS_{max} = 5, \gamma = 0.001, \tau = 1.7, SEP = 0.4 \cdot NP$
SHADE	$H = NP, M_F(1 : H) = M_{Cr}(1 : H) = 0.5$
CoDE	1) $[F = 1.0, Cr = 0.1], 2) [F = 1.0, Cr = 0.9], 3) [F = 0.8, Cr = 0.2]$
DE	$Cr = 0.9, F = 0.5$
PSO	$\omega \in [0.4, 0.9], c1 = c2 = 2$

4.4.3. Comparación con DE, PSO y tres variantes avanzadas del algoritmo DE

Con el fin de evaluar el rendimiento del algoritmo HE-DEPSO, sus resultados de optimización y propiedades de convergencia son comparadas y analizadas respecto a los obtenidos por los algoritmos: DE, PSO, CoDE, SHADE y DEPSO, sobre el conjunto de funciones de prueba CEC 2017.

Las Tablas 4.4.3 y 4.4.4 reportan los resultados estadísticos de cada algoritmo en $D = 10$ y $D = 30$ respectivamente. La medida de error de solución $|F_{min} - F(X^*)|$, donde F_{min} es la solución conocida del problema y X^* es la mejor solución encontrada en cada repetición de cada algoritmo, fue utilizada para obtener estos resultados. En estas tablas se muestra la media y la desviación estándar (Std) de la medida del error de solución para cada función y algoritmo sobre las 31 repeticiones independientes, así como la clasificación alcanzada basada en el valor de media obtenido. Los mejores resultados son resaltados en negrita.

Para identificar diferencias estadísticas significativas entre los resultados obtenidos por el algoritmo HE-DEPSO y aquellos obtenidos por cada uno de los demás algoritmos, se realizó la prueba no paramétrica de suma de rangos de Wilcoxon con un nivel de confidencialidad de $\alpha = 0.05$. En estas tablas, la significación estadística relacionada al rendimiento del algoritmo HE-DEPSO es representada

mediante los símbolos “+”, “≈” y “-”, los cuales indican, que el rendimiento del algoritmo HE-DEPSO es “mejor/similar/peor” que el algoritmo a comparar. La fila “W/T/L” cuenta el número total de “+”, “≈” y “-”, respectivamente.

Tabla 4.4.3: Comparación de resultados entre HE-DEPSO, DE, PSO y tres variantes avanzadas del algoritmo DE, sobre el conjunto de funciones de prueba CEC 2017 con $D = 10$.

Función	Métrica	HE-DEPSO	DEPSO	SHADE	CoDE	DE	PSO
F_1	Media	0.0000E+00	1.8337E-15	0.0000E+00	1.1580E-10	3.5103E-05	1.1366E+10
	Std	0.0000E+00	4.8427E-15	0.0000E+00	6.0507E-11	2.0793E-05	9.7631E+09
	Clasificación	1	4	1	3	2	5
	Significación		≈	≈	+	+	+
F_3	Media	0.0000E+00	0.0000E+00	0.0000E+00	4.0392E-11	4.9147E-04	1.5964E+02
	Std	0.0000E+00	0.0000E+00	0.0000E+00	2.7390E-11	4.8441E-04	6.2297E+02
	Clasificación	1	1	1	2	3	4
	Significación		≈	≈	+	+	+
F_4	Media	0.0000E+00	2.3838E-14	2.0805E+00	2.3103E-09	1.0409E-03	5.8071E+01
	Std	0.0000E+00	2.8513E-14	6.7544E-01	1.2601E-09	6.4578E-04	6.9704E+01
	Clasificación	1	2	5	3	4	6
	Significación		+	+	+	+	+
F_5	Media	2.8439E+00	4.0119E+00	3.8489E+00	8.5594E+00	2.9710E+01	3.0728E+01
	Std	7.2094E-01	1.6940E+00	1.0401E+00	1.4246E+00	3.9355E+00	1.2073E+01
	Clasificación	1	3	2	4	5	6
	Significación		+	+	+	+	+
F_6	Media	0.0000E+00	4.6321E-03	2.7530E-04	4.6895E-03	4.5748E-05	1.1970E+01
	Std	0.0000E+00	2.5790E-02	5.5026E-04	2.4954E-03	2.2095E-05	9.6167E+00
	Clasificación	1	4	3	5	2	6
	Significación		+	+	+	+	+
F_7	Media	1.2678E+01	2.1067E+01	1.3953E+01	2.0678E+01	3.9143E+01	3.0657E+01
	Std	7.0313E-01	6.7134E+00	7.6551E-01	2.2560E+00	3.7389E+00	8.6791E+00
	Clasificación	1	4	2	3	6	5
	Significación		+	+	+	+	+
F_8	Media	2.2853E+00	4.2045E+00	3.8070E+00	8.6884E+00	2.6037E+01	2.4833E+01
	Std	8.1114E-01	1.9346E+00	7.3957E-01	1.8765E+00	5.5290E+00	1.1493E+01
	Clasificación	1	3	2	4	6	5
	Significación		+	+	+	+	+
F_9	Media	0.0000E+00	7.3346E-15	0.0000E+00	0.0000E+00	0.0000E+00	6.8233E+01
	Std	0.0000E+00	2.8391E-14	0.0000E+00	0.0000E+00	0.0000E+00	1.9812E+02
	Clasificación	1	2	1	1	1	3
	Significación		≈	≈	≈	+	+
F_{10}	Media	2.4499E+02	8.9742E+02	1.5812E+02	7.4438E+02	1.4730E+03	9.1356E+02
	Std	1.1864E+02	1.6890E+02	7.6565E+01	1.1745E+02	1.7509E+02	3.3829E+02
	Clasificación	2	4	1	3	6	5
	Significación		+	-	+	≈	+
F_{11}	Media	2.5687E-07	8.6658E-01	2.3208E+00	1.2502E+00	6.7909E+00	8.1368E+01
	Std	1.4103E-06	6.1559E-01	4.7869E-01	4.6467E-01	9.9142E-01	8.3317E+01
	Clasificación	1	2	4	3	5	6
	Significación		+	+	+	+	+

Continúa en la página siguiente

Tabla 4.4.3. *Continúa de la página anterior*

Función	Métrica	HE-DEPSO	DEPSO	SHADE	CoDE	DE	PSO
F_{12}	Media	8.1966E+01	1.1535E+01	3.5298E+01	2.7652E+02	5.3458E+02	8.7444E+07
	Std	7.0615E+01	2.8045E+01	1.4666E+01	6.4786E+01	1.2345E+02	3.8318E+08
	Clasificación	3	1	2	4	5	6
	Significación		–	–	+	+	+
F_{13}	Media	3.8882E+00	4.6716E+00	5.9948E+00	1.1231E+01	1.2796E+01	6.3448E+03
	Std	2.4042E+00	9.7586E-01	6.0682E-01	2.0901E+00	1.7030E+00	1.2009E+04
	Clasificación	1	2	3	4	5	6
	Significación		≈	+	+	+	+
F_{14}	Media	2.4091E+00	6.8868E+00	1.9065E+01	2.0295E+01	2.6185E+01	1.7551E+02
	Std	5.1573E+00	8.0249E+00	3.0340E+00	1.2717E-01	1.2365E+00	9.5615E+01
	Clasificación	1	2	3	4	5	6
	Significación		+	+	+	+	+
F_{15}	Media	3.6646E-01	5.7132E-01	1.0838E+00	1.3990E+00	2.1239E+00	4.6041E+02
	Std	2.3145E-01	4.9456E-01	3.3723E-01	5.4455E-01	1.2934E+00	1.0508E+03
	Clasificación	1	2	3	4	5	6
	Significación		+	+	+	+	+
F_{16}	Media	1.4450E+00	1.6357E+00	4.5365E+00	4.3009E+00	1.9028E+01	1.2496E+02
	Std	3.2202E-01	3.2942E+00	2.2215E+00	2.5242E+00	1.6484E+01	1.0780E+02
	Clasificación	1	2	4	3	5	6
	Significación		–	+	+	+	+
F_{17}	Media	7.2016E+00	1.2075E+01	2.1001E+01	2.0112E+01	2.8168E+01	5.7710E+01
	Std	3.0221E+00	8.5311E+00	3.2211E+00	2.8326E+00	1.8397E+00	2.4017E+01
	Clasificación	1	2	4	3	5	6
	Significación		+	+	+	+	+
F_{18}	Media	3.8770E-01	8.2949E+00	1.9823E+01	2.0117E+01	2.0542E+01	2.7241E+04
	Std	1.4583E-01	6.7399E+00	1.9348E+00	3.4569E-02	1.0532E-01	1.8731E+04
	Clasificación	1	2	3	4	5	6
	Significación		+	+	+	+	+
F_{19}	Media	9.7895E-02	5.4625E-01	1.0349E+00	5.9210E-01	1.2208E+00	5.0978E+03
	Std	1.2871E-01	3.7564E-01	1.5715E-01	4.4716E-02	2.0310E-01	1.1584E+04
	Clasificación	1	2	4	3	5	6
	Significación		+	+	+	+	+
F_{20}	Media	5.2046E+00	6.9310E+00	1.7787E+01	2.0000E+01	2.5137E+01	5.9825E+01
	Std	3.1153E+00	6.8469E+00	3.9314E+00	1.4932E-04	1.9501E+00	4.0129E+01
	Clasificación	1	2	3	4	5	6
	Significación		≈	+	+	+	+
F_{21}	Media	1.7676E+02	1.0695E+02	1.5575E+02	1.2554E+02	1.9598E+02	1.9045E+02
	Std	4.6029E+01	2.0972E+01	4.2795E+01	4.8087E+01	5.7606E+01	6.2148E+01
	Clasificación	4	1	3	2	6	5
	Significación		–	–	–	+	+
F_{22}	Media	1.0000E+02	9.7428E+01	9.9647E+01	7.4280E+01	1.0070E+02	1.2711E+02
	Std	0.0000E+00	1.5939E+01	2.1997E+00	4.4524E+01	5.7442E-01	5.6792E+01
	Clasificación	4	2	3	1	5	6
	Significación		+	+	≈	≈	+
F_{23}	Media	3.0613E+02	3.0845E+02	3.0859E+02	3.2072E+02	3.3600E+02	3.4479E+02
	Std	1.4783E+00	2.1842E+00	1.4096E+00	3.4098E+00	5.2784E+00	1.6592E+01
	Clasificación	1	2	3	4	5	6
	Significación		+	+	+	+	+

Continúa en la página siguiente

Tabla 4.4.3. *Continúa de la página anterior*

Función	Métrica	HE-DEPSO	DEPSO	SHADE	CoDE	DE	PSO
F_{24}	Media	3.2397E+02	2.6066E+02	3.0017E+02	3.0685E+02	3.6119E+02	3.7743E+02
	Std	4.1615E+01	1.1813E+02	6.5109E+01	9.2286E+01	4.7582E+00	1.5025E+01
	Clasificación	4	1	2	3	5	6
	Significación		≈	–	≈	+	+
F_{25}	Media	4.1869E+02	4.2783E+02	4.0980E+02	3.9783E+02	4.0247E+02	4.7998E+02
	Std	2.3095E+01	2.2521E+01	2.0463E+01	1.5694E-01	1.3981E+01	6.1710E+01
	Clasificación	4	5	3	1	2	6
	Significación		≈	–	–	+	+
F_{26}	Media	3.0304E+02	3.0849E+02	3.0085E+02	2.9625E+02	3.8419E+02	5.6150E+02
	Std	1.1775E+01	3.2854E+01	1.7405E+01	6.3643E+01	1.6020E+02	2.5242E+02
	Clasificación	3	4	2	1	5	6
	Significación		+	≈	+	≈	+
F_{27}	Media	3.7584E+02	3.7153E+02	3.7683E+02	4.1575E+02	3.9004E+02	4.1060E+02
	Std	2.3098E+01	5.3891E-01	7.7935E+00	1.8031E+01	3.4010E+01	2.0006E+01
	Clasificación	2	1	3	5	4	6
	Significación		≈	+	+	+	+
F_{28}	Media	4.7251E+02	4.7218E+02	4.7338E+02	4.7816E+02	4.7253E+02	6.1933E+02
	Std	3.0463E-02	1.8112E+00	2.7252E+00	6.1120E+00	1.6134E-01	1.5566E+02
	Clasificación	2	1	4	5	3	6
	Significación		+	+	+	≈	+
F_{29}	Media	2.3616E+02	2.3508E+02	2.4658E+02	2.5569E+02	2.8284E+02	3.4235E+02
	Std	6.1991E+00	8.2538E+00	6.1477E+00	9.0010E+00	1.4685E+01	9.4838E+01
	Clasificación	2	1	3	4	5	6
	Significación		≈	+	+	+	+
F_{30}	Media	2.0061E+02	2.0113E+02	2.0227E+02	2.0313E+02	2.0320E+02	1.6044E+06
	Std	1.5718E-01	5.2187E-01	3.8772E-01	4.7938E-01	8.5716E-01	2.1981E+06
	Clasificación	1	2	3	4	5	6
	Significación		+	+	+	+	+
Clasificación promedio		1.68	2.27	2.75	3.24	4.48	5.48
Clasificación final		1	2	3	4	5	6
W/T/L		-/-/-	17/9/3	20/4/5	24/3/2	25/4/0	29/0/0

Tabla 4.4.4: Comparación de resultados entre HE-DEPSO, DE, PSO y tres variantes avanzadas del algoritmo DE, sobre el conjunto de funciones de prueba CEC 2017 con $D = 30$.

Función	Métrica	HE-DEPSO	DEPSO	SHADE	CoDE	DE	PSO
F_1	Media	1.6045E-14	2.0034E-04	2.3598E+03	1.2860E+08	2.0817E+09	6.3600E+10
	Std	6.0758E-15	4.9086E-04	1.2985E+03	3.5599E+07	7.3074E+08	5.8100E+10
	Clasificación	1	2	3	4	5	6
	Significación		+	+	+	+	+
F_3	Media	1.4486E-13	1.3343E+02	6.1189E+04	7.0539E+04	4.1473E+05	2.1200E+04
	Std	4.1092E-14	8.8712E+01	2.2749E+04	2.6622E+04	1.3053E+05	2.9800E+04
	Clasificación	1	2	4	5	6	3
	Significación		+	+	+	+	+
F_4	Media	2.8405E+00	1.2605E+01	7.7040E+01	9.6061E+01	3.5736E+01	1.7300E+03
	Std	2.1842E+00	8.6176E+00	3.8595E+00	2.0503E+01	3.2288E+00	1.3000E+03
	Clasificación	1	2	4	5	3	6
	Significación		+	+	+	+	+
F_5	Media	2.3850E+01	1.8175E+02	9.3463E+01	1.8418E+02	2.4169E+02	1.8500E+02
	Std	5.0701E+00	1.2845E+01	1.0871E+01	1.2630E+01	1.2302E+01	4.4900E+01
	Clasificación	1	3	2	4	6	5
	Significación		+	+	+	+	+
F_6	Media	4.6793E-01	5.7971E-03	3.7760E+01	5.8751E+01	2.4651E+01	5.3800E+01
	Std	3.5101E-01	2.9636E-02	5.8394E+00	5.2672E+00	3.3305E+00	1.2000E+01
	Clasificación	2	1	4	6	3	5
	Significación		-	+	+	+	+
F_7	Media	5.2487E+01	2.3377E+02	1.1601E+02	2.1572E+02	2.6887E+02	2.9100E+02
	Std	3.3716E+00	1.4172E+01	8.0551E+00	1.2272E+01	1.3070E+01	1.1300E+02
	Clasificación	1	4	2	3	5	6
	Significación		+	+	+	+	+
F_8	Media	2.0726E+01	1.7779E+02	9.3771E+01	1.8161E+02	2.3844E+02	1.7100E+02
	Std	4.6475E+00	1.1810E+01	9.9764E+00	1.3211E+01	1.6657E+01	3.8700E+01
	Clasificación	1	4	2	5	6	3
	Significación		+	+	+	+	+
F_9	Media	1.1236E+00	2.8880E-03	7.6008E+02	4.1017E+03	4.8783E+02	2.8400E+03
	Std	9.0322E-01	1.6080E-02	2.7907E+02	8.3155E+02	1.5998E+02	1.2400E+03
	Clasificación	2	1	4	6	3	5
	Significación		-	+	+	+	+
F_{10}	Media	2.6526E+03	6.7636E+03	3.4600E+03	6.3166E+03	7.9631E+03	5.2400E+03
	Std	3.0794E+02	2.5866E+02	2.3363E+02	2.0917E+02	3.0010E+02	8.1500E+02
	Clasificación	1	5	2	4	6	3
	Significación		+	+	+	+	+
F_{11}	Media	3.2932E+01	7.9240E+01	7.3961E+01	2.7578E+02	2.1665E+02	9.3600E+02
	Std	1.7540E+01	9.4859E+00	1.6453E+01	6.0425E+01	4.1872E+01	1.7000E+03
	Clasificación	1	3	2	5	4	6
	Significación		+	+	+	+	+

Continúa en la página siguiente

Tabla 4.4.4. *Continúa de la página anterior*

Función	Métrica	HE-DEPSO	DEPSO	SHADE	CoDE	DE	PSO
F_{12}	Media	6.0704E+03	7.4374E+04	1.7001E+06	5.9649E+08	2.5500E+09	1.1600E+10
	Std	5.2411E+03	7.3297E+04	5.7892E+05	1.8156E+08	8.9672E+08	1.5200E+10
	Clasificación	1	2	3	4	5	6
	Significación		+	+	+	+	+
F_{13}	Media	5.3737E+01	2.3200E+03	7.2862E+03	3.3010E+08	1.0201E+08	1.0300E+10
	Std	4.9535E+01	5.3868E+03	2.4057E+04	8.7976E+07	3.9539E+07	1.5000E+10
	Clasificación	1	2	3	5	4	6
	Significación		+	+	+	+	+
F_{14}	Media	5.1892E+01	9.7833E+01	7.5446E+01	1.5942E+04	4.0951E+03	2.2100E+05
	Std	1.3639E+01	2.2150E+01	1.0230E+01	4.5198E+03	8.1599E+02	4.9000E+05
	Clasificación	1	3	2	5	4	6
	Significación		+	+	+	+	+
F_{15}	Media	6.7166E+01	7.9018E+01	1.0487E+02	3.0409E+07	3.1028E+06	2.9100E+08
	Std	4.6957E+01	3.4973E+01	2.4704E+01	9.9730E+06	1.2198E+06	1.6200E+09
	Clasificación	1	2	3	5	4	6
	Significación		≈	≈	+	+	+
F_{16}	Media	2.7544E+02	1.4323E+03	8.5936E+02	1.6794E+03	2.3141E+03	1.7100E+03
	Std	1.4178E+02	1.6353E+02	1.1941E+02	1.7833E+02	1.8579E+02	6.0700E+02
	Clasificación	1	3	2	4	6	5
	Significación		+	+	+	+	+
F_{17}	Media	6.6918E+01	3.8170E+02	1.9226E+02	6.8169E+02	1.1646E+03	8.4800E+02
	Std	1.3274E+01	1.0402E+02	6.2540E+01	9.4653E+01	1.4710E+02	3.2100E+02
	Clasificación	1	3	2	4	6	5
	Significación		+	+	+	+	+
F_{18}	Media	6.1722E+01	2.4655E+02	2.3829E+05	3.4473E+05	7.9235E+05	1.9200E+06
	Std	2.7440E+01	2.4587E+02	1.4060E+05	1.1610E+05	1.9796E+05	4.8700E+06
	Clasificación	1	2	3	4	5	6
	Significación		+	+	+	+	+
F_{19}	Media	4.0473E+01	3.7316E+01	3.2834E+01	5.0014E+03	3.7873E+04	4.2500E+08
	Std	2.2587E+01	7.4498E+00	6.1811E+00	2.4562E+03	2.1473E+04	7.1600E+08
	Clasificación	3	2	1	4	5	6
	Significación		≈	≈	+	+	+
F_{20}	Media	8.7890E+01	2.6052E+02	3.3149E+02	4.8347E+02	1.0591E+03	6.0400E+02
	Std	3.2291E+01	9.0720E+01	7.0732E+01	7.4079E+01	1.1750E+02	2.3100E+02
	Clasificación	1	2	3	4	6	5
	Significación		+	+	+	+	+
F_{21}	Media	2.2047E+02	3.7820E+02	2.7893E+02	3.9662E+02	4.4351E+02	3.8400E+02
	Std	4.7309E+00	1.3317E+01	3.7140E+01	7.1300E+00	1.3511E+01	5.0800E+01
	Clasificación	1	3	2	5	6	4
	Significación		≈	≈	≈	+	≈
F_{22}	Media	1.0186E+02	2.9760E+03	2.2144E+02	6.5363E+03	8.0766E+03	4.5500E+03
	Std	2.0196E+00	3.4464E+03	2.9380E+01	5.3749E+02	2.9236E+02	2.0400E+03
	Clasificación	1	3	2	5	6	4
	Significación		+	≈	+	+	+
F_{23}	Media	3.7543E+02	5.4924E+02	4.4917E+02	5.5419E+02	5.8732E+02	7.2800E+02
	Std	8.3139E+00	1.3235E+01	9.7459E+00	1.2816E+01	1.9032E+01	7.9500E+01
	Clasificación	1	3	2	4	5	6
	Significación		+	+	+	+	+

Continúa en la página siguiente

Tabla 4.4.4. Continúa de la página anterior

Función	Métrica	HE-DEPSO	DEPSO	SHADE	CoDE	DE	PSO
F_{24}	Media	4.4893E+02	6.2854E+02	5.3810E+02	6.9034E+02	6.7462E+02	8.2800E+02
	Std	6.6585E+00	1.3484E+01	1.6953E+01	1.5747E+01	1.2401E+01	1.0000E+02
	Clasificación	1	3	2	5	4	6
	Significación		≈	≈	+	≈	+
F_{25}	Media	3.8051E+02	3.7837E+02	3.7882E+02	4.1044E+02	3.9396E+02	5.6400E+02
	Std	1.1308E+01	6.0195E-02	1.1887E+00	6.6423E+00	4.4183E+00	2.0000E+02
	Clasificación	3	1	2	5	4	6
	Significación		–	≈	+	≈	+
F_{26}	Media	1.3469E+03	2.6899E+03	1.4149E+03	2.6587E+03	3.0652E+03	4.9700E+03
	Std	9.6715E+01	2.2460E+02	5.7298E+02	9.0222E+01	1.2139E+02	9.0600E+02
	Clasificación	1	4	2	3	5	6
	Significación		+	≈	+	+	+
F_{27}	Media	5.0001E+02	5.0001E+02	5.0001E+02	5.0001E+02	5.0001E+02	7.1600E+02
	Std	1.5786E-04	1.2917E-04	9.3150E-05	5.9917E-05	8.1534E-05	1.0700E+02
	Clasificación	1	1	1	1	1	2
	Significación		≈	≈	≈	≈	≈
F_{28}	Media	4.9902E+02	4.9059E+02	4.9845E+02	5.0001E+02	5.0001E+02	1.2300E+03
	Std	3.0735E+00	4.2116E+00	1.3995E+00	6.6494E-05	6.6473E-05	8.4400E+02
	Clasificación	3	1	2	4	4	5
	Significación		–	≈	≈	≈	+
F_{29}	Media	3.8546E+02	9.3919E+02	4.6431E+02	1.6187E+03	2.0724E+03	1.9200E+03
	Std	4.5529E+01	2.1194E+02	4.2813E+01	1.8521E+02	1.8256E+02	6.1300E+02
	Clasificación	1	3	2	4	6	5
	Significación		+	+	+	+	+
F_{30}	Media	2.8659E+02	2.3933E+02	1.7439E+04	1.4650E+07	2.4426E+06	4.7800E+07
	Std	5.8034E+01	6.7166E+00	2.1431E+04	6.8820E+06	1.5971E+06	8.9800E+07
	Clasificación	2	1	3	5	4	6
	Significación		–	+	+	+	+
Clasificación promedio		1.31	2.44	2.44	4.37	4.75	5.13
Clasificación final		1	2	2	3	4	5
W/T/L		-/-/-	19/5/5	20/9/0	26/3/0	25/4/0	27/2/0

4.4.3.1. Resultados de optimización

De acuerdo a los resultados reportados en la Tabla 4.4.3, para $D = 10$, el algoritmo propuesto HE-DEPSO obtiene la solución óptima global para las funciones $F_1 - F_4$, F_6 y F_9 . En la función unimodal F_3 , DEPSO y SHADE consiguen la solución óptima global. Por otra parte, SHADE, CoDE y DE, obtienen la solución óptima global en la función multimodal simple F_9 . Para las funciones F_5 , $F_7 - F_9$, F_{11} , $F_{13} - F_{20}$, F_{23} y F_{30} el algoritmo HE-DEPSO logra el mejor resultado entre todos los algoritmos. DEPSO es el mejor en las funciones F_{12} , F_{21} , F_{24} y $F_{27} - F_{29}$. SHADE obtiene los mejores resultados en la función F_{10} , mientras que CoDE los obtiene en las funciones F_{22} , F_{25} y F_{26} . PSO no muestra resultados superiores en ninguna función. Los resultados de esta tabla indican que, para estas pruebas, el algoritmo propuesto HE-DEPSO obtiene la mejor clasificación de acuerdo al valor de la media entre todos los algoritmos. Por otro lado, los resultados de la prueba de suma de rangos de Wilcoxon muestran que HE-DEPSO es superior a DEPSO, SHADE, CoDE, DE y PSO en 17, 20, 24, 25, y 29 funciones de un total de 29 funciones de prueba respectivamente.

Para $D = 30$, los resultados estadísticos presentados en la Tabla 4.4.4, muestran que el algoritmo HE-DEPSO obtiene los mejores resultados en las funciones $F_1 - F_5$, $F_7 - F_8$, $F_{10} - F_{18}$, $F_{20} - F_{24}$, $F_{26} - F_{27}$ y F_{29} . Por otra parte, DEPSO, SHADE, CoDE y DE, logran los mejores resultados en la función F_{27} . DEPSO es el mejor en las funciones F_6 , F_9 , F_{25} , F_{28} y F_{30} , mientras que SHADE, demuestra se el mejor en la función F_{19} en comparación al resto de algoritmos. Con estos resultados, es posible identificar que HE-DEPSO obtiene la mejor clasificación entre todos los algoritmos para estas pruebas. De acuerdo con los resultados de la prueba de suma de rangos de Wilcoxon, entre las 29 funciones de prueba, HE-DEPSO tiene 19, 20, 26, 25, y 27 elementos que son mejores que DEPSO, SHADE, CoDE, DE y PSO, respectivamente.

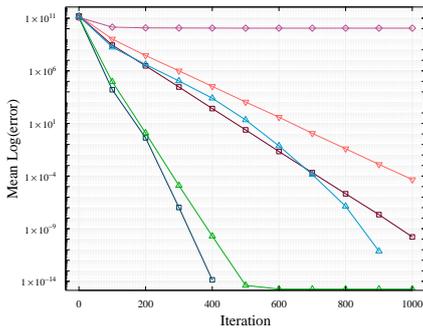
4.4.3.2. Propiedades de convergencia

Las propiedades de convergencia pueden ser resumidas en cuatro tipos los cuales se representan mediante las gráficas presentadas en las Figuras 4.4.1 y 4.4.2.

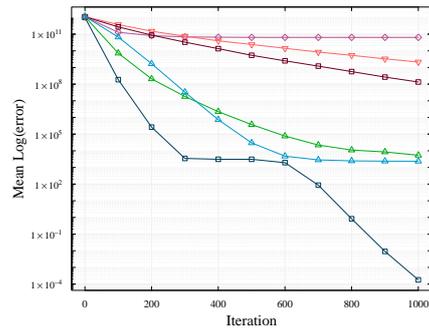
1. Las propiedades de convergencia de las funciones $F_1 - F_4$, F_6 y F_9 , pueden ser identificadas dentro de una misma clase, esto se puede observar con ayuda de las Figuras 4.4.1a a 4.4.1d. En este tipo, el algoritmo HE-DEPSO muestra una convergencia más rápida que el resto de algoritmos en $D = 10$, mientras que en $D = 30$, obtiene un mejor valor de media de error.
2. En las Figuras 4.4.1e a 4.4.1h, las curvas de convergencia de las funciones F_5 y F_8 son presentadas, las cuales son similares a las correspondientes para las funciones F_{10} , F_{16} , F_{17} y F_{20} . En estas funciones, todos los algoritmos se encuentran en diversos grados de estancamiento evolutivo o presentan una evolución lenta.
3. Las curvas de convergencia de las funciones $F_{12} - F_{15}$, $F_{18} - F_{19}$, F_{23} y $F_{29} - F_{30}$, son similares a las curvas de las funciones F_7 y F_{24} (ver Figuras 4.4.2a a 4.4.2d) obtenidas para el algoritmo HE-DEPSO. En este tipo, HE-DEPSO presenta al principio una rápida convergencia, y posteriormente, sigue evolucionando a la baja.
4. El proceso evolutivo presentado en las curvas de convergencia de las funciones F_{22} y F_{27} , sigue la misma tendencia que las curvas de convergencia de las funciones F_{25} y F_{28} como se puede ver en las Figuras 4.4.2e a 4.4.2h. Aquí, la mayoría de los algoritmos presentan estancamiento de forma acelerada, aunque algunos de ellos logran seguir evolucionando a la baja.

En el Apéndice B se muestra en la Figura B.1 todas las curvas de convergencia para las funciones de prueba CEC 2017 con $D = 10$, mientras que en la Figura B.2, se muestran para $D = 30$.

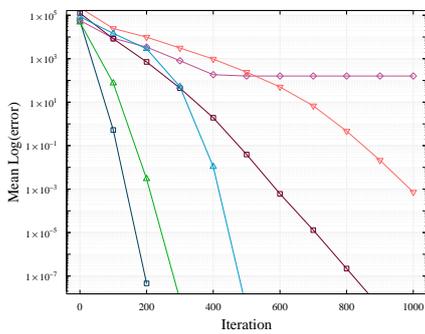
HE-DEPSO DEPSO PSO DE CoDE SHADE



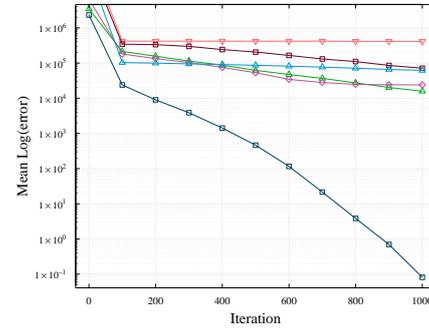
(a) F_1 ($D = 10$)



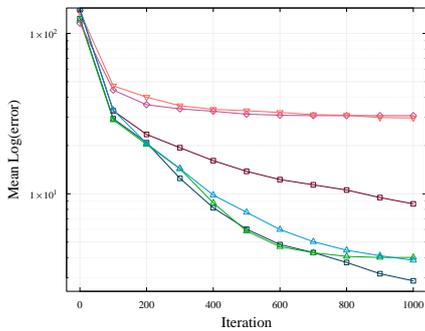
(b) F_1 ($D = 30$)



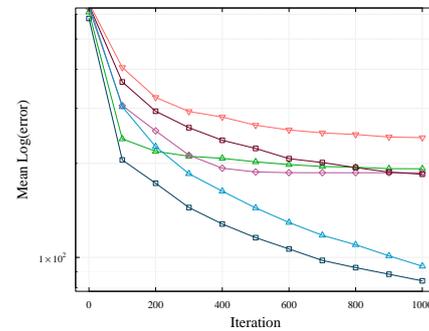
(c) F_3 ($D = 10$)



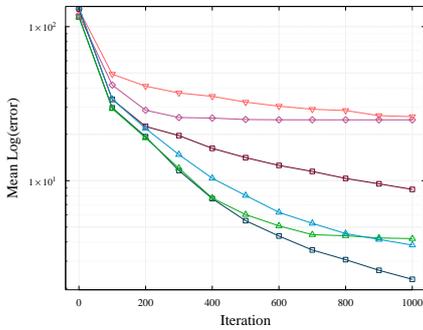
(d) F_3 ($D = 30$)



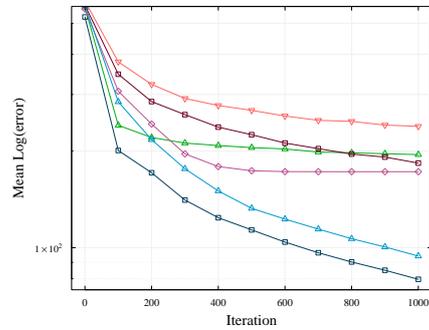
(e) F_5 ($D = 10$)



(f) F_5 ($D = 30$)



(g) F_8 ($D = 10$)



(h) F_8 ($D = 30$)

Figura 4.4.1: Curvas de convergencia de la medida de error de solución para las funciones F_1 , F_3 , F_5 y F_8 , con $D = 10, 30$. El eje horizontal y vertical representan la iteraciones y los valores de media de error para las 31 repeticiones independientes.

HE-DEPSO DEPSO PSO DE CoDE SHADE

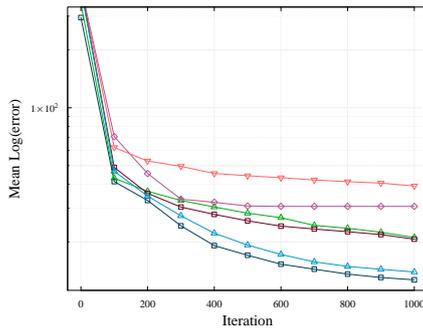
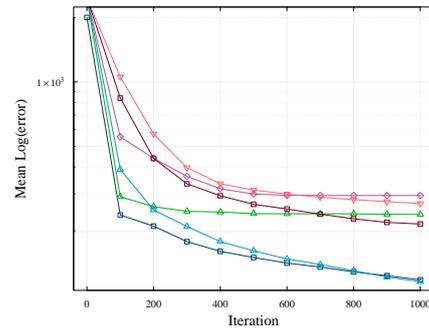
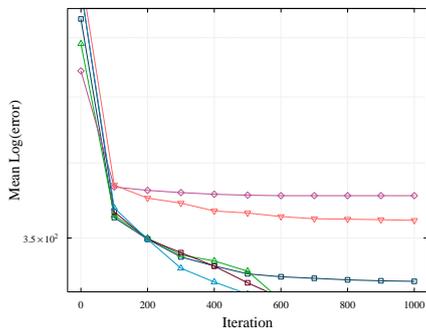
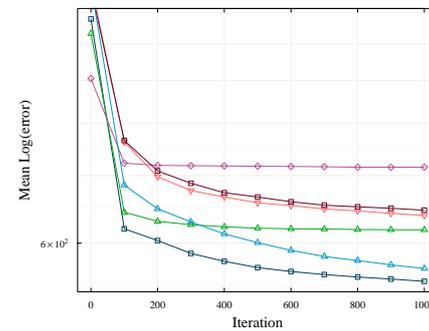
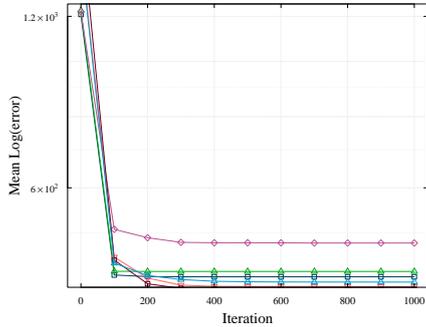
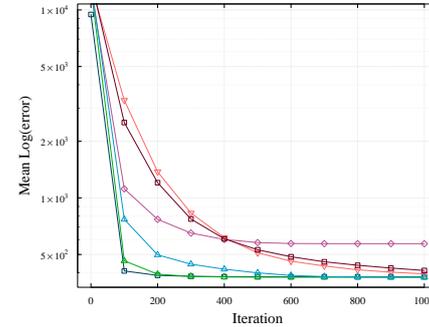
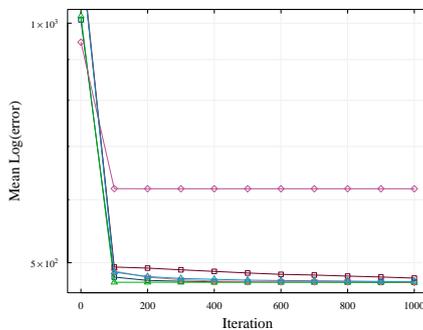
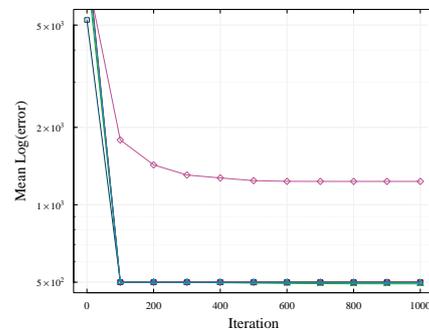
(a) F_7 ($D = 10$)(b) F_7 ($D = 30$)(c) F_{24} ($D = 10$)(d) F_{24} ($D = 30$)(e) F_{25} ($D = 10$)(f) F_{25} ($D = 30$)(g) F_{28} ($D = 10$)(h) F_{28} ($D = 30$)

Figura 4.4.2: Curvas de convergencia de la medida de error de solución para las funciones F_1 , F_3 , F_5 y F_8 , con $D = 10, 30$. El eje horizontal y vertical representan la iteraciones y los valores de media de error para las 31 repeticiones independientes.

4.4.4. Discusión

Los experimentos realizados anteriormente, prueban la superioridad del algoritmo propuesto HE-DEPSO al resolver diferentes problemas de optimización. Las razones por las cuales el algoritmo HE-DEPSO obtiene dicho rendimiento superior pueden ser resumidas de la siguiente manera.

1. El operador de mutación híbrido adaptativo implementado en HE-DEPSO se encuentra basado en la estrategia de mutación auto-adaptativa del algoritmo DEPSO, la cual ha demostrado ser útil al abordar diversos problemas complejos de optimización. Por otro lado, el trabajo colaborativo entre la estrategia de mutación “DE/current-to-EHE/1” con información histórica de los individuos elite y la estrategia del algoritmo PSO, genera un buen balance entre exploración y explotación en diferentes etapas del proceso evolutivo.
2. El control auto-adaptativo de los parámetros F y Cr adoptado del algoritmo SHADE, permite mitigar la sensibilidad de los parámetros. De esta forma, la probabilidad de cruce y el factor de escala son ajustados de forma dinámica durante el proceso evolutivo a nivel de cada individuo, lo cual hace que el algoritmo propuesto sea adecuado para una mayor variedad de problemas de optimización.

Capítulo 5

Análisis numérico del modelo de texturas de cuatro ceros con HE-DEPSO

En este capítulo, se aborda el problema principal del trabajo de tesis, que consiste en analizar numéricamente la viabilidad de utilizar una matriz de textura con cuatro ceros para representar las matrices de masa de los quarks. Se obtienen expresiones analíticas para la matriz V_{CKM} predichas por el modelo con cuatro ceros de textura y se evalúa su compatibilidad y viabilidad mediante el método de ajuste de χ^2 utilizando los datos experimentales más recientes y las masas de los quarks. Se implementa el algoritmo HE-DEPSO para minimizar la función χ^2 y determinar el espacio de parámetros permitido del modelo. Finalmente, se utiliza este espacio para analizar el poder predictivo del modelo, determinar su viabilidad según los últimos datos experimentales y discutir sus implicaciones dentro del formalismo de texturas.

5.1. Modelo de texturas de cuatro ceros

El presente trabajo está enfocado en analizar el modelo de texturas de Fritzsche de cuatro ceros. En esta sección, se presenta una descripción detallada de dicho modelo, incluyendo la derivación de las expresiones analíticas de los elementos de la matriz de mezcla de quarks V_{CKM} predichas por este modelo. Estas expresiones son utilizadas para evaluar la validez y capacidad predictiva del modelo de cuatro ceros al considerar los datos experimentales más recientes de la matriz V_{CKM} y las masas de los quarks.

De acuerdo a lo mencionado en la Sección 2.4, en el formalismo de texturas se proponen estructuras con ceros en algunas entradas de la matriz de masa para encontrar analíticamente las matrices que la diagonalizan y calcular la matriz de

mezcla V_{CKM} , validando así la estructura de texturas elegida. Sin pérdida de generalidad, se consideran a las matrices de masa M_u y M_d hermíticas, de tal manera que la estructura más general tiene la forma:

$$M_q = \begin{pmatrix} E_q & D_q & F_q \\ D_q^* & C_q & B_q \\ F_q^* & B_q^* & A_q \end{pmatrix} \quad q = u, d. \quad (5.1.1)$$

El modelo de texturas de Fritzsche de cuatro ceros se caracteriza por tener cuatro elementos nulos en la matriz de masa de los quarks, lo que permite obtener relaciones entre los elementos de la matriz V_{CKM} y las masas de los quarks. Esta textura se forma a partir de la matriz de la Ecuación (5.1.1), tomando las entradas $(1, 1)$, $(1, 3)$ y $(3, 1)$ iguales a cero, dando lugar a la siguiente estructura matricial:

$$M_q = \begin{pmatrix} 0 & D_q & 0 \\ D_q^* & C_q & B_q \\ 0 & B_q^* & A_q \end{pmatrix} \quad q = u, d. \quad (5.1.2)$$

En la Ecuación (5.1.2), los elementos A_q y C_q son reales, mientras que B_q y D_q son complejos y se expresan en forma polar como $Z_q = |Z_q|e^{i\phi_{Z_q}}$, donde $|Z_q|$ es la magnitud y $i\phi_{Z_q}$ es su fase angular ($Z = B, D$). Por lo tanto, la Ecuación (5.1.2) puede reescribirse en forma polar de la siguiente manera:

$$M_q = \begin{pmatrix} 0 & |D_q|e^{i\phi_{D_q}} & 0 \\ |D_q|e^{-i\phi_{D_q}} & C_q & |B_q|e^{i\phi_{B_q}} \\ 0 & |B_q|e^{-i\phi_{B_q}} & A_q \end{pmatrix} \quad (5.1.3)$$

Separando las fases en la matriz de la Ecuación (5.1.3), mediante el producto de una matriz ortogonal \bar{M}_q y una matriz diagonal unitaria de fase P_q ($M_q = P_q^\dagger \bar{M}_q P_q$), se tiene:

$$M_q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{-i\phi_{D_q}} & 0 \\ 0 & 0 & e^{-i(\phi_{D_q} + \phi_{B_q})} \end{pmatrix} \begin{pmatrix} 0 & |D_q| & 0 \\ |D_q| & C_q & |B_q| \\ 0 & |B_q| & A_q \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{i\phi_{D_q}} & 0 \\ 0 & 0 & e^{i(\phi_{D_q} + \phi_{B_q})} \end{pmatrix} \quad (5.1.4)$$

donde la matriz ortogonal \bar{M}_q es:

$$\bar{M}_q = \begin{pmatrix} 0 & |D_q| & 0 \\ |D_q| & C_q & |B_q| \\ 0 & |B_q| & A_q \end{pmatrix} \quad (5.1.5)$$

La matriz ortogonal \bar{M}_q se puede diagonalizar por medio de una transformación ortogonal:

$$O_q^T \bar{M}_q O_q = \text{diag}\{\lambda_1^q, \lambda_2^q, \lambda_3^q\}, \quad (5.1.6)$$

donde λ_i^q denota cada uno de los tres eigenvalores de \bar{M}_q , y O_q representa una matriz de transformación.

Es conveniente expresar la matriz de transformación O_q como una matriz de rotación, de forma que se puede hacer uso de los invariantes ante transformaciones de similaridad en conjunto con los valores propios.

$$T_r(\bar{M}_q) = \sum_i \lambda_i^q \quad (5.1.7)$$

$$\text{Det}(\bar{M}_q) = \prod_i \lambda_i^q \quad (5.1.8)$$

$$T_r(\bar{M}_q^2) = \sum_i (\lambda_i^q)^2 \quad (5.1.9)$$

Por medio de las Ecuaciones (5.1.7) a (5.1.9) se puede obtener una relación directa entre las masas de los quarks y sus respectivos acoplamientos de Yukawa. Los valores de los elementos de masa $|B_q|^2$, $|D_q|^2$ y C_q en términos del parámetro libre A_q están dados por:

$$C_q = \lambda_1^q + \lambda_2^q + \lambda_3^q - A_q \quad (5.1.10)$$

$$|D_q|^2 = \frac{-\lambda_1^q \lambda_2^q \lambda_3^q}{A_q} \quad (5.1.11)$$

$$|B_q|^2 = \frac{(A_q - \lambda_1^q)(A_q - \lambda_2^q)(\lambda_3^q - A_q)}{A_q} \quad (5.1.12)$$

A partir de la fuerte jerarquía $|\lambda_1^q| \ll |\lambda_2^q| \ll |\lambda_3^q|$, sin pérdida de la generalidad, se puede tomar $\lambda_3^q > 0$ y $A_q > 0$, lo cual requiere que $\lambda_1^q \lambda_2^q < 0$ ya que la Ecuación (5.1.13) se debe satisfacer.

$$\text{Det}(\bar{M}_q) = -|D_q|^2 A_q = \lambda_1^q \lambda_2^q \lambda_3^q. \quad (5.1.13)$$

Bajo esta condición las relaciones entre λ_i^q con las masas físicas de los quarks son:

$$(\lambda_1^u, \lambda_2^u, \lambda_3^u) = (-\eta_u m_u, \eta_u m_c, m_t) \quad (5.1.14)$$

$$(\lambda_1^d, \lambda_2^d, \lambda_3^d) = (-\eta_d m_d, \eta_d m_s, m_b) \quad (5.1.15)$$

donde m_u, m_c y m_t son las masas de los quarks u, c y t, respectivamente, y m_d, m_s y m_b son las masas de los quarks d, s y b. En este trabajo se considera la misma estructura de texturas de cuatro ceros para la matriz de masa de los quarks u y la matriz de masa de los quarks tipo d (una estructura de masas paralela).

A partir de la ecuación de vectores propios:

$$\bar{M}_q \vec{v}_i = \lambda_i^q \vec{v}_i, \quad i = 1, 2, 3 \quad (5.1.16)$$

y haciendo uso de las Ecuaciones (5.1.10) a (5.1.12), en función de las masas, se puede obtener la matriz de transformación O_q al calcular de forma explícita cada uno de los vectores propios \vec{v}_i , es decir:

$$\begin{pmatrix} -\lambda_i^q & |D_q| & 0 \\ |D_q| & c_q - \lambda_i^q & |B_q| \\ 0 & |B_q| & A_q - \lambda_i^q \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (5.1.17)$$

de donde se obtienen las siguientes tres ecuaciones:

$$-\lambda_i^q x_i + |D_q| y_i = 0 \quad (5.1.18)$$

$$|D_q|x_i + (c_q - \lambda_i^q)y_i + |B_q|z_i = 0 \quad (5.1.19)$$

$$|B_q|y_i + (A_q - \lambda_i^q)z_i = 0 \quad (5.1.20)$$

de las cuales solo dos son linealmente independientes, por lo tanto, de la Ecuación (5.1.18) se tiene:

$$y_i = \frac{\lambda_i^q x_i}{|D_q|} \quad (5.1.21)$$

Utilizando la Ecuación (5.1.21) en la Ecuación (5.1.20):

$$z_i = \frac{-|B_q|\lambda_i^q x_i}{|D_q|(A_q - \lambda_i^q)} \quad (5.1.22)$$

Con lo cual el eigenvector \vec{v}_i toma la forma:

$$\vec{v}_i = \begin{pmatrix} 1 \\ \frac{\lambda_i^q}{|D_q|} \\ \frac{-|B_q|\lambda_i^q}{|D_q|(A_q - \lambda_i^q)} \end{pmatrix} x_i \quad (5.1.23)$$

A partir de las Ecuaciones (5.1.20) a (5.1.22) se tiene que $x_i = 1$, por lo tanto:

$$\vec{v}_i = \begin{pmatrix} 1 \\ \frac{\lambda_i^q}{|D_q|} \\ \frac{-|B_q|\lambda_i^q}{|D_q|(A_q - \lambda_i^q)} \end{pmatrix} \quad (5.1.24)$$

Normalizando:

$$\vec{v}_i = \left(1 + \frac{(\lambda_i^q)^2}{|D_q|^2} + \frac{|B_q|^2(\lambda_i^q)^2}{|D_q|^2(A_q - \lambda_i^q)^2}\right)^{-1/2} \begin{pmatrix} 1 \\ \frac{\lambda_i^q}{|D_q|} \\ \frac{-|B_q|\lambda_i^q}{|D_q|(A_q - \lambda_i^q)} \end{pmatrix} \quad (5.1.25)$$

Expresando a cada uno de los vectores propios normalizados \hat{v}_i , que componen a la matriz de transformación O_q en términos de los valores propios λ_i^q , por medio de la Ecuación (5.1.25), se tiene:

$$\hat{v}_1 = \begin{pmatrix} \sqrt{\frac{\lambda_2^q \lambda_3^q (A_q - \lambda_1^q)}{A_q (\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_1^q)}} \\ -\eta_q \sqrt{\frac{\lambda_1^q (\lambda_1^q - A_q)}{(\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_1^q)}} \\ \eta_q \sqrt{\frac{\lambda_1^q (A_q - \lambda_2^q) (A_q - \lambda_3^q)}{A_q (\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_1^q)}} \end{pmatrix} \quad (5.1.26)$$

$$\hat{v}_2 = \begin{pmatrix} \eta_q \sqrt{\frac{\lambda_1^q \lambda_3^q (\lambda_2^q - A_q)}{A_q (\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} \\ \sqrt{\frac{\lambda_2^q (A_q - \lambda_2^q)}{(\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} \\ -\sqrt{\frac{\lambda_2^q (A_q - \lambda_1^q) (\lambda_3^q - A_q)}{A_q (\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} \end{pmatrix} \quad (5.1.27)$$

$$\hat{v}_3 = \begin{pmatrix} \sqrt{\frac{\lambda_1^q \lambda_2^q (A_q - \lambda_3^q)}{A_q (\lambda_3^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} \\ \sqrt{\frac{\lambda_3^q (\lambda_3^q - A_q)}{(\lambda_3^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} \\ \sqrt{\frac{\lambda_3^q (A_q - \lambda_1^q) (A_q - \lambda_2^q)}{A_q (\lambda_3^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} \end{pmatrix} \quad (5.1.28)$$

Es entonces que, con los vectores propios \hat{v}_1 , \hat{v}_2 y \hat{v}_3 se puede construir la matriz de transformación O_q :

$$O_q = \begin{pmatrix} \hat{v}_1(1) & \hat{v}_2(1) & \hat{v}_3(1) \\ \hat{v}_1(2) & \hat{v}_2(2) & \hat{v}_3(2) \\ \hat{v}_1(3) & \hat{v}_2(3) & \hat{v}_3(3) \end{pmatrix} \quad (5.1.29)$$

$$O_q = \begin{pmatrix} \sqrt{\frac{\lambda_2^q \lambda_3^q (A_q - \lambda_1^q)}{A_q (\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_1^q)}} & \eta_q \sqrt{\frac{\lambda_1^q \lambda_3^q (\lambda_2^q - A_q)}{A_q (\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} & \sqrt{\frac{\lambda_1^q \lambda_2^q (A_q - \lambda_3^q)}{A_q (\lambda_3^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} \\ -\eta_q \sqrt{\frac{\lambda_1^q (\lambda_1^q - A_q)}{(\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_1^q)}} & \sqrt{\frac{\lambda_2^q (A_q - \lambda_2^q)}{(\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} & \sqrt{\frac{\lambda_3^q (\lambda_3^q - A_q)}{(\lambda_3^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} \\ \eta_q \sqrt{\frac{\lambda_1^q (A_q - \lambda_2^q) (A_q - \lambda_3^q)}{A_q (\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_1^q)}} & -\sqrt{\frac{\lambda_2^q (A_q - \lambda_1^q) (\lambda_3^q - A_q)}{A_q (\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} & \sqrt{\frac{\lambda_3^q (A_q - \lambda_1^q) (A_q - \lambda_2^q)}{A_q (\lambda_3^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} \end{pmatrix} \quad (5.1.30)$$

donde $\eta = +1$ y $\eta = -1$ corresponden respectivamente a $\lambda_2^q > 0$ y $\lambda_2^q < 0$, esto conforme a las posibilidades $(\lambda_1^q, \lambda_2^q) = (-\lambda_1^q, +\lambda_2^q)$ o $(+\lambda_1^q, -\lambda_2^q)$.

A partir de lo anterior, se puede construir la matriz de mezcla V_{CKM} predicha por el modelo de cuatro texturas:

$$V_{CKM}^{th} = O_u^\dagger P_u P_d^\dagger O_d \quad (5.1.31)$$

donde:

$$\begin{aligned} P_u P_d^\dagger &= \text{diag}\{1, e^{i(\phi_{D_u} - \phi_{D_d})}, e^{i[(\phi_{D_u} - \phi_{D_d}) + (\phi_{B_u} - \phi_{B_d})]}\} \\ &= \text{diag}\{1, e^{i\phi_1}, e^{i(\phi_1 + \phi_2)}\} \end{aligned} \quad (5.1.32)$$

Con las fases ϕ_1 y ϕ_2 definidas como:

$$\phi_1 = \phi_{D_u} - \phi_{D_d} = \text{arg}(D_u) - \text{arg}(D_d) \quad (5.1.33)$$

$$\phi_2 = (\phi_{B_u} - \phi_{B_d}) = \text{arg}(B_u) - \text{arg}(B_d) \quad (5.1.34)$$

De manera que la matriz V_{CKM} queda parametrizada con dos fases las cuales se consideran medidas en radianes con argumento principal $[0, 2\pi]$. Por lo tanto, los nueve elementos de la matriz V_{CKM} predichos por el modelo de cuatro texturas son:

$$(V_{CKM}^{th})_{ij} = (O_u)_{1i}(O_d)_{1j} + (O_u)_{2i}(O_d)_{2j}e^{i\phi_1^q} + (O_u)_{3i}(O_d)_{3j}e^{i(\phi_1^q + \phi_2^q)} \quad (5.1.35)$$

donde los índices i y j corresponden respectivamente a los índices (u, c, t) y (d, s, b) .

La magnitud de los elementos de la matriz de mezcla vienen dado por:

$$\begin{aligned} |(V_{CKM}^{th})_{ij}| &= \left\{ [(O_u)_{1i}(O_d)_{1j}]^2 + [(O_u)_{2i}(O_d)_{2j}]^2 + [(O_u)_{3i}(O_d)_{3j}]^2 \right. \\ &\quad + 2[(O_u)_{1i}(O_d)_{1j}][(O_u)_{2i}(O_d)_{2j}] \text{Cos}(\phi_1) \\ &\quad + 2[(O_u)_{1i}(O_d)_{1j}][(O_u)_{3i}(O_d)_{3j}] \text{Cos}(\phi_1 + \phi_2) \\ &\quad \left. + 2[(O_u)_{2i}(O_d)_{2j}][(O_u)_{3i}(O_d)_{3j}] \text{Cos}(\phi_2) \right\}^{1/2}. \end{aligned} \quad (5.1.36)$$

Hasta este punto, se han calculado las expresiones analíticas para los elementos de la matriz de mezcla de quarks utilizando el formalismo de texturas de cuatro ceros. Estas expresiones se pueden comparar con los datos experimentales para verificar la consistencia del modelo con los datos experimentales.

5.2. Análisis numérico del modelo de texturas de cuatro ceros

La Ecuación (5.1.35) proporciona las expresiones teóricas de los elementos de la matriz V_{CKM}^{th} del modelo de texturas de cuatro ceros, con una dependencia explícita de los parámetros A_u, A_d, ϕ_1 y ϕ_2 . Para comprobar la viabilidad de este modelo, se buscarán los valores permitidos para estos parámetros que concuerden con los datos experimentales recientes, considerando sus márgenes de error. Esto se realizará mediante un procedimiento de minimización de la función estimadora χ^2 . Además, se evaluará el poder predictivo del modelo al reproducir los valores experimentales de la matriz V_{CKM} . Los resultados determinarán si el modelo de texturas de cuatro ceros es viable para describir la estructura de la matriz de masa de los quarks. Finalmente, se discutirán las implicaciones y posibles extensiones del modelo en el contexto de la física de partículas.

5.2.1. Análisis de χ^2

Para probar la compatibilidad del modelo de texturas de cuatro ceros con los datos experimentales más recientes, se utiliza el método de ajuste de χ^2 . Concretamente, se define la función χ^2 la cual tiene la forma usual (ver Apéndice C):

$$\chi^2(x) \equiv \sum_{i=1}^n \left(\frac{P_i(x) - \bar{O}_i}{\sigma_i} \right)^2, \quad (5.2.1)$$

donde el vector x contiene los parámetros del modelo, $P_i(x)$ es la predicción del modelo para la observable O_i y \bar{O}_i el valor central para O_i . σ_i representa el error de O_i .

Con este acercamiento, el espacio de parámetros de las matrices M_u y M_d puede restringirse considerando los valores más recientes de los elementos de la matriz V_{CKM} (ver Ecuación (2.3.37)), junto con los valores actualizados de las masas de los quarks a escala M_Z (ver Ecuación (2.3.38)).

En este análisis, se utilizaron como entradas los elementos independientes: $|V_{us}|$,

$|V_{ub}|$, $|V_{cb}|$ y el invariante de Jarlskog (J) (ver Ecuación (2.3.36)), dado que sus magnitudes han sido establecidas con notable precisión. En la Ecuación (5.2.2) se indica el valor del invariante de Jarlskog [87].

$$J = (3.00^{+0.15}_{-0.09}) \times 10^{-5} \quad (5.2.2)$$

Con lo establecido anteriormente, la función χ^2 definida para el modelo de texturas de cuatro ceros toma la siguiente estructura:

$$\begin{aligned} \chi^2(x) = & \frac{(|(V_{CKM})_{us}^{th}(x)| - |V_{us}|)^2}{\sigma_{V_{us}}^2} + \frac{(|(V_{CKM})_{ub}^{th}(x)| - |V_{ub}|)^2}{\sigma_{V_{ub}}^2} \\ & + \frac{(|(V_{CKM})_{cb}^{th}(x)| - |V_{cb}|)^2}{\sigma_{V_{cb}}^2} + \frac{(J^{th}(x) - J)^2}{\sigma_J^2} \end{aligned} \quad (5.2.3)$$

donde $x = (A_u, A_d, \phi_1, \phi_2)$, el súper índice “*th*” denota que el término proviene de la parte analítica (ver Ecuación (5.1.36)) y la correspondiente sin índices denota la parte experimental con incertidumbre $\sigma_{V_{ij}}^2$.

La utilidad principal del análisis de χ^2 estándar radica en determinar si el modelo de texturas de Fritzsche con cuatro ceros puede ser descartado a partir de los datos experimentales. Para evaluar la compatibilidad entre este modelo y los datos, en este trabajo se emplea un criterio específico: se considera que el modelo de texturas es compatible con los datos experimentales si el valor de Chi-cuadrado reducido (χ_{red}^2) es inferior a uno, lo que indica un buen ajuste [118]:

$$0 < \chi_{red}^2(x) < 1, \quad (5.2.4)$$

donde:

$$\chi_{red}^2(x) = \frac{\chi^2(x)}{4}. \quad (5.2.5)$$

Dado el criterio anterior, en lo que sigue se procederá a buscar, con ayuda del algoritmo HE-DEPSO, regiones de los parámetros A_u , A_d , ϕ_1 y ϕ_2 donde el valor de χ^2 reducido sea menor que uno, así como $\chi^2 < 1 \times 10^{-1}$ y $\chi^2 < 1 \times 10^{-2}$.

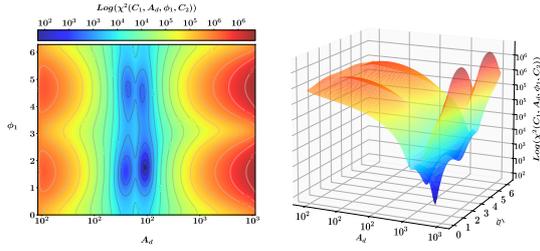
Esto nos permitirá identificar los valores de los parámetros que mejor se ajustan a los datos experimentales disponibles. Buscar estas regiones es fundamental para determinar la validez y el poder predictivo del modelo, fortaleciendo la evidencia a su favor si se encuentran regiones significativas que cumplan con este criterio.

Es crucial notar que la función χ^2 (ver Ecuación (5.2.3)), aunque aparentemente simple con pocos parámetros libres, la dificultad de encontrar el rango numérico de los mismos que cumplan la condición dada en la Ecuación (5.2.4) proviene de la engorrosa composición de funciones a evaluar que la constituyen. Para tener una noción del relieve topográfico de esta función, en la Figura 5.2.1 se muestran distintas proyecciones de la función χ^2 en diferentes planos. En la Figura 5.2.1c se muestra la proyección en el plano A_u y A_d (con ϕ_1 y ϕ_2 fijos). Asimismo, en Figura 5.2.1e se proyecta en A_u y ϕ_2 (con A_d y ϕ_1 fijos). Por último, en Figura 5.2.1f se visualiza la proyección en el plano ϕ_1 y ϕ_2 (con A_u y A_d fijos). El código de colores utilizado indica que las zonas rojas son elevadas y las azules son más bajas.

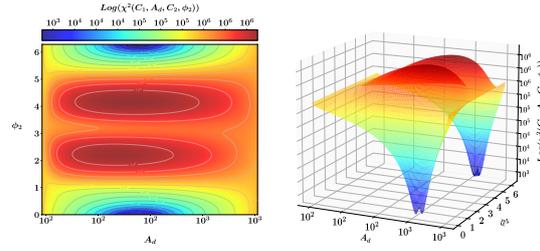
5.2.2. Optimización de χ^2 con HE-DEPSO

En la sección anterior se describió la idea del análisis de χ^2 dentro del cual se involucra la minimización de la función χ^2 , una función real positiva que depende de n parámetros. Con los cual, para utilizar el método de χ^2 y determinar las regiones de los parámetros A_u , A_d , ϕ_1 y ϕ_2 donde se cumpla el criterio establecido (ver Ecuación (5.2.4)), es esencial emplear un algoritmo especializado en la optimización de funciones con un solo objetivo y valores reales, sujeto a restricciones de frontera. En el campo de la FAE, las técnicas de búsqueda exhaustiva son comunes y efectivas en ciertos casos y condiciones, pero pueden ser poco precisas [119, 120]. Por otro lado, la implementación de métodos de optimización tradicionales, como los mencionados en la Subsección 2.1.2.1, puede enfrentar dificultades significativas debido varios factores:

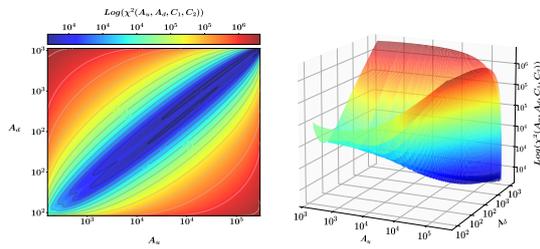
- La dependencia no lineal entre las cantidades (elementos de la matriz O_q).
- Debido a la naturaleza de las expresiones matemáticas involucradas, la función a optimizar χ^2 generalmente no será diferenciable.



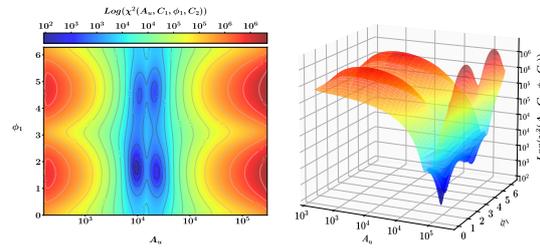
(a) Dependencia de la función χ^2 respecto a las variables A_d y ϕ_1 (gráfica de la derecha) y curvas de nivel (gráfica izquierda) correspondientes. Se observa una periodicidad de los máximos de la función χ^2 respecto a la variable ϕ_2 .



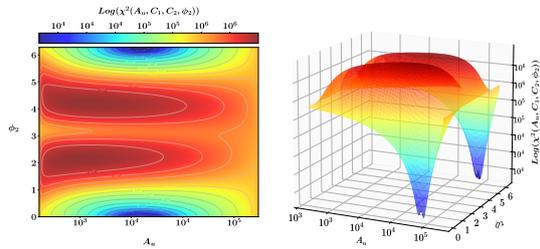
(b) Dependencia de la función χ^2 respecto a las variables A_d y ϕ_2 (gráfica de la derecha) y curvas de nivel (gráfica izquierda) correspondientes. Se observa una periodicidad de los máximos y de los mínimos de la función χ^2 respecto a la variable ϕ_2 .



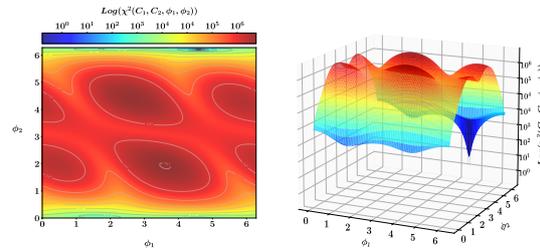
(c) Dependencia de la función χ^2 respecto a las variables A_u y A_d (gráfica de la derecha). De las curvas de nivel (gráfica izquierda) se observa que la región que minimiza a χ^2 se encuentra alrededor de una línea recta a 45° .



(d) Dependencia de la función χ^2 respecto a las variables A_u y ϕ_1 (gráfica de la derecha) y curvas de nivel (gráfica izquierda) correspondientes. Se observa una periodicidad de los máximos de la función χ^2 respecto a la variable ϕ_1 .



(e) Dependencia de la función χ^2 respecto a las variables A_u y ϕ_2 (gráfica de la derecha) y curvas de nivel (gráfica izquierda) correspondientes. Se observa una periodicidad de los máximos y de los mínimos de la función χ^2 respecto a la variable ϕ_1 .



(f) Dependencia de la función χ^2 respecto a las variables ϕ_1 y ϕ_2 (gráfica de la derecha) y curvas de nivel (gráfica izquierda) correspondientes.

Figura 5.2.1: Proyecciones de la función χ^2 sobre las variables A_u, A_d, ϕ_1, ϕ_2 .

- El entorno que describe la función a optimizar χ^2 (ver Figura 5.2.1) presenta múltiples mínimos locales.

lo que generalmente genera problemas de convergencia y demuestra la desventaja de utilizar métodos de optimización tradicionales en este contexto. En vista

de esto, en el presente trabajo se propone abordar el problema de optimizar la función χ^2 mediante un algoritmo más apropiado, capaz de explorar mejor el espacio de búsqueda y obtener soluciones más precisas. Dado el buen rendimiento del algoritmo HE-DEPSO demostrado en la Subsección 4.4.3, se implementará para minimizar la función χ^2 y encontrar las regiones de los parámetros A_u , A_d , ϕ_1 y ϕ_2 que cumplan con el criterio de la Ecuación (5.2.4).

Antes de aplicar el algoritmo HE-DEPSO para buscar las regiones de los parámetros A_u , A_d , ϕ_1 y ϕ_2 , se realizará un estudio previo del comportamiento del algoritmo al optimizar la función χ^2 . Este análisis se enfocará en las características de optimización, convergencia y estabilidad del algoritmo, lo que proporcionará una comprensión más profunda de las propiedades del algoritmo propuesto al ser aplicado a otro tipo de problemas.

Al plantear un problema de optimización con un solo objetivo y valores reales, sujeto a restricciones de frontera, como lo representa el problema de optimización de la función χ^2 , es muy importante establecer el espacio de búsqueda dentro del cual se llevará a cabo el proceso de optimización. Además, al hacer uso de un ABI es importante definir la forma en que las posibles soluciones serán representadas y manipuladas por el algoritmo.

En el problema de optimización de la función $\chi^2 : \mathbb{R}^4 \rightarrow \mathbb{R}$, la función objetivo está dada por la Ecuación (5.2.3), mientras que su espacio de búsqueda está condicionado por las posibles combinaciones de signos de los parámetros η_u y η_d (ver Ecuación (5.1.30) y Tabla 5.2.1), los cuales definen los diferentes casos de estudio a considerar para los parámetros A_u , A_d , ϕ_1 y ϕ_2 de la función χ^2 . De acuerdo a lo establecido dentro de la Sección 5.1, las fases ϕ_1 y ϕ_2 toman valores dentro del rango $[0, 2\pi]$. Los rangos de los parámetros A_u y A_d varían según los signos de η_u y η_d . En el Apéndice D se establecen los rangos respectivos de A_u y A_d para cada combinación de signos de η_u y η_d . La Tabla 5.2.2 resume los rangos de búsqueda de los parámetros A_u , A_d , ϕ_1 y ϕ_2 para cada combinación de η_u y η_d .

La representación de las posibles soluciones con las que trabajará el algoritmo HE-DEPSO tienen la siguiente forma:

Tabla 5.2.1: Diferentes casos de estudio.

Caso de estudio	η_u	η_d
I	+1	+1
II	+1	-1
III	-1	+1
IV	-1	-1

Tabla 5.2.2: Espacios de búsqueda conforme a las posibles combinaciones de η_u y η_d .

Caso de estudio	Espacio de búsqueda			
	A_u	A_d	ϕ_1	ϕ_2
I	$[m_c, m_t]$	$[m_s, m_b]$	$[0, 2\pi]$	$[0, 2\pi]$
II	$[m_c, m_t]$	$[m_d, m_b]$	$[0, 2\pi]$	$[0, 2\pi]$
III	$[m_u, m_t]$	$[m_s, m_b]$	$[0, 2\pi]$	$[0, 2\pi]$
IV	$[m_u, m_t]$	$[m_d, m_b]$	$[0, 2\pi]$	$[0, 2\pi]$

$$x_i = (A_u, A_d, \phi_1, \phi_2). \quad (5.2.6)$$

Dicho en otras palabras, el i -ésimo individuo de la población en el algoritmo HE-DEPSO estará representado por medio de un vector x_i de cuatro dimensiones, cuyas componentes $(A_u, A_d, \phi_1, \phi_2)$ viven respectivamente en los rangos descritos en la Tabla 5.2.2.

A través de un experimento se evaluará la eficacia del algoritmo HE-DEPSO en encontrar el mínimo global de la función χ^2 para analizar sus capacidades de optimización. Se utilizará el espacio de búsqueda del caso de estudio I de la Tabla 5.2.2. Se comparará el rendimiento del HE-DEPSO con los algoritmos PSO y DE, junto con variantes avanzadas del algoritmo DE, denominadas: CoDE [106], SHADE [108] y DEPSO [19]. Para asegurar una comparación justa, se fijaron los parámetros comunes de todos los algoritmos: $t_{max} = 1000$, $NP = 100$, y se realizaron 31 repeticiones independientes. Los demás parámetros de cada algoritmo se ajustaron según lo indicado en la Tabla 4.4.2.

Para evaluar el rendimiento del algoritmo HE-DEPSO propuesto, se compararon y analizaron sus resultados de optimización con los algoritmos DE, PSO y tres

variantes avanzadas de DE aplicados a la función χ^2 . Siguiendo un protocolo similar al presentado en la Subsección 4.4.3, la Tabla 5.2.3 reporta los resultados estadísticos de cada algoritmo.

Tabla 5.2.3: Comparación de resultados entre HE-DEPSO, DE, PSO y tres variantes avanzadas del algoritmo DE, sobre la función χ^2 .

Función	Métrica	HE-DEPSO	DEPSO	SHADE	CoDE	DE	PSO
χ^2	Media	1.7267E-28	3.4971E-28	7.2296E-03	2.3720E-05	6.1610E-13	7.4247E+00
	Std	1.4496E-28	5.6400E-28	8.7102E-03	7.7748E-05	3.4294E-12	1.6815E+01
	Clasificación	1	2	5	4	3	6
	Significación		+	+	+	+	+
Clasificación final		1	2	5	4	3	6
W/T/L		-/-/-	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0

De acuerdo a los resultados de la Tabla 5.2.3, el algoritmo HE-DEPSO destaca al lograr los mejores resultados en optimización para la función χ^2 , superando a DEPSO. Aunque DEPSO se acerca a los resultados de HE-DEPSO, no logra igualar su precisión. El algoritmo DE también logró una buena precisión, sin embargo, esta fue inferior a la alcanzada por HE-DEPSO y DEPSO. Los algoritmos SHADE, CoDE y PSO obtuvieron los peores resultados en comparación con los demás. En términos de clasificación media, HE-DEPSO lidera entre todos los algoritmos evaluados. Además, los resultados del test de Wilcoxon confirman la superioridad de HE-DEPSO sobre DEPSO, SHADE, CoDE, DE y PSO en la optimización de la función χ^2 .

En la Figura 5.2.2, se muestran las curvas de convergencia de la medida de error de solución para la función χ^2 , obtenidas por los algoritmos HE-DEPSO, DEPSO, SHADE, CoDE, DE y PSO, a lo largo de las 31 repeticiones independientes. El algoritmo HE-DEPSO presenta un excelente comportamiento de convergencia, aunque el algoritmo DEPSO converge más rápidamente a partir aproximadamente de la iteración número 300. Sin embargo, el algoritmo HE-DEPSO destaca por superar en todo momento a los algoritmos SHADE, CoDE, DE y PSO, y por converger hacia un valor más bajo de media de error que el resto de algoritmos. La convergencia prolongada del algoritmo HE-DEPSO en comparación con el algoritmo DEPSO puede deberse a una mayor capacidad de exploración y al mecanismo mediante el cual actualiza sus parámetros.

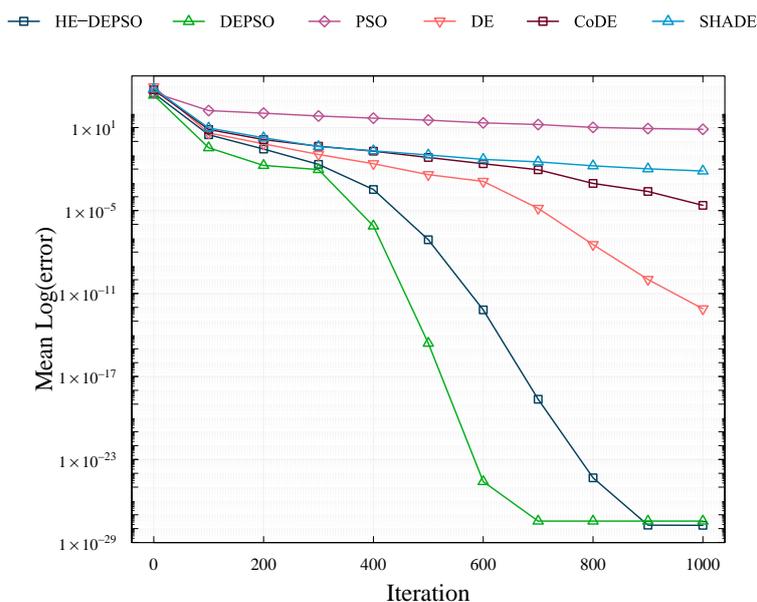


Figura 5.2.2: Curvas de convergencia de la medida de error de solución para la función χ^2 . El eje horizontal y vertical representan la iteraciones y los valores de media de error para las 31 repeticiones independientes.

En la Figura 5.2.3, se muestran los diagramas de cajas de los algoritmos HE-DEPSO, DEPSO, SHADE, CoDE, DE y PSO, basados en los mejores valores de ajuste global de 31 repeticiones en la función χ^2 . Estos diagramas representan la distribución de estos valores, divididos en cuartiles, con la mediana marcada por una línea horizontal roja en un recuadro azul. Los límites del recuadro corresponden a los cuartiles superior ($Q3$) e inferior ($Q1$). Las líneas que se extienden desde el recuadro indican los valores máximo y mínimo, excluyendo los valores atípicos representados por el símbolo “+” en azul, que caen fuera del rango definido por $Q1 - 1.5 \cdot (Q3 - Q1)$ y $Q3 + 1.5 \cdot (Q3 - Q1)$. Los círculos verdes representan la media de los mejores valores de ajuste global de las 31 repeticiones.

A partir de los diagramas de cajas, se encontró que el algoritmo HE-DEPSO supera a los demás algoritmos comparados en términos de estabilidad y calidad de solución en 31 repeticiones independientes. La distribución de los mejores valores de ajuste global obtenidos por HE-DEPSO muestra una mayor concentración alrededor de un mejor óptimo, reflejado en una media más baja en comparación con los otros algoritmos, lo que indica un rendimiento superior y estable.

Al estudiar las características del algoritmo HE-DEPSO en la optimización de la función χ^2 , en comparación con otros cinco algoritmos, se demostró que HE-

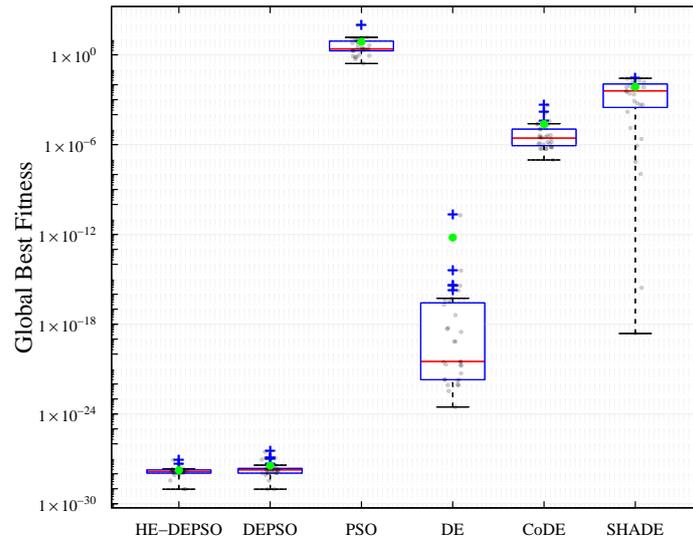


Figura 5.2.3: Diagramas de caja de los mejores valores de ajuste global obtenidos por HE-DEPSO y los algoritmos DEPSO, SHADE, CoDE, DE y PSO en las 31 repeticiones independientes en la función χ^2 . En eje horizontal se tienen a los algoritmos a comparar y en el eje vertical valores de ajuste global.

DEPSO mantiene una alta efectividad y eficiencia al resolver este problema de optimización, mostrando además un comportamiento muy estable. Cabe destacar que este estudio previo evidencia la dificultad que representa optimizar la función χ^2 , ya que variantes avanzadas del algoritmo DE, como SHADE y CoDE, presentaron claras dificultades para encontrar valores de ajuste más bajos en comparación al algoritmo base DE.

5.2.3. Resultados y discusión

En esta sección se presentan los resultados obtenidos tras la búsqueda de regiones válidas correspondientes a los parámetros A_u , A_d , ϕ_1 y ϕ_2 mediante la aplicación del algoritmo HE-DEPSO. Este proceso se llevó a cabo siguiendo el procedimiento detallado en la Subsección 5.2.2, haciendo uso de los valores experimentales más recientes asociados a los elementos de la matriz V_{CKM} , así como a las masas de los quarks y al invariante de Jarlskog (ver Ecuaciones (2.3.37), (2.3.38) y (5.2.2)).

Para cada caso de estudio, según se especifica en las Tablas 5.2.1 y 5.2.2, la estrategia de búsqueda se enfocó en identificar las regiones de los parámetros que satisficieran las siguientes precisiones para el criterio χ_{red}^2 propuesto:

$$\chi_{red}^2 < 1, \quad (5.2.7)$$

$$\chi_{red}^2 < 1 \times 10^{-1}, \quad (5.2.8)$$

$$\chi_{red}^2 < 1 \times 10^{-2}. \quad (5.2.9)$$

La selección específica de estas precisiones se ha realizado con el propósito de analizar de manera más detallada los datos obtenidos. Para la obtención de los datos en cada nivel de precisión y en cada caso de estudio, se ha ejecutado el algoritmo HE-DEPSO (considerando el criterio χ_{red}^2) de manera iterativa hasta recolectar 5000 puntos o soluciones óptimas. En este estudio, se logró identificar un total de 60000 soluciones óptimas. En cada ejecución, una vez se ha alcanzado la precisión requerida, se ha procedido a almacenar la solución encontrada en un archivo externo y reiniciar la búsqueda.

5.2.3.1. Regiones permitidas para A_u , A_d , ϕ_1 y ϕ_2

Con las soluciones encontradas con ayuda del algoritmo HE-DEPSO, se encontraron regiones permitidas para los parámetros A_u , A_d , ϕ_1 y ϕ_2 , en cada uno de los cuatro casos de estudio. En la Figura 5.2.4 se grafican las regiones permitidas para los parámetros A_u y A_d escaladas como A_u/m_t y A_d/m_b en todos los casos de estudio. Estas regiones, generadas con las soluciones encontradas, se dividen en tres colores según la precisión de χ_{red}^2 : negro (< 1), azul ($< 1 \times 10^{-1}$) y naranja ($< 1 \times 10^{-2}$). Bajo una precisión de $\chi_{red}^2 < 1$, se encontraron dos regiones distintas.

Para los casos de estudio I y III, estas dos regiones se sitúan entre rangos similares con valores:

- Caso de estudio I.

$$\text{Región 1: } \frac{A_u}{m_t} \in [0.3519, 0.4260], \quad \frac{A_d}{m_b} \in [0.3711, 0.4210]$$

$$\text{Región 2: } \frac{A_u}{m_t} \in [0.6486, 0.8728], \quad \frac{A_d}{m_b} \in [0.6884, 0.8729]$$

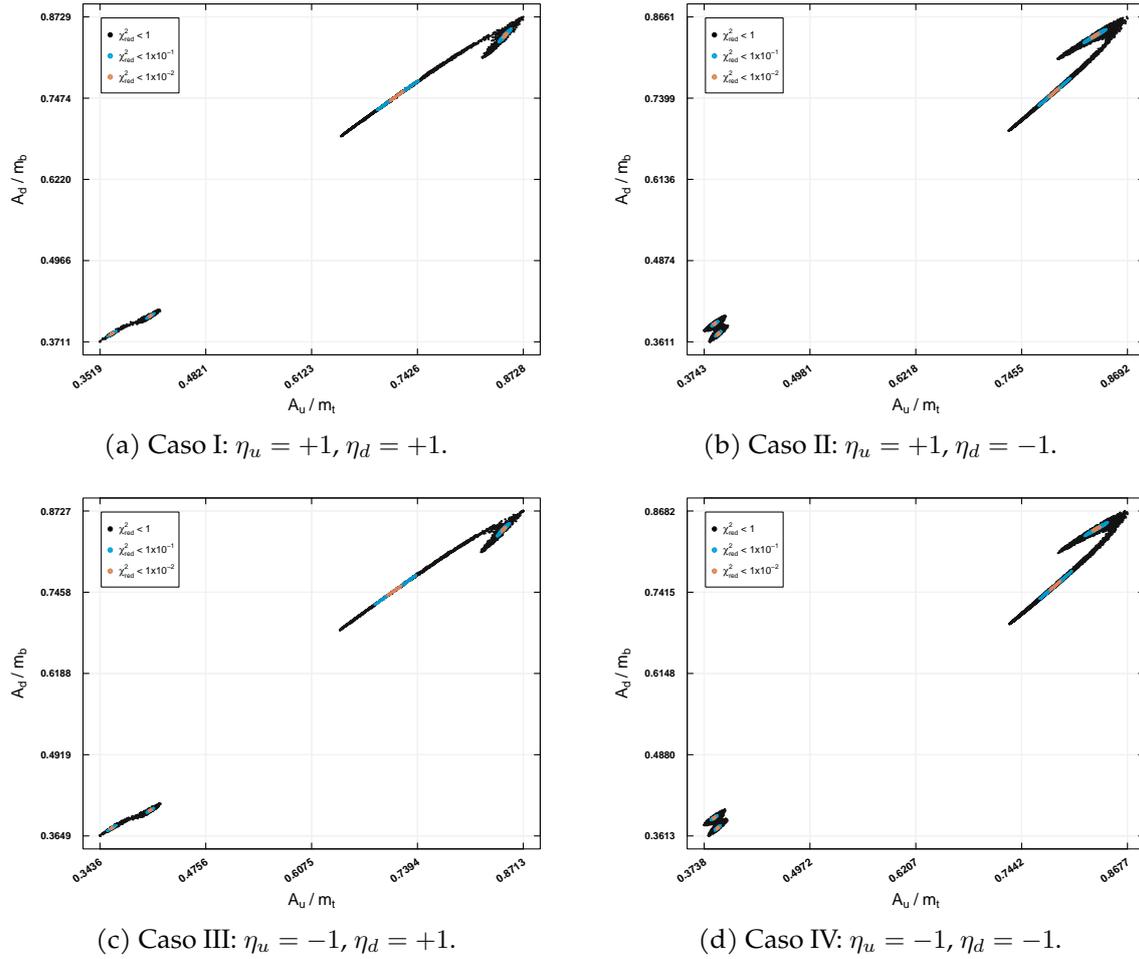


Figura 5.2.4: Regiones permitidas para A_u/m_t y A_d/m_b restringidas según datos experimentales actuales con diferentes niveles de precisión: $\chi_{Red}^2 < 1$ (color negro), $\chi_{Red}^2 < 1 \times 10^{-1}$ (color azul) y $\chi_{Red}^2 < 1 \times 10^{-2}$ (color naranja), en cada uno de los cuatro casos de estudio.

■ Caso de estudio III.

$$\text{Región 1: } \frac{A_u}{m_t} \in [0.3436, 0.4191], \quad \frac{A_d}{m_b} \in [0.3649, 0.4157]$$

$$\text{Región 2: } \frac{A_u}{m_t} \in [0.6427, 0.8713], \quad \frac{A_d}{m_b} \in [0.6865, 0.8727]$$

Para los casos de estudio II y IV, estas dos regiones aparecen en lugares similares dentro de los siguientes rangos de valores:

■ Caso de estudio II.

$$\text{Región 1: } \frac{A_u}{m_t} \in [0.3743, 0.4025], \quad \frac{A_d}{m_b} \in [0.3611, 0.4050]$$

$$\text{Región 2: } \frac{A_u}{m_t} \in [0.7303, 0.8692], \quad \frac{A_d}{m_b} \in [0.6890, 0.8661]$$

■ Caso de estudio IV.

$$\text{Región 1: } \frac{A_u}{m_t} \in [0.3738, 0.4010], \quad \frac{A_d}{m_b} \in [0.3613, 0.4050]$$

$$\text{Región 2: } \frac{A_u}{m_t} \in [0.7299, 0.8677], \quad \frac{A_d}{m_b} \in [0.6918, 0.8682]$$

En todos los casos de estudio analizados, las regiones correspondientes a las precisiones $\chi_{red}^2 < 1 \times 10^{-1}$ y $\chi_{red}^2 < 1 \times 10^{-2}$ se encuentran completamente contenidas dentro de la regiones permitidas para una precisión de $\chi_{red}^2 < 1 \times 10^{-2}$.

Para los parámetros ϕ_1 y ϕ_2 , en la Figura 5.2.5 se grafican sus regiones permitidas escaladas como ϕ_1/π y ϕ_2/π en todos los casos de estudio. La estructura con la que se muestran estas gráficas, es similar a la descrita para las regiones de los parámetros A_u y A_d . Para una precisión de $\chi_{red}^2 < 1$, se pueden distinguir al igual que para los parámetros A_u y A_d , dos regiones de valores permitidos.

En los casos de estudio I y IV, estas dos regiones se sitúan entre rangos similares con valores:

■ Caso de estudio I.

$$\text{Región 1: } \frac{\phi_1}{\pi} \in [0.530, 0.546], \quad \frac{\phi_2}{\pi} \in [1.960, 1.990]$$

$$\text{Región 2: } \frac{\phi_1}{\pi} \in [1.455, 1.470], \quad \frac{\phi_2}{\pi} \in [1.973, 1.976]$$

■ Caso de estudio IV.

$$\text{Región 1: } \frac{\phi_1}{\pi} \in [0.512, 0.527], \quad \frac{\phi_2}{\pi} \in [0.010, 0.040]$$

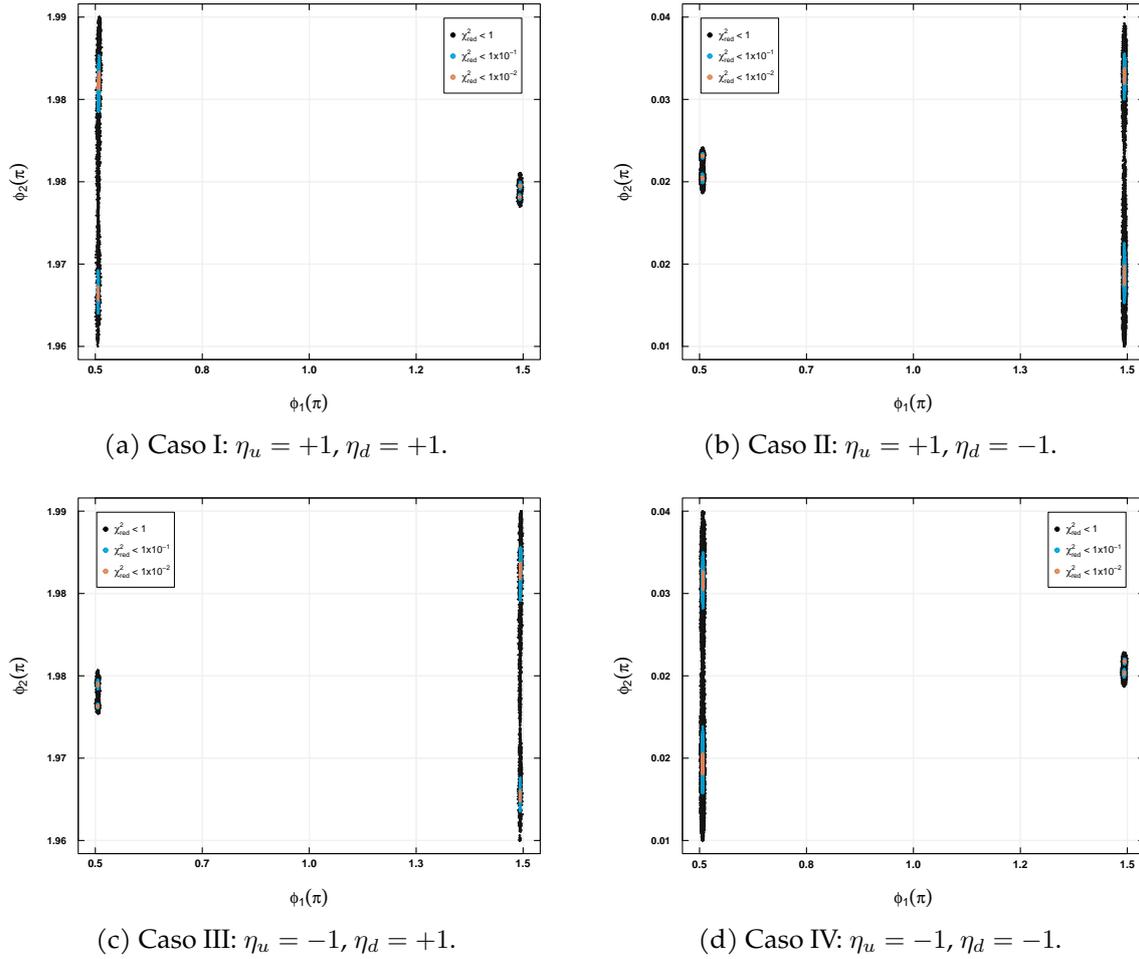


Figura 5.2.5: Regiones permitidas para $\phi_1/\pi, \phi_2/\pi$ restringidas según datos experimentales actuales con diferentes niveles de precisión: $\chi_{Res}^2 < 1$ (color negro), $\chi_{Red}^2 < 1 \times 10^{-1}$ (color azul) y $\chi_{Red}^2 < 1 \times 10^{-2}$ (color naranja), en cada uno de los cuatro casos de estudio.

$$\text{Región 2: } \frac{\phi_1}{\pi} \in [1.472, 1.487], \quad \frac{\phi_2}{\pi} \in [0.023, 0.026]$$

Para los casos de estudio II y III, estas dos regiones aparecen en lugares similares dentro de los siguientes rangos de valores:

■ Caso de estudio II.

$$\text{Región 1: } \frac{\phi_1}{\pi} \in [0.47, 0.49], \quad \frac{\phi_2}{\pi} \in [0.022, 0.027]$$

$$\text{Región 2: } \frac{\phi_1}{\pi} \in [1.51, 1.53], \quad \frac{\phi_2}{\pi} \in [0.010, 0.040]$$

■ Caso de estudio III.

$$\text{Región 1: } \quad \frac{\phi_1}{\pi} \in [0.455, 0.471], \quad \frac{\phi_2}{\pi} \in [1.972, 1.977]$$

$$\text{Región 2: } \quad \frac{\phi_1}{\pi} \in [1.530, 1.544], \quad \frac{\phi_2}{\pi} \in [1.960, 1.990]$$

Al igual que para los parámetros A_u y A_d , las regiones para las precisiones $\chi_{red}^2 < 1 \times 10^{-1}$ y $\chi_{red}^2 < 1 \times 10^{-2}$, se localizan dentro de la región permitida para una precisión de $\chi_{red}^2 < 1$, en todos los casos de estudio.

Considerando las gráficas y rangos de valores para todos los casos de estudio, se identificaron dos regiones destacadas para los parámetros A_u/m_t y A_d/m_b : la Región 1 (R_1) con valores entre 0.34 y 0.43 para A_u/m_t y entre 0.36 y 0.42 para A_d/m_b , y la Región 2 (R_2) con valores entre 0.64 y 0.87 para ambos parámetros. Esto permite considerar la aproximación $R = A_u/m_t \simeq A_d/m_b$, dividiendo a R en dos regiones: una cercana a 0.4 ($R \sim 0.4$) y otra cercana a 0.8 ($R \sim 0.8$). Si bien A_u/m_t y A_d/m_b parecen ser insensibles (en general) al signo de η_u y η_d en cada caso de estudio, se encontró una mayor sensibilidad al signo de η_d . Cuando $\eta_d = +1$ independientemente del signo de η_u (casos I y III), se observa una distribución específica de las soluciones o puntos dentro de las regiones R_1 y R_2 . Por el contrario, cuando $\eta_d = -1$ independientemente del signo de η_u (casos II y IV), se tiene otra distribución de las soluciones encontradas en dichas regiones.

Después de analizar los gráficos y rangos de valores para los parámetros ϕ_1 y ϕ_2 en todos los casos de estudio, se observó que el rango permitido para ϕ_1 es amplio en comparación con ϕ_2 . Se puede dividir en dos partes aproximadas: $\phi_1 \sim 0.5\pi$ y $\phi_1 \sim 1.5\pi$. En contraste, para ϕ_2 se encontró que en los casos I y III se acerca a 2π , por lo que su rango permitido puede expresarse como $\phi_2 \lesssim 2\pi$; mientras que en los casos II y IV se acerca a 0, lo que permite representar su rango como $\phi_2 \gtrsim 0$. Tras el análisis de los cuatro casos de estudio, se presentan en la Tabla 5.2.4 las correlaciones obtenidas entre R , ϕ_1 y ϕ_2 .

Tras estos resultados, es crucial destacar que se lograron encontrar soluciones óptimas para los parámetros A_u , A_d , ϕ_1 y ϕ_2 , validando la coherencia del modelo de

Tabla 5.2.4: Correlaciones entre R , ϕ_1 y ϕ_2 en los cuatro casos de estudio.

Caso de estudio	$R \sim 0.4$		$R \sim 0.8$	
I	$\phi_1 \sim 1.5\pi$,	$\phi_2 \lesssim 2\pi$	$\phi_1 \sim 0.5\pi$,	$\phi_2 \lesssim 2\pi$
II	$\phi_1 \sim 0.5\pi$,	$\phi_2 \gtrsim 0$	$\phi_1 \sim 1.5\pi$,	$\phi_2 \gtrsim 0$
III	$\phi_1 \sim 0.5\pi$,	$\phi_2 \lesssim 2\pi$	$\phi_1 \sim 1.5\pi$,	$\phi_2 \lesssim 2\pi$
IV	$\phi_1 \sim 1.5\pi$,	$\phi_2 \gtrsim 0$	$\phi_1 \sim 0.5\pi$,	$\phi_2 \gtrsim 0$

texturas de cuatro ceros con datos experimentales actuales, aunque sin confirmar su viabilidad. El descubrimiento de las regiones $R \sim 0.4$ y $R \sim 0.8$ revela dos regiones distintas en el espacio de parámetros de las matrices de masa M_u y M_d , respaldando investigaciones previas [89]. Estas soluciones óptimas han permitido definir con mayor precisión estas regiones, incluso identificando subregiones potencialmente significativas previamente no consideradas.

5.2.3.2. Predicciones de los elementos de la matriz V_{CKM}

Para verificar la viabilidad del modelo de texturas de cuatro ceros de Fritsch, se pretende reproducir los elementos de la matriz V_{CKM} . Dado que dentro de la construcción del criterio de χ_{red}^2 establecido para analizar la compatibilidad del modelo con los datos experimentales actuales, se ajustan tres elementos ($|V_{us}|$, $|V_{ub}|$, $|V_{cb}|$) más el invariante de Jarlskog (J), a continuación se presentan las predicciones para los elementos restantes de la matriz V_{CKM} ($|V_{ud}|$, $|V_{cd}|$, $|V_{cs}|$, $|V_{tb}|$, $|V_{td}|$ y $|V_{ts}|$), obtenidas a través del algoritmo HE-DEPSO.

A partir de los datos obtenidos por el algoritmo HE-DEPSO para encontrar soluciones que cumplieran con el criterio χ_{red}^2 , se calcula S_{13} , S_{12} , S_{23} y δ_{13} utilizando las Ecuaciones (2.3.32) a (2.3.36). Estos valores, junto con la parametrización Chau-Keng (ver Ecuación (2.3.31)), permiten calcular las predicciones numéricas para los elementos $|V_{ud}|$, $|V_{cd}|$, $|V_{cs}|$, $|V_{tb}|$, $|V_{td}|$ y $|V_{ts}|$.

Las predicciones de los elementos de la matriz V_{CKM} se muestran en gráficas de dispersión que representan los valores de los pares: $|V_{ud}|$ vs $|V_{cd}|$, $|V_{cs}|$ vs $|V_{tb}|$ y $|V_{td}|$ vs $|V_{ts}|$. Cada gráfica incluye el valor experimental central de cada elemento junto con su error asociado. Los puntos que caen dentro de la región delimitada

por los errores de cada valor experimental tienen una buena predicción, mientras que los puntos fuera de esta región no la tienen.

En la Figura 5.2.6, se presentan las predicciones para los elementos $|V_{ud}|$ y $|V_{cd}|$ en los cuatro casos de estudio. Los puntos negros, azules y naranjas indican una precisión de $\chi_{red}^2 < 1$, $\chi_{red}^2 < 1 \times 10^{-1}$ y $\chi_{red}^2 < 1 \times 10^{-2}$, respectivamente. Para $|V_{cd}|$, el valor central experimental (0.22636) se indica con una línea sólida roja en el eje correspondiente, mientras que su error experimental (± 0.00048) se muestra con dos líneas punteadas rojas a cada lado del valor central. En el caso de $|V_{ud}|$, el valor central experimental (0.97401) se representa con una línea sólida azul en el eje $|V_{ud}|$, y su error experimental (± 0.00011) se indica con dos líneas punteadas azules a ambos lados del valor central. Las predicciones para los dos elementos muestran que los datos con una precisión de $\chi_{red}^2 < 1$ pueden estar fuera de la región de errores experimentales, lo que sugiere una predicción deficiente. Por otro lado, las predicciones con $\chi_{red}^2 < 1 \times 10^{-1}$ y $\chi_{red}^2 < 1 \times 10^{-2}$ se mantienen dentro de esta región, indicando predicciones acertadas. Este patrón se repite en los cuatro casos de estudio analizados.

Las predicciones para los elementos $|V_{cs}|$ y $|V_{tb}|$ se muestran en la Figura 5.2.7 para los cuatro casos de estudio. El valor central experimental de $|V_{cs}|$ se representa con una línea sólida roja en el centro del eje $|V_{cs}|$ en 0.97320, mientras que su error experimental de ± 0.00011 se indica con líneas punteadas rojas a cada lado. El valor central experimental de $|V_{tb}|$ se muestra con una línea sólida azul en el centro del eje $|V_{tb}|$ en 0.999172, y su error experimental de $\pm 0.000024^1$ se representa con líneas punteadas azules a cada lado. Según las predicciones mostradas en la Figura 5.2.7, los datos obtenidos con $\chi_{red}^2 < 1$ no representan buenas predicciones, ya que quedan fuera de los errores experimentales. Por otro lado, las predicciones con $\chi_{red}^2 < 1 \times 10^{-1}$ y $\chi_{red}^2 < 1 \times 10^{-2}$ se mantienen dentro de la región de error, lo que indica que son buenas predicciones. Este patrón se repite en los cuatro casos de estudio.

Finalmente, en la Figura 5.2.8, se muestran las predicciones para los elementos $|V_{td}|$ y $|V_{ts}|$ en los cuatro casos de estudio. Para $|V_{td}|$, se representa el valor central

¹Se ha considerado el valor de error menor

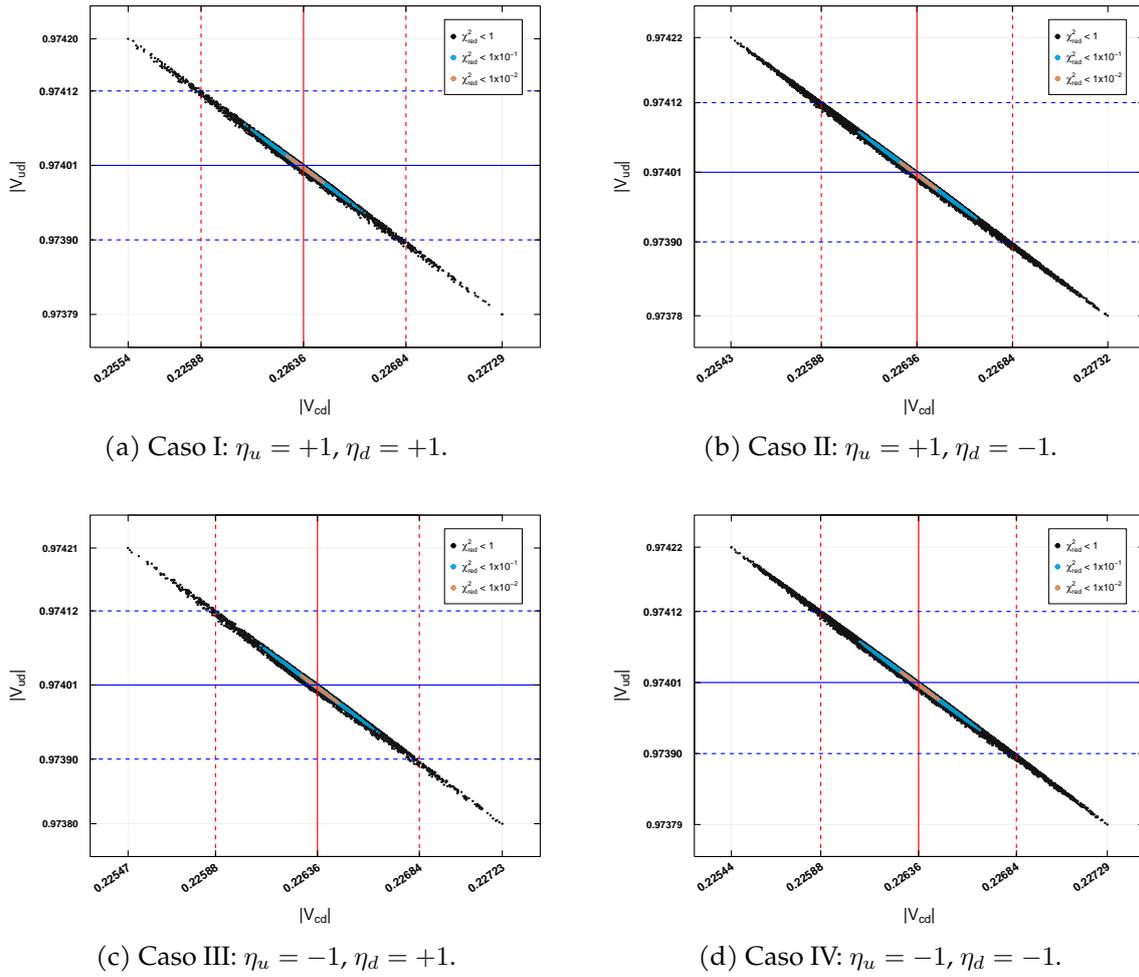


Figura 5.2.6: Predicción de $|V_{cd}|$ y $|V_{ud}|$ en cada uno de los cuatro casos de estudio.

experimental con una línea sólida roja en $|V_{td}| = 0.00854$, y su error experimental con dos líneas punteadas rojas a ± 0.00016 . Para $|V_{ts}|$, el valor central experimental se muestra con una línea sólida azul en $|V_{ts}| = 0.03978$, y su error experimental con dos líneas punteadas azules a ± 0.00060 . Las predicciones para estos dos elementos, indican que los datos obtenidos con precisiones de $\chi_{red}^2 < 1$ y $\chi_{red}^2 < 1 \times 10^{-1}$ pueden estar fuera de los errores experimentales, por lo que no son buenas predicciones. En contraste, la predicción con $\chi_{red}^2 < 1 \times 10^{-2}$ permanece dentro de los errores, siendo una buena predicción. Estas características se observan en los cuatro casos de estudio.

Al analizar las predicciones de los elementos de la matriz V_{CKM} ($|V_{ud}|$, $|V_{cd}|$, $|V_{cs}|$, $|V_{tb}|$, $|V_{td}|$ y $|V_{ts}|$), se encontró que estas predicciones son insensibles a los signos de η_u y η_d . Por otro lado, se observó que solo las soluciones obtenidas por el algoritmo

HE-DEPSO con una precisión de $\chi_{red}^2 < 1 \times 10^{-2}$ se ajustan dentro de los límites de error experimental. Estas soluciones se consideran válidas y capaces de reproducir la matriz V_{CKM} en todos los casos de estudio y elementos. Las predicciones con $\chi_{red}^2 < 1$ y $\chi_{red}^2 < 1 \times 10^{-1}$ no cumplen con este requisito, especialmente para V_{td} , por lo que no se consideran válidas. Con base en los resultados obtenidos, se puede establecer que la viabilidad del modelo de texturas de cuatro ceros, considerando los datos experimentales actuales, se encuentra limitada dependiendo de la precisión de χ_{red}^2 utilizada. A pesar de esto, resulta interesante explorar las implicaciones que esta información tiene dentro del formalismo de texturas.

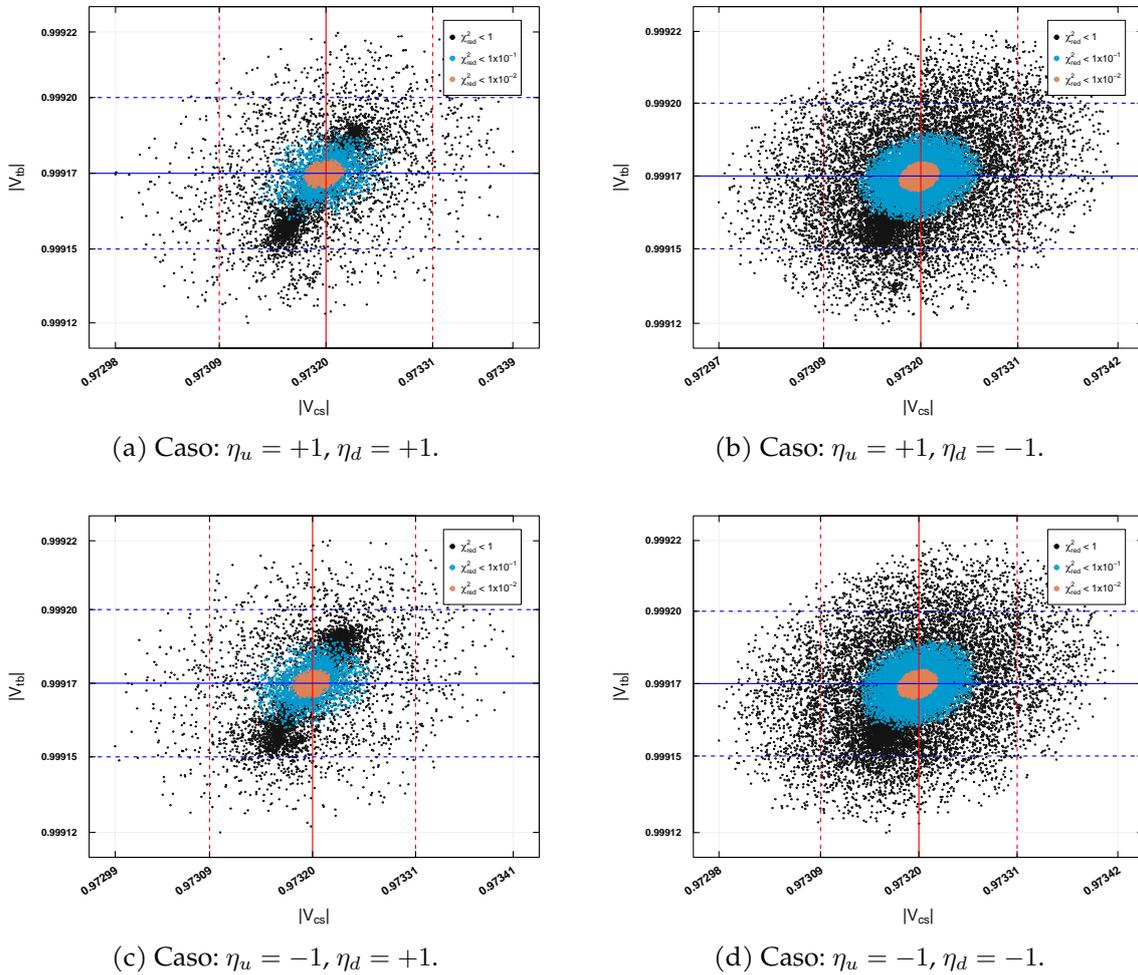


Figura 5.2.7: Predicción de $|V_{cs}|$ y $|V_{tb}|$ en cada uno de los cuatro casos de estudio.

5.2.3.3. Espacio completo de parámetros de las matrices de masa de quarks

Utilizando la información sobre las regiones válidas para A_u/m_t y A_d/m_b , junto

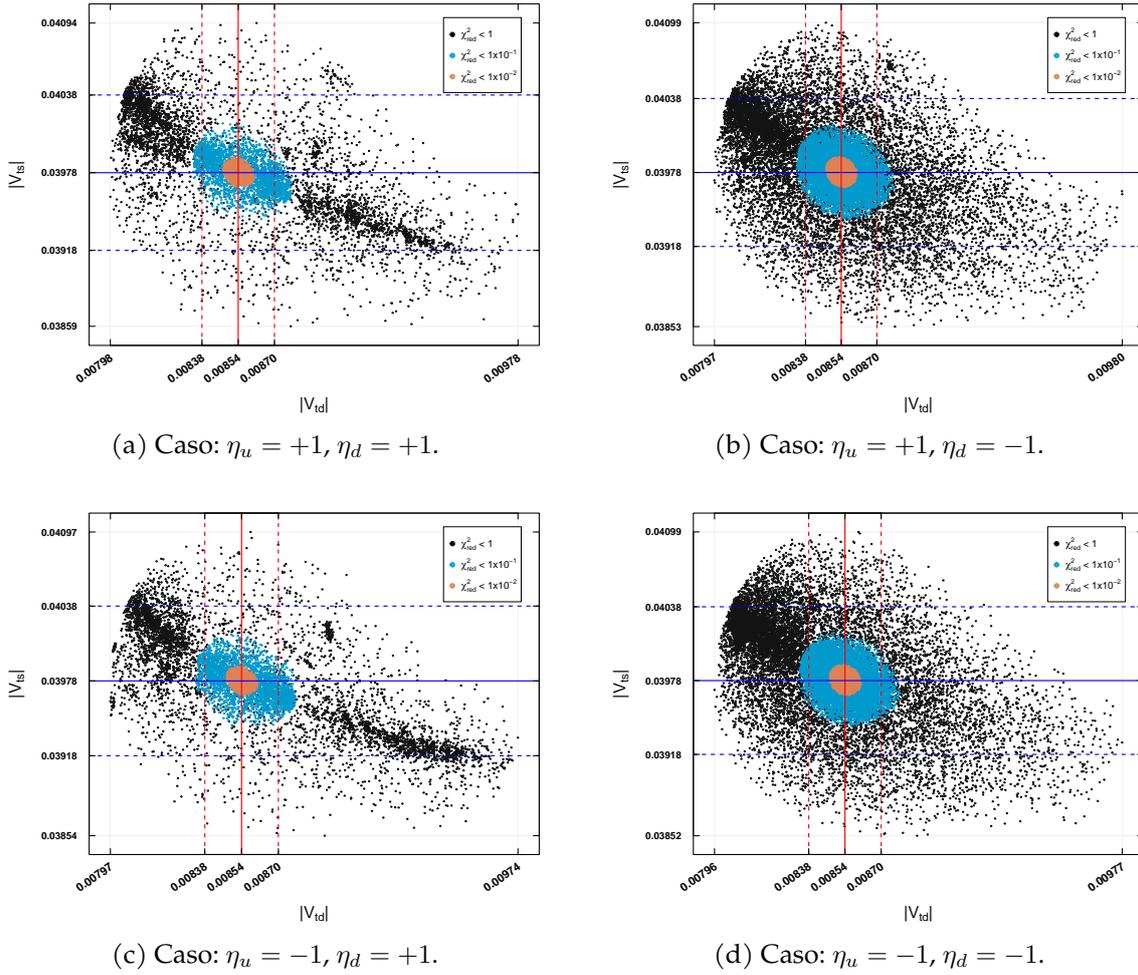


Figura 5.2.8: Predicción de $|V_{td}|$ y $|V_{ts}|$ en cada uno de los cuatro casos de estudio.

con las Ecuaciones (5.1.10) a (5.1.12), es posible explorar el espacio completo de los elementos no nulos restantes de las matrices de masa de quarks M_u y M_d : $|B_q|$, C_q , y $|D_q|$. Esta exploración puede proporcionar información valiosa sobre las relaciones entre los elementos y sus implicaciones fenomenológicas.

En la Figura 5.2.9 se presentan las regiones permitidas para $|B_q|$, escaladas por $|B_u|/A_u$ y $|B_d|/A_d$. En la Figura 5.2.10 se muestran las regiones válidas para C_q , escaladas por C_u/A_u y C_d/A_d . Por último, en la Figura 5.2.11 se exhiben las regiones correspondientes a $|D_q|$, escaladas por $|D_u|/A_u$ y $|D_d|/A_d$, abarcando los cuatro casos de estudio.

Al analizar los resultados numéricos de las regiones permitidas para los elementos restantes de la matriz V_{CKM} , se identificaron dos áreas representativas asociadas

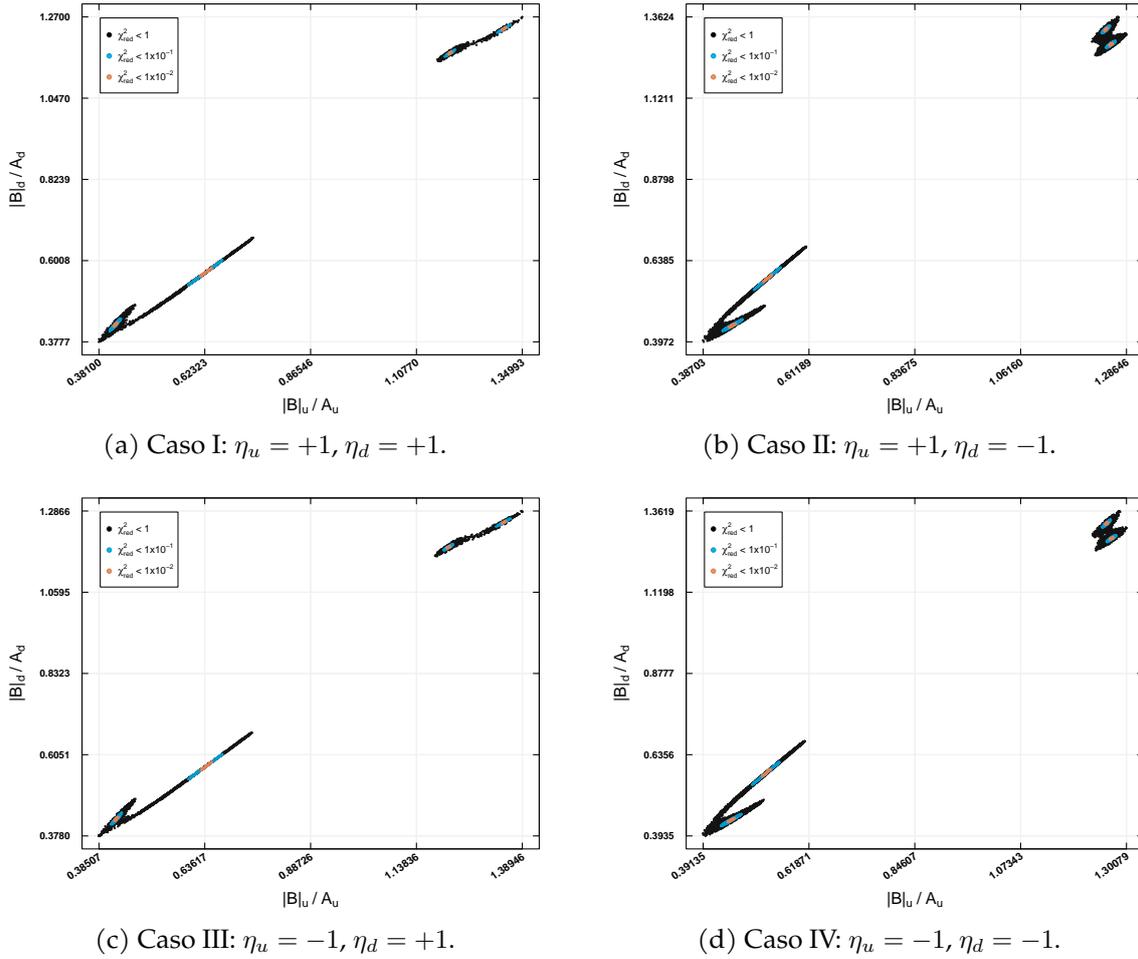


Figura 5.2.9: Regiones permitidas para $|B_u|/A_u$ y $|B_d|/A_d$ restringidas según datos experimentales actuales con diferentes niveles de precisión: $\chi_{Red}^2 < 1$ (color negro), $\chi_{Red}^2 < 1 \times 10^{-1}$ (color azul) y $\chi_{Red}^2 < 1 \times 10^{-2}$ (color naranja), en cada uno de los cuatro casos de estudio.

a las regiones $R \sim 0.4$ y $R \sim 0.8$ mencionadas dentro de la Subsección 5.2.3.1. En la región $R \sim 0.8$, las soluciones válidas que reproducen la matriz V_{CKM} permiten establecer una aproximación de los valores de estos parámetros: $|B_u|/A_u = |B_d|/A_d \sim 0.42$, $C_u/A_u \sim 0.17$, $C_d/A_d \sim 0.20$, $|D_u|/A_u \sim 0.0002$, y $|D_d|/A_d \sim 0.005$. En consecuencia, numéricamente, las matrices de masa de quarks M_u y M_d para $R \sim 0.8$ son:

$$\bar{M}_u \approx A_u \begin{pmatrix} 0 & 0.0002 & 0 \\ 0.0002 & 0.17 & 0.42 \\ 0 & 0.42 & 1 \end{pmatrix}, \quad (5.2.10)$$

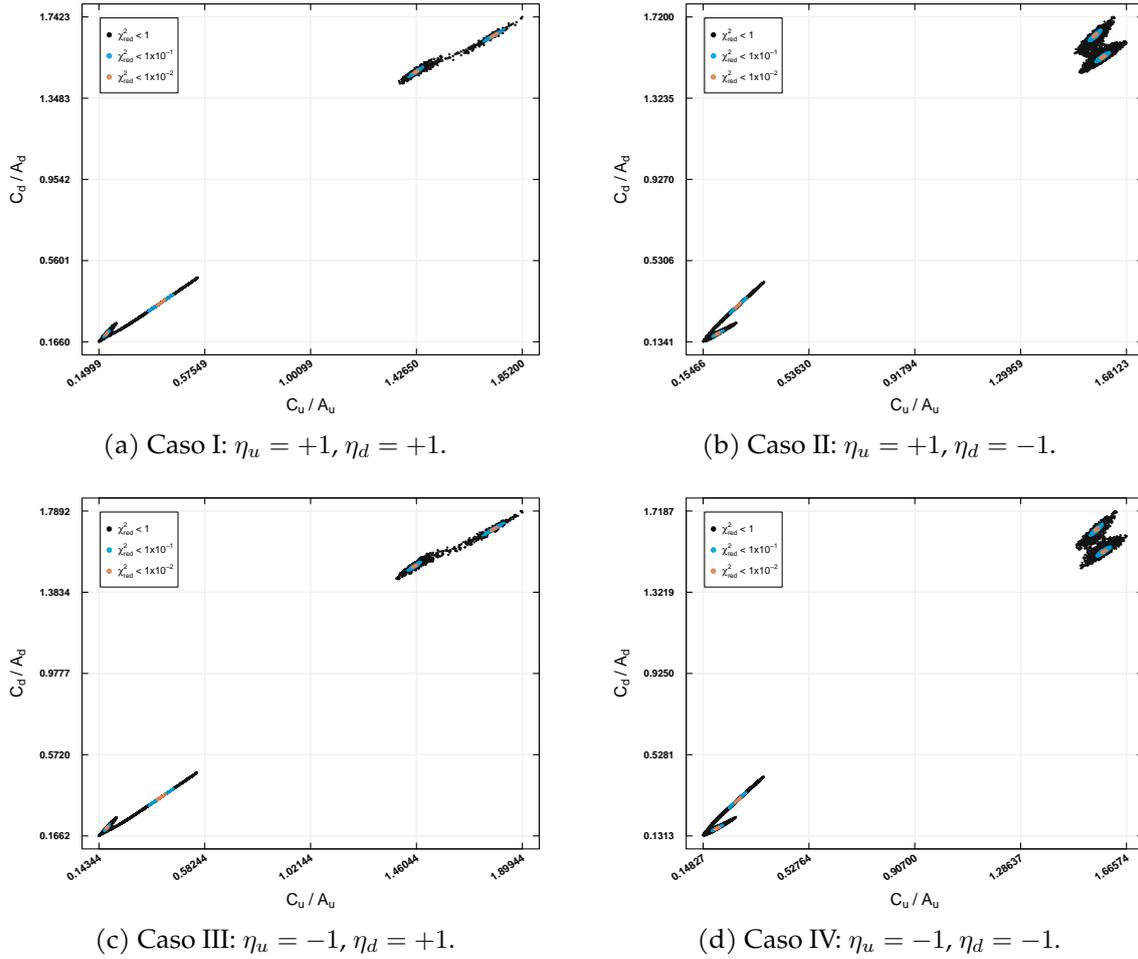


Figura 5.2.10: Regiones permitidas para C_u/A_u y C_d/A_d restringidas según datos experimentales actuales con diferentes niveles de precisión: $\chi_{Red}^2 < 1$ (color negro), $\chi_{Red}^2 < 1 \times 10^{-1}$ (color azul) y $\chi_{Red}^2 < 1 \times 10^{-2}$ (color naranja), en cada uno de los cuatro casos de estudio.

$$\bar{M}_d \approx A_d \begin{pmatrix} 0 & 0.005 & 0 \\ 0.005 & 0.20 & 0.42 \\ 0 & 0.42 & 1 \end{pmatrix}, \quad (5.2.11)$$

lo que permite observar que en la región $R \sim 0.8$ se tiene una fuerte jerarquía entre los elementos $(1, 2)$ y $(2, 2)$ de las matrices de masa de quarks M_u y M_d , mientras que se identifica una jerarquía débil entre los elementos $(2, 2)$, $(2, 3)$ y $(3, 3)$ de las matrices M_u y M_d . Esto implica que se cumple la relación $A_q > |B_q| > C_q$.

Para la región $R \sim 0.4$, las soluciones válidas que reproducen la matriz V_{CKM} permiten aproximar los parámetros de la siguiente manera: $|B_u|/A_u \sim 1.25, |B_d|/A_d \sim$

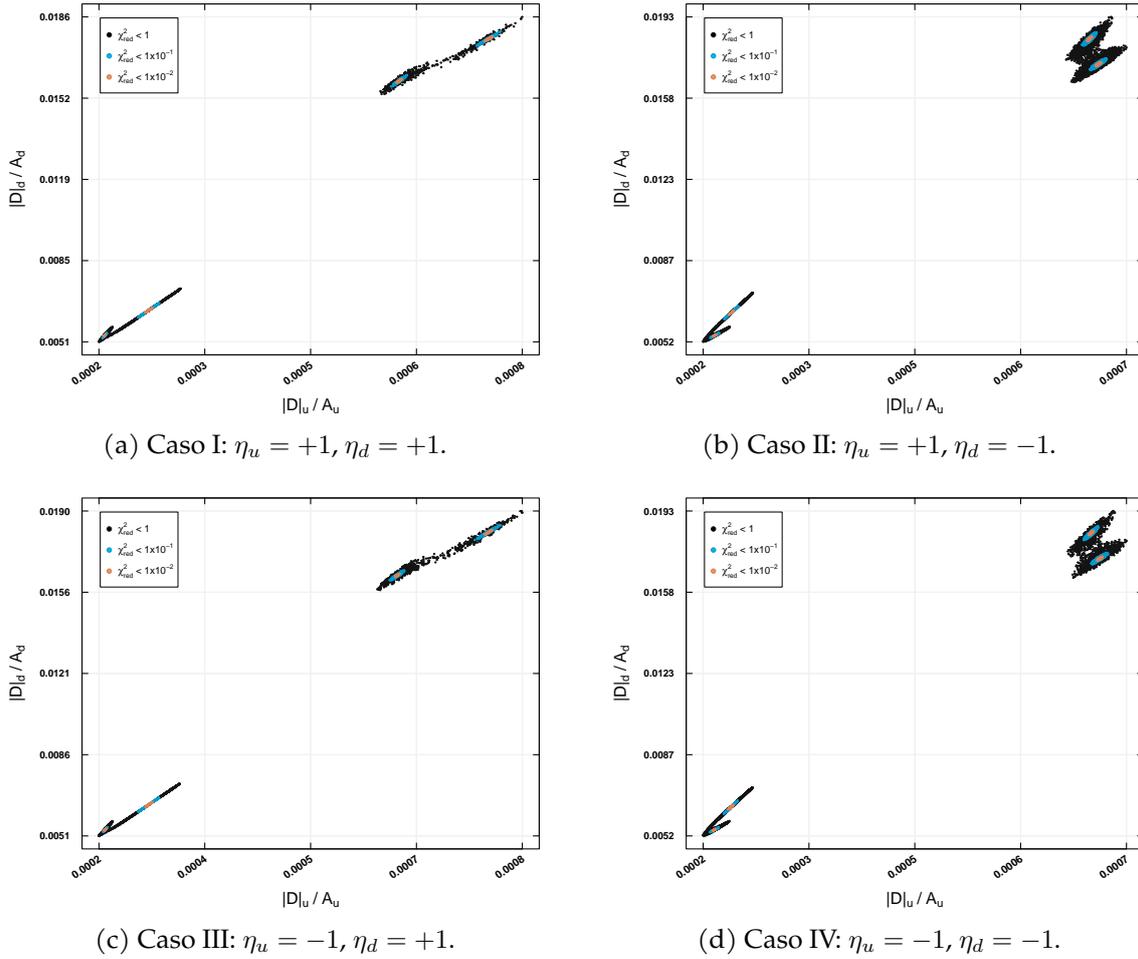


Figura 5.2.11: Regiones permitidas para $|D_u|/A_u$ y $|D_d|/A_d$ restringidas según datos experimentales actuales con diferentes niveles de precisión: $\chi_{Red}^2 < 1$ (color negro), $\chi_{Red}^2 < 1 \times 10^{-1}$ (color azul) y $\chi_{Red}^2 < 1 \times 10^{-2}$ (color naranja), en cada uno de los cuatro casos de estudio.

1.32, $C_u/A_u \sim 1.58$, $C_d/A_d \sim 1.52$, $|D_u|/A_u \sim 0.0006$, y $|D_d|/A_d \sim 0.017$. Numéricamente, esto implica que las matrices de masa de quarks M_u y M_d para $R \sim 0.4$ son:

$$\bar{M}_u \approx A_u \begin{pmatrix} 0 & 0.0006 & 0 \\ 0.0006 & 1.58 & 1.25 \\ 0 & 1.25 & 1 \end{pmatrix}, \quad (5.2.12)$$

$$\bar{M}_d \approx A_d \begin{pmatrix} 0 & 0.017 & 0 \\ 0.017 & 1.52 & 1.32 \\ 0 & 1.32 & 1 \end{pmatrix}, \quad (5.2.13)$$

de las cuales se puede observar una jerarquía distinta entre los elementos de las matrices de masa M_u y M_d a la presentada en la región $R \sim 0.8$. En este caso, se mantiene la fuerte jerarquía entre el elemento $(1, 2)$ y los elementos $(2, 2)$, $(2, 3)$ y $(3, 3)$. La particularidad de la región $R \sim 0.4$ se muestra dentro del sector $(2, 3)$ de las matrices de masa M_u y M_d , en donde se tiene una jerarquía diferente entre los elementos $(2, 2)$, $(2, 3)$ y $(3, 3)$, que pudiera permitir la aproximación $A_q \sim |B_q| \sim C_q$.

El análisis del espacio completo de parámetros de las matrices de masa de quarks M_u y M_d revela el efecto de las dos regiones identificadas, $R \sim 0.4$ y $R \sim 0.8$. En la región $R \sim 0.8$, las matrices de masa se caracterizan por una fuerte jerarquía entre los elementos $(1, 2)$ y $(2, 2)$, y una jerarquía débil entre los elementos $(2, 2)$, $(2, 3)$ y $(3, 3)$. Por otro lado, en la región $R \sim 0.4$, las propiedades estructurales de las matrices de masa permiten posibles casos en los que $A_q \sim |B_q| \sim C_q$. Cabe destacar que estas propiedades se mantienen para cada una de las subregiones dentro de $R \sim 0.4$ y $R \sim 0.8$, generadas con las soluciones válidas que reproducen la matriz V_{CKM} en los cuatro casos de estudio y en cada uno de los elementos $|B_q|$, C_q , y $|D_q|$.

Dentro del Apéndice E se discute la posibilidad de restringir cada uno de los términos de ajuste que componen la función χ^2 utilizada dentro del criterio de χ^2_{red} propuesto en esta investigación.

Capítulo 6

Conclusiones y trabajo futuro.

En este trabajo se analizó numéricamente el modelo de texturas de cuatro ceros para las matrices de masa de quarks M_u y M_d , utilizando el método de ajuste de χ^2 para vincular el modelo teórico con los datos experimentales actuales. Debido a la dificultad que representa optimizar la función χ^2 , se propuso el uso de técnicas de optimización avanzadas, como los ABI. En particular, se desarrolló una nueva variante del algoritmo DEPSO denominada HE-DEPSO, que combina una estrategia de mutación basada en la información histórica de los mejores individuos y la adaptación de parámetros del algoritmo SHADE. Con este algoritmo, se encontraron las regiones válidas de los parámetros libres del modelo de texturas de cuatro ceros de Fritzsche que mejor se ajustan a los datos experimentales disponibles, lo que permitió indagar en la compatibilidad del modelo con los datos actuales, identificar su viabilidad y explorar el rango completo de los parámetros de las matrices de masa de quarks M_u y M_d .

Las principales conclusiones de esta tesis se resumen en los siguientes puntos:

- Se demostró que optimizar la función χ^2 dentro del análisis numérico del modelo de texturas de cuatro ceros es un desafío significativo. Mediante la visualización de gráficas que muestran diferentes proyecciones de los parámetros de la función, se logró identificar múltiples mínimos locales, lo que demuestra la complejidad de este problema de optimización.
- Se propuso una nueva variante del algoritmo DEPSO denominada HE-DEPSO para la optimización de la función χ^2 en el análisis numérico del modelo de texturas de cuatro ceros.
- Incorporar la información histórica de los mejores individuos mediante una nueva estrategia de mutación, así como adoptar el control de parámetros

del algoritmo SHADE, permitieron mejorar el equilibrio entre exploración y explotación del algoritmo DEPSO.

- El algoritmo HE-DEPSO demostró ser una herramienta eficaz y estable para la optimización, obteniendo buenos resultados tanto en el conjunto de funciones de prueba CEC 2017 como en la optimización de la función χ^2 , lo que lo convierte en una opción viable y prometedora para resolver problemas de optimización complejos.
- La implementación del algoritmo HE-DEPSO para optimizar la función χ^2 en el análisis numérico del modelo de texturas de cuatro ceros ha promovido el uso de técnicas bio-inspiradas dentro de la FAE.
- El algoritmo HE-DEPSO permitió encontrar un total de 60,000 soluciones óptimas para los parámetros libres del modelo de texturas con cuatro ceros (A_u , A_d , ϕ_1 y ϕ_2). Estas soluciones se distribuyeron entre los cuatro casos de estudio y las diferentes precisiones solicitadas según el criterio χ_{red}^2 . Esto demuestra la compatibilidad del modelo con los datos experimentales actuales.
- El algoritmo HE-DEPSO fue capaz de encontrar la precisión más baja y de forma consistente del criterio χ_{red}^2 respecto a otros algoritmos.
- Se confirmó la presencia de dos regiones válidas ($R \sim 0.4$ y $R \sim 0.8$) para el espacio completo de parámetros de las matrices de masa de quarks M_u y M_d .
- Se descubrió que la viabilidad del modelo de texturas de cuatro ceros se ve limitada por la precisión de χ_{red}^2 empleada. Solo las soluciones que cumplieron con una precisión de $\chi_{red}^2 < 1 \times 10^{-2}$ pudieron reproducir con éxito los elementos de la matriz $|V_{CKM}|$ dentro de los límites experimentales en todos los casos analizados. Esto condujo a la identificación de dos subregiones dentro de $R \sim 0.4$ y $R \sim 0.8$.
- Se encontró que en la región $R \sim 0.4$ se preserva la fuerte jerarquía entre los elementos $(3, 3)$, $(2, 3)$, $(2, 2)$ y el elemento $(2, 2)$ de las matrices de masa de quarks, mientras que en la región $R \sim 0.8$ los elementos A_q , $|B_q|$ y C_q

de la matriz de masa tienden a tener valores similares, aproximándose a la relación $A_q \sim |B_q| \sim C_q$.

6.1. Aportaciones

Las aportaciones de este trabajo de tesis son las siguientes:

- Un algoritmo bio-inspirado que combina la potencia de diferentes enfoques para resolver problemas complejos con un rendimiento superior a las técnicas metaheurísticas bio-inspiradas comúnmente utilizadas en física de partículas, como PSO y DE.
- Un estudio experimental del algoritmo propuesto, validado mediante pruebas estadísticas en diferentes funciones de prueba y comparación con otros algoritmos bio-inspirados.
- Un análisis detallado sobre la viabilidad del modelo de texturas con cuatro ceros, basado en los datos experimentales actuales.

6.2. Trabajo futuro

El trabajo futuro que puede surgir a partir de esta tesis se resume en los siguientes puntos:

- Analizar numéricamente la viabilidad de otros modelos de texturas con los datos experimentales más recientes.
- Ampliar la capacidad de optimización del algoritmo HE-DEPSO para abordar problemas de optimización multi-objetivo y con restricciones.
- Aplicar el algoritmo HE-DEPSO en la solución de otros problemas de optimización de la vida real, especialmente en problemas con gran número de parámetros o dimensiones.
- Comparar el algoritmo HE-DEPSO contra otros algoritmos y en otros conjuntos de funciones de prueba para verificar su robustez.

Referencias

- [1] Z. Beheshti and S. M. Shamsuddin, "A review of population-based meta-heuristic algorithm," *International Journal of Advances in Soft Computing and Its Applications*, vol. 5, pp. 1–35, 03 2013.
- [2] G. Altarelli and F. Feruglio, "Neutrino masses and mixings: A theoretical perspective," *Phys. Rept.*, vol. 320, pp. 295–318, 1999.
- [3] G. Altarelli and F. Feruglio, "Models of neutrino masses and mixings," *New J. Phys.*, vol. 6, p. 106, 2004.
- [4] H. Fritzsch and Z.-z. Xing, "Mass and flavor mixing schemes of quarks and leptons," *Prog. Part. Nucl. Phys.*, vol. 45, pp. 1–81, 2000.
- [5] Z.-z. Xing, "Flavor structures of charged fermions and massive neutrinos," *Phys. Rept.*, vol. 854, pp. 1–147, 2020.
- [6] M. Gupta and G. Ahuja, "Flavor mixings and textures of the fermion mass matrices," *Int. J. Mod. Phys. A*, vol. 27, p. 1230033, 2012.
- [7] C. D. Froggatt and H. B. Nielsen, "Hierarchy of Quark Masses, Cabibbo Angles and CP Violation," *Nucl. Phys. B*, vol. 147, pp. 277–298, 1979.
- [8] S. F. King and C. Luhn, "Neutrino Mass and Mixing with Discrete Symmetry," *Rept. Prog. Phys.*, vol. 76, p. 056201, 2013.
- [9] F. Borzumati and Y. Nomura, "Low scale seesaw mechanisms for light neutrinos," *Phys. Rev. D*, vol. 64, p. 053005, 2001.
- [10] M. Lindner, T. Ohlsson, and G. Seidl, "Seesaw mechanisms for Dirac and Majorana neutrino masses," *Phys. Rev. D*, vol. 65, p. 053014, 2002.
- [11] W. Flieger and J. Gluza, "General neutrino mass spectrum and mixing properties in seesaw mechanisms," *Chin. Phys. C*, vol. 45, no. 2, p. 023106, 2021.

- [12] S. F. Novaes, "Standard model: An Introduction," in *10th Jorge Andre Swieca Summer School: Particle and Fields*, pp. 5–102, 1 1999.
- [13] P. Langacker, "Introduction to the Standard Model and Electroweak Physics," in *Theoretical Advanced Study Institute in Elementary Particle Physics: The Dawn of the LHC Era*, pp. 3–48, 2010.
- [14] G. Isidori, Y. Nir, and G. Perez, "Flavor Physics Constraints for Physics Beyond the Standard Model," *Ann. Rev. Nucl. Part. Sci.*, vol. 60, p. 355, 2010.
- [15] H. Fritzsch, "Calculating the cabibbo angle," *Physics Letters B*, vol. 70, no. 4, pp. 436–440, 1977.
- [16] L. Tani, D. Rand, C. Veelken, and M. Kadastik, "Evolutionary algorithms for hyperparameter optimization in machine learning for application in high energy physics," *The European Physical Journal C*, vol. 81, feb 2021.
- [17] A. Hofler, B. c. v. Terzić, M. Kramer, A. Zvezdin, V. Morozov, Y. Roblin, F. Lin, and C. Jarvis, "Innovative applications of genetic algorithms to problems in accelerator physics," *Phys. Rev. ST Accel. Beams*, vol. 16, Jan 2013.
- [18] B. C. Allanach, D. Grellscheid, and F. Quevedo, "Genetic algorithms and experimental discrimination of susy models," *Journal of High Energy Physics*, vol. 2004, p. 69, aug 2004.
- [19] S. Wang, Y. Li, and H. Yang, "Self-adaptive mutation differential evolution algorithm based on particle swarm optimization," *Applied Soft Computing*, vol. 81, p. 105496, Aug. 2019.
- [20] S. Kiranyaz, T. Ince, and M. Gabbouj, *Multidimensional particle swarm optimization for machine learning and pattern recognition*. Adaptation, learning, and optimization, Berlin, Germany: Springer, 2014 ed., July 2013.
- [21] M. Tomassini, *The Parallel Genetic Cellular Automata: Application to Global Function Optimization*, p. 385–391. Springer Vienna, 1993.
- [22] E. Alba and B. Dorronsoro, *Cellular Genetic Algorithms*, vol. 42. 01 2008.
- [23] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley, Mar. 2003.

- [24] A. P. Engelbrecht, *Computational Intelligence: An Introduction*. Wiley, hardcover ed., 11 2007.
- [25] Thomas Back, *Evolutionary algorithms in theory and practice*. New York, NY: Oxford University Press, Jan. 1996.
- [26] A. Torn and A. Zilinskas, *Global Optimization*. Lecture notes in computer science, Berlin, Germany: Springer, 1989 ed., Feb. 1989.
- [27] T. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, p. 284–294, 2000.
- [28] A. Vasuki, *Nature-Inspired Optimization Algorithms*. Chapman and Hall/CRC, May 2020.
- [29] T. Weise, *Global Optimization Algorithms - Theory and Application*. Self-Published, second ed., 2009 2009.
- [30] A. Leon, "A classified bibliography on optimization," *Recent advances in optimization techniques*, vol. 599, p. 649, 1966.
- [31] A. A. Törn, *Global optimization as a combination of global and local search*. Handelshögskolan vid Åbo akademi, 1974.
- [32] L. C. W. Dixon, "The global optimization problem: an introduction," *Towards Global Optimiation 2*, pp. 1–15, 1978.
- [33] A. E. Ezugwu, A. K. Shukla, R. Nath, A. A. Akinyelu, J. O. Agushaka, H. Chiroma, and P. K. Muhuri, "Metaheuristics: a comprehensive overview and classification along with bibliometric analysis," *Artificial Intelligence Review*, vol. 54, pp. 4237–4316, Mar. 2021.
- [34] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, England: Cambridge University Press, Mar. 2004.
- [35] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 2012.
- [36] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, p. 400–407, Sept. 1951.

- [37] J. Rajpurohit, D. T. Sharma, and A. Abraham, "Glossary of metaheuristic algorithms," pp. 181–205, 01 2017.
- [38] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, p. 67–82, Apr. 1997.
- [39] *Handbook of Memetic Algorithms*. Springer Berlin Heidelberg, 2012.
- [40] W. Wong and C. I. Ming, "A review on metaheuristic algorithms: Recent trends, benchmarking and applications," in *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, pp. 1–5, 2019.
- [41] S. Zhao, T. Zhang, S. Ma, and M. Chen, "Dandelion optimizer: A nature-inspired metaheuristic algorithm for engineering applications," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105075, Sept. 2022.
- [42] M. Kaveh and M. S. Mesgari, "Application of meta-heuristic algorithms for training neural networks and deep learning architectures: A comprehensive review," *Neural Processing Letters*, vol. 55, p. 4519–4622, Oct. 2022.
- [43] S. B. Joseph, E. G. Dada, A. Abidemi, D. O. Oyewola, and B. M. Khammas, "Metaheuristic algorithms for pid controller parameters tuning: Review, approaches and open problems," *Heliyon*, 2022.
- [44] L. Liberti, "Introduction to global optimization," *Ecole Polytechnique*, 2008.
- [45] H. Rajabi Moshtaghi, A. Toloie Eshlaghy, and M. R. Motadel, "A comprehensive review on meta-heuristic algorithms and their classification with novel approach," *Journal of Applied Research on Industrial Engineering*, vol. 8, Mar. 2021.
- [46] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization," *ACM Computing Surveys*, vol. 35, pp. 268–308, Sept. 2003.
- [47] G. Dhiman and V. Kumar, "Emperor penguin optimizer: A bio-inspired algorithm for engineering problems," *Knowledge-Based Systems*, vol. 159, p. 20–50, Nov. 2018.
- [48] H. Stegherr, M. Heider, and J. Hähner, "Classifying metaheuristics: Towards a unified multi-level classification system," *Natural Computing*, vol. 21, p. 155–171, Dec. 2020.

- [49] S. Salih and A. Alsewari, "A new algorithm for normal and large-scale optimization problems: Nomadic people optimizer," *Neural Computing and Applications*, vol. 32, pp. 10359–10386, 07 2020.
- [50] S. Harifi, J. Mohammadzadeh, M. Khalilian, and S. Ebrahimnejad, "Giza pyramids construction: an ancient-inspired metaheuristic algorithm for optimization," *Evolutionary Intelligence*, vol. 14, 12 2021.
- [51] M. Kumar, A. J. Kulkarni, and S. C. Satapathy, "Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology," *Future Generation Computer Systems*, vol. 81, p. 252–272, Apr. 2018.
- [52] A. Yadav and D. K. Vishwakarma, "A comparative study on bio-inspired algorithms for sentiment analysis," *Cluster Computing*, vol. 23, p. 2969–2989, Feb. 2020.
- [53] B. Ismail, N. I. Abdul Wahab, M. L. Othman, M. A. Mohd Radzi, K. Naidu, and M. Mat Naain, "A comprehensive review on optimal location and sizing of reactive power compensation using hybrid-based approaches for power loss reduction, voltage stability improvement, voltage profile enhancement and loadability enhancement," *IEEE Access*, vol. 8, 12 2020.
- [54] R. Yadav, I. Sreedevi, and D. Gupta, "Bio-inspired hybrid optimization algorithms for energy efficient wireless sensor networks: A comprehensive review," *Electronics*, vol. 11, p. 1545, May 2022.
- [55] D. Rai and Seth, "Bio-inspired optimization techniques: a critical comparative study," *ACM SIGSOFT Software Engineering Notes*, vol. 38, pp. 1–7, 07 2013.
- [56] I. F. Jr., X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," 2013.
- [57] A. H. Gandomi, *Metaheuristic applications in structures and infrastructures*. Elsevier, 2013.
- [58] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*. Springer Berlin Heidelberg, 2015.

- [59] C. Darwin, *On the Origin of Species by Means of Natural Selection, Or, The Preservation of Favoured Races in the Struggle for Life*. The World's Classics, J. Murray, 1859.
- [60] L. A. Scardua, *Applied Evolutionary Algorithms for engineers using python*. CRC PRESS, 2023.
- [61] A. Brabazon, M. O'Neill, and S. McGarraghy, *Natural Computing Algorithms*. Springer Berlin Heidelberg, 2015.
- [62] X. Yu and M. Gen, *Introduction to Evolutionary Algorithms*. Springer London, 2010.
- [63] J. Holland, *Adaptation in natural and artificial systems*, vol. Reprinted by MIT Press, 1992. Ann Arbor, Michigan: University of Michigan Press, 1975.
- [64] D. Floreano and C. Mattiussi, *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*. The MIT Press, 2008.
- [65] A. Petrowski and S. Ben Hamida, *Evolutionary algorithms*, vol. Volume 9 of *Computer engineering series, metaheuristics set*. ISTE Ltd, John Wiley & sons , 2017.
- [66] R. Storn and K. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 23, 01 1995.
- [67] D. Simon, *Evolutionary Optimization Algorithms*. Wiley, 2013.
- [68] A. Q. H. Badar, *Evolutionary Optimization Algorithms*. CRC Press, Aug. 2021.
- [69] Bansal, *Evolutionary and Swarm Intelligence Algorithms (Studies in Computational Intelligence, 779)*. Springer, hardcover ed., 6 2018.
- [70] L. Brezočnik, I. Fister, and V. Podgorelec, "Swarm intelligence algorithms for feature selection: A review," *Applied Sciences*, vol. 8, no. 9, 2018.
- [71] E. Osaba and X.-S. Yang, eds., *Applied Optimization and Swarm Intelligence (Springer Tracts in Nature-Inspired Computing)*. Springer, hardcover ed., 5 2021.
- [72] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, IEEE, 1995.

- [73] A. G. Gad, "Particle swarm optimization algorithm and its applications: A systematic review," *Archives of Computational Methods in Engineering*, vol. 29, p. 2531–2561, Apr. 2022.
- [74] J. Kennedy, "The particle swarm: social adaptation of knowledge," in *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*, IEEE.
- [75] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, MHS-95, IEEE.
- [76] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pp. 69–73, 1998.
- [77] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, IEEE.
- [78] S. L. Glashow *Nucl. Phys.*, vol. 22, pp. 579–588, 1961.
- [79] A. Salam and J. Ward *Physics Letters*, vol. 13, no. 2, pp. 168–171, 1964.
- [80] S. Weinberg *Phys. Rev. Lett.*, vol. 19, pp. 1264–1266, Nov 1967.
- [81] S. L. Glashow, J. Iliopoulos, and L. Maiani *Phys. Rev. D*, vol. 2, pp. 1285–1292, Oct 1970.
- [82] H. Harari, "Three Generations of Quarks and Leptons," in *5th International Conference on Meson Spectroscopy*, p. 0170, 7 1977.
- [83] D. Griffiths, *Introduction to elementary particles*. Weinheim, Germany: Wiley-VCH Verlag, 2 ed., Aug. 2008.
- [84] H. Fritzsch, "The physics of flavor - challenge for the future," 2004.
- [85] L.-L. Chau and W.-Y. Keung, "Comments on the parametrization of the kobayashi-maskawa matrix," *Phys. Rev. Lett.*, vol. 53, pp. 1802–1805, Nov 1984.

- [86] J. P. S. Gustavo Castelo Branco, Luís Lavoura, *Cp Violation*. International Series of Monographs on Physics, Oxford University Press, USA, 1st ed., 1999.
- [87] P. Z. et al. (Particle Data Group), "Review of Particle Physics," *Progress of Theoretical and Experimental Physics*, vol. 2020, p. 083C01, 08 2020.
- [88] H. Fritzsch, Z. zhong Xing, and D. Zhang, "Correlations between quark mass and flavor mixing hierarchies," *Nuclear Physics B*, vol. 974, p. 115634, jan 2022.
- [89] Z. zhong Xing and Z. hua Zhao, "On the four-zero texture of quark mass matrices and its stability," *Nuclear Physics B*, vol. 897, pp. 302–325, aug 2015.
- [90] S. Kaundal, A. Bagai, G. Ahuja, and M. Gupta, "Minimal texture of quark mass matrices and precision ckm measurements," *Progress of Theoretical and Experimental Physics*, vol. 2020, Jan. 2020.
- [91] A. Bagai, S. Kaundal, G. Ahuja, and M. Gupta, "Going beyond the minimal texture of quark mass matrices in the era of precision measurements," *Progress of Theoretical and Experimental Physics*, vol. 2020, Sept. 2020.
- [92] S. Sharma, P. Fakay, G. Ahuja, and M. Gupta, "Finding a unique texture for quark mass matrices," *Physical Review D*, vol. 91, Mar. 2015.
- [93] O. Félix-Beltrán, F. González-Canales, J. Hernández-Sánchez, S. Moretti, R. Noriega-Papaqui, and A. Rosado, "Analysis of the quark sector in the 2hdm-iii with a four-zero yukawa texture using the most recent data on the ckm matrix," 2013.
- [94] "Review of particle physics," *Phys. Rev. D*, vol. 86, p. 010001, Jul 2012.
- [95] P. Ludl and W. Grimus, "A complete survey of texture zeros in general and symmetric quark mass matrices," *Physics Letters B*, vol. 744, p. 38–42, May 2015.
- [96] S. Antusch and V. Maurer, "Running quark and lepton parameters at various scales," *Journal of High Energy Physics*, vol. 2013, Nov. 2013.
- [97] K. Olive, "Review of particle physics," *Chinese Physics C*, vol. 38, p. 090001, Aug. 2014.

- [98] S. LUO and Z.-Z. XING, "THEORETICAL OVERVIEW ON THE FLAVOR ISSUES OF MASSIVE NEUTRINOS," *International Journal of Modern Physics A*, vol. 27, p. 1230031, nov 2012.
- [99] R. L. Workman., "Review of Particle Physics," *Progress of Theoretical and Experimental Physics*, vol. 2022, 08 2022. 083C01.
- [100] Y. Zhang, D. Zhou, *et al.*, "Application of differential evolution algorithm in future collider optimization," in *7th International Particle Accelerator Conference (IPAC'16), Busan, Korea, May 8-13*, pp. 1025–1027, 2016.
- [101] J. Wu, Y. Zhang, Q. Qin, Y. Wang, C. Yu, and D. Zhou, "Dynamic aperture optimization with diffusion map analysis at cepc using differential evolution algorithm," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 959, p. 163517, 2020.
- [102] Bilal, M. Pant, H. Zaheer, L. Garcia-Hernandez, and A. Abraham, "Differential evolution: A review of more than two decades of research," *Engineering Applications of Artificial Intelligence*, vol. 90, p. 103479, 2020.
- [103] M. F. Ahmad, N. A. M. Isa, W. H. Lim, and K. M. Ang, "Differential evolution: A recent review based on state-of-the-art works," *Alexandria Engineering Journal*, vol. 61, no. 5, p. 3831–3872, 2022.
- [104] J.-S. Pan, N. Liu, and S.-C. Chu, "A hybrid differential evolution algorithm and its application in unmanned combat aerial vehicle path planning," *IEEE Access*, vol. 8, pp. 17691–17712, 2020.
- [105] J. Zhang and A. Sanderson, "Jade: Adaptive differential evolution with optional external archive," *Evolutionary Computation, IEEE Transactions on*, vol. 13, pp. 945 – 958, 11 2009.
- [106] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, 2011.
- [107] F. Peñuñuri, C. Cab, O. Carvente, M. Zambrano-Arjona, and J. Tapia, "A study of the classical differential evolution control parameters," *Swarm and Evolutionary Computation*, vol. 26, p. 86–96, 2016.

- [108] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *2013 IEEE Congress on Evolutionary Computation*, pp. 71–78, 2013.
- [109] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
- [110] S. Chakraborty, A. K. Saha, S. Sharma, S. K. Sahoo, and G. Pal, "Comparative performance analysis of differential evolution variants on engineering design problems," *Journal of Bionic Engineering*, vol. 19, p. 1140–1160, June 2022.
- [111] X. Zhang, Y. Xue, X. Lu, and S. Jia, "Differential-evolution-based coevolution ant colony optimization algorithm for bayesian network structure learning," *Algorithms*, vol. 11, p. 188, Nov. 2018.
- [112] M. Song, K. Chen, and J. Wang, "Three-dimensional wind turbine positioning using gaussian particle swarm optimization with differential evolution," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 172, p. 317–324, Jan. 2018.
- [113] X. Zhong and P. Cheng, "An elite-guided hierarchical differential evolution algorithm," *Applied Intelligence*, vol. 51, p. 4962–4983, jan 2021.
- [114] Q. Yang, X. Guo, X.-D. Gao, D.-D. Xu, and Z.-Y. Lu, "Differential elite learning particle swarm optimization for global numerical optimization," *Mathematics*, vol. 10, no. 8, 2022.
- [115] Y. Huang, Y. Yu, J. Guo, and Y. Wu, "Self-adaptive artificial bee colony with a candidate strategy pool," *Applied Sciences*, vol. 13, no. 18, 2023.
- [116] S. Wang, Y. Li, H. Yang, and H. Liu, "Self-adaptive differential evolution algorithm with improved mutation strategy," *Soft Computing*, vol. 22, p. 3433–3447, Apr. 2017.
- [117] J. J. L. B. Y. Q. N. H. Awad, M. Z. Ali and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective bound constrained real-parameter numerical optimization," Technical Report, Nanyang Technological University, Singapore, 11 2016.

-
- [118] R. Andrae, T. Schulze-Hartung, and P. Melchior, "Dos and don'ts of reduced chi-squared," 2010.
- [119] M. GUPTA, G. AHUJA, and R. VERMA, "Implications of precision measurements on texture specific fermion mass matrices," *International Journal of Modern Physics A*, vol. 24, p. 3462–3468, July 2009.
- [120] R. Verma, G. Ahuja, N. Mahajan, M. Randhawa, and M. Gupta, "Exploring the parameter space of texture four-zero quark mass matrices," *Journal of Physics G: Nuclear and Particle Physics*, vol. 37, p. 075020, June 2010.

Apéndice A

Funciones de prueba CEC 2017

En este trabajo se hace uso de las funciones de prueba propuestas en [117] para evaluar el rendimiento de algoritmo propuesto. Este conjunto de funciones de prueba está conformado de 29¹ problemas de optimización global, el cual está organizado en cuatro grupos: dos funciones unimodales, siete funciones multimodales, diez funciones híbridas y diez funciones compuestas. En todos estos problemas se busca encontrar el mínimo global dentro de un espacio de búsqueda delimitado al intervalo $[-100, 100]$ en cada una de las dimensiones.

A.1. Funciones Básicas

1. Función Bent Cigar

$$f_1(\mathbf{X}) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2 \quad (\text{A.1.1})$$

2. Suma de dos funciones exponenciales

$$f_2(\mathbf{X}) = \sum_{i=1}^D |x_i|^{i+1} \quad (\text{A.1.2})$$

3. Función Zakharov

$$f_3(\mathbf{X}) = \sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D 0.5x_i \right)^2 + \left(\sum_{i=1}^D 0.5x_i \right)^4 \quad (\text{A.1.3})$$

¹La función F_2 ha sido excluida del conjunto de funciones de prueba debido a que muestra un comportamiento inestable especialmente en dimensiones altas.

4. Función de Rosenbrock

$$f_4(\mathbf{X}) = \sum_{i=1}^{D-1} \left[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right] \quad (\text{A.1.4})$$

5. Función de Rastrigin

$$f_5(\mathbf{X}) = 10D + \sum_{i=1}^D \left[x_i^2 - 10\cos(2\pi x_i) \right] \quad (\text{A.1.5})$$

6. Función de Schaffer ampliada

$$f_6(\mathbf{X}) = g(x_1, x_2) + g(x_2, x_3) + \dots + g(x_{D-1}, x_D) + g(x_D, x_1) \quad (\text{A.1.6})$$

donde: $g(x, y) = 0.5 + \frac{\text{sen}^2(\sqrt{x^2 + y^2}) - 0.5}{(1 + 0.001(x^2 + y^2))^2}$

7. Función Lunacek Bi-Rastrigin

$$f_7(\mathbf{X}) = \min \left(\sum_{i=1}^D (x_i - \mu_0)^2, dD + s \sum_{i=1}^D (x_i - \mu_1)^2 + 10(D - \sum_{i=1}^D \cos(2\pi z_i)) \right),$$

donde: $\mu_0 = 2.5$, $\mu_1 = -\sqrt{\frac{\mu_0^2 - d}{s}}$, $s = 1 - \frac{1}{2\sqrt{D + 20} - 8.2}$, $d = 1$ (A.1.7)

8. Función de Rastrigin discontinua y rotada

$$f_8(\mathbf{X}) = \sum_{i=1}^D (z_i^2 - 10\cos(2\pi z_i) + 10) + f_{13}^*,$$

donde:

$$z = M_1 \Lambda^{10} M_2 T_{asy}^{0.2}(T_{osz}(y)), \quad y_i = \begin{cases} x_i, & \text{si } |x_i| \leq 0.5 \\ \frac{\text{round}(2x_i)}{2}, & \text{si } |x_i| > 0.5 \end{cases}, \quad x = \frac{5.12}{100} M_1(x - o) \quad (\text{A.1.8})$$

donde: Λ^α es una matriz diagonal en D dimensiones con el i -ésimo elemento en la diagonal dado por $\lambda_{ii} = \alpha^{\frac{2(D-1)}{i-1}}$, $i = 1, 2, \dots, D$. Las definiciones de T_{asy} y T_{osz} se describen a mayor detalle en [117].

9. Función de Levy

$$f_9(\mathbf{X}) = \text{sen}^2(\pi w_1) + \sum_{i=1}^{D-1} (w_i - 1)^2 \left[1 + 10 \text{sen}^2(\pi w_i + 1) \right] + (w_D - 1)^2 \left[1 + \text{sen}^2(2\pi w_D) \right],$$

donde: $w_i = 1 + \frac{x_i - 1}{4}$ con $i = 1, \dots, D$.

(A.1.9)

10. Función de Schwefel modificada

$$f_{10}(\mathbf{X}) = 418.9829 \times D - \sum_{i=1}^D g(z_i),$$

donde:

$$z_i = x_i + 420.9687462275036,$$

$$g(z_i) = \begin{cases} z_i \text{sen}(\sqrt{|z_i|}), & \text{si } |z_i| \leq 500 \\ (500 - \text{mod}(z_i, 500))h(z_i) - \frac{(z_i - 500)^2}{10000D}, & \text{si } z_i > 500 \\ (\text{mod}(z_i, 500) - 500)h(z_i) - \frac{(z_i + 500)^2}{10000D}, & \text{si } z_i < -500 \end{cases}, \quad (\text{A.1.10})$$

$$h(z_i) = \text{sen}(\sqrt{|500 - \text{mod}(|z_i|, 500)|})$$

11. Función elíptica condicionada

$$f_{11}(\mathbf{X}) = \sum_{i=1}^D (10^6) \frac{i-1}{D-1} x_i^2 \quad (\text{A.1.11})$$

12. Función de disco

$$f_{12}(\mathbf{X}) = 10^6 x_1^2 + \sum_{i=2}^D x_i^2 \quad (\text{A.1.12})$$

13. Función de Ackley

$$f_{13}(\mathbf{X}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e \quad (\text{A.1.13})$$

14. Función de Weierstrass

$$f_{14}(\mathbf{X}) = \sum_{i=1}^D \left(\sum_{k=0}^{kmax} \left[a^k \cos(2\pi b^k (x_i + 0.5)) \right] \right) - D \sum_{k=0}^{kmax} \left[a^k \cos(\pi b^k) \right] \quad (\text{A.1.14})$$

donde: $a = 0.5$, $b = 3$, $kmax = 20$

15. Función de Griewank

$$f_{15}(\mathbf{X}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (\text{A.1.15})$$

16. Función de Katsuura

$$f_{16}(\mathbf{X}) = \frac{10}{D^2} \prod_{i=1}^D \left(1 + i \sum_{j=1}^{32} \frac{|2^j x_i - \text{round}(2^j x_i)|}{2^j} \right)^{\frac{10}{D^{1.2}}} - \frac{10}{D^2} \quad (\text{A.1.16})$$

17. Función HappyCat

$$f_{17}(\mathbf{X}) = \left| \sum_{i=1}^D x_i^2 - D \right|^{\frac{1}{4}} + \frac{1}{D} \sum_{i=1}^D \left(\frac{1}{2} x_i^2 + x_i \right) + \frac{1}{2} \quad (\text{A.1.17})$$

18. Función HGBat

$$f_{18}(\mathbf{X}) = \left| \left(\sum_{i=1}^D x_i^2 \right)^2 - \left(\sum_{i=1}^D x_i \right)^2 \right|^{\frac{1}{2}} + \frac{1}{D} \sum_{i=1}^D \left(\frac{1}{2} x_i^2 + x_i \right) + \frac{1}{2} \quad (\text{A.1.18})$$

19. Función de Griewank ampliada más la función de Rosenbrock

$$f_{19}(\mathbf{X}) = f_7(f_4(x_1, x_2)) + f_7(f_4(x_2, x_3)) + \dots + f_7(f_4(x_{D-1}, x_D)) + f_7(f_4(x_D, x_1)) \quad (\text{A.1.19})$$

20. Función de Schaffer F7

$$f_{20}(\mathbf{X}) = \left[\frac{1}{D-1} \sum_{i=1}^{D-1} \left(\sqrt{s_i} \left(\text{sen}(50s_i^{0.2}) + 1 \right) \right) \right]^2, \quad s_i = \sqrt{x_i^2 + x_{i+1}^2} \quad (\text{A.1.20})$$

A.2. Funciones unimodales

Función Bent Cigar rotada y desplazada

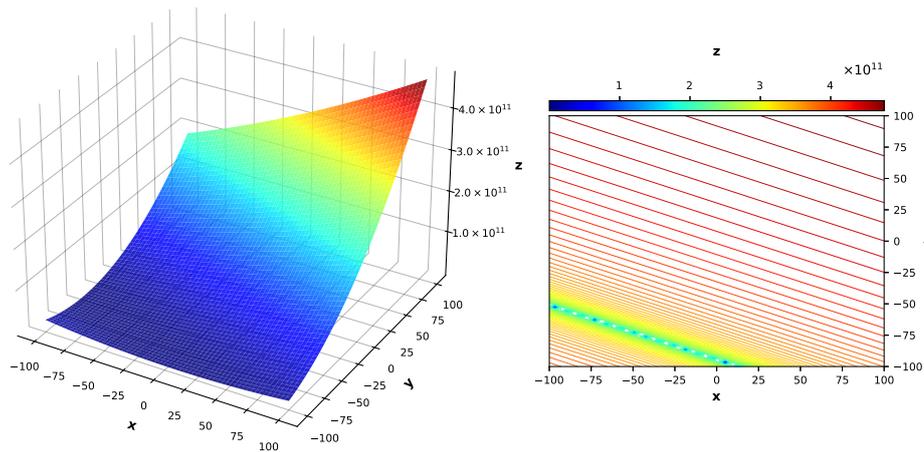
$$F_1(\mathbf{X}) = f_1(M(\mathbf{X} - O_1)) + 100 \quad (\text{A.2.1})$$

Suma de dos funciones exponenciales (rotada y desplazada)

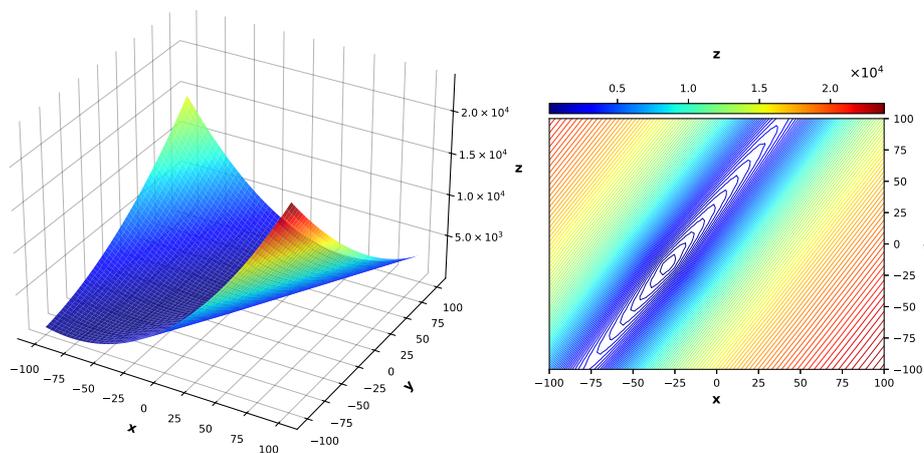
$$F_2(\mathbf{X}) = f_2(M(\mathbf{X} - O_2)) + 200 \quad (\text{A.2.2})$$

Función Zakharov rotada y desplazada

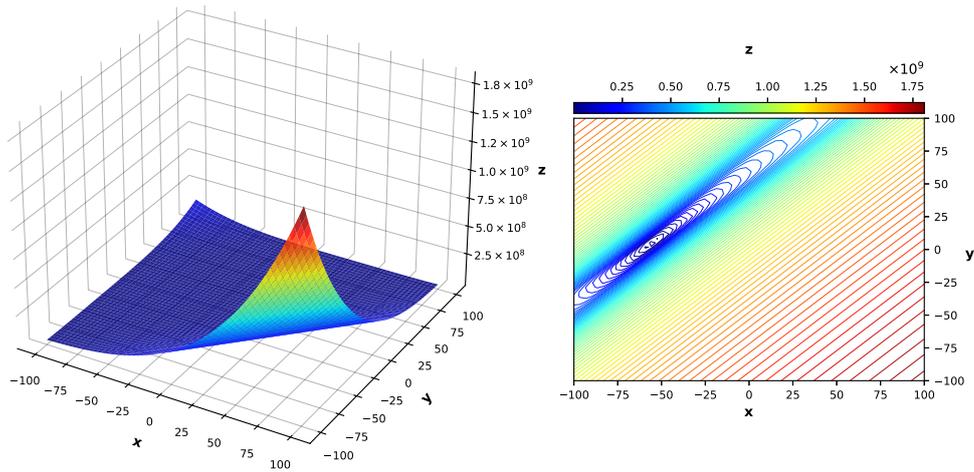
$$F_3(\mathbf{X}) = f_3(M(\mathbf{X} - O_3)) + 300 \quad (\text{A.2.3})$$



F_1



F_2



F3

Figura A.2.1: Funciones de prueba unimodales y sus respectivas curvas de nivel ($D = 2$).

A.3. Funciones multimodales

Función de Rosenbrock rotada y desplazada

$$F_4(\mathbf{X}) = f_4 \left(M \left(\frac{2.048(\mathbf{X} - O_4)}{100} \right) + 1 \right) + 400 \quad (\text{A.3.1})$$

Función de Rastrigin rotada y desplazada

$$F_5(\mathbf{X}) = f_5(M(\mathbf{X} - O_5)) + 500 \quad (\text{A.3.2})$$

Función de Schaffer F6 rotada y desplazada

$$F_6(\mathbf{X}) = f_{20} \left(M \left(\frac{0.5(\mathbf{X} - O_6)}{100} \right) \right) + 600 \quad (\text{A.3.3})$$

Función Lunacek Bi-Rastrigin rotada y desplazada

$$F_7(\mathbf{X}) = f_7 \left(M \left(\frac{600(\mathbf{X} - O_7)}{100} \right) \right) + 700 \quad (\text{A.3.4})$$

Función de Rastrigin discontinua y rotada

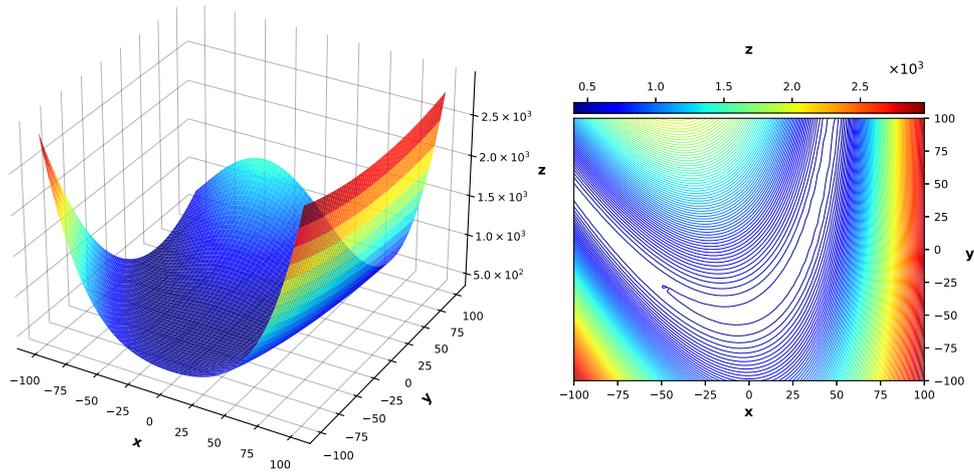
$$F_8(\mathbf{X}) = f_8 \left(M \left(\frac{5.12(\mathbf{X} - O_8)}{100} \right) \right) + 800 \quad (\text{A.3.5})$$

Función de Levy rotada y desplazada

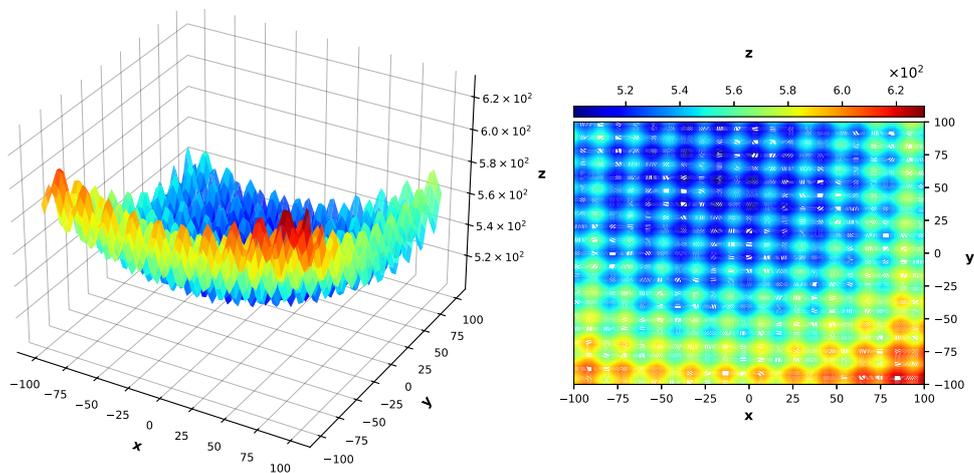
$$F_9(\mathbf{X}) = f_9 \left(M \left(\frac{5.12(\mathbf{X} - O_9)}{100} \right) \right) + 900 \quad (\text{A.3.6})$$

Función de Schwefel modificada

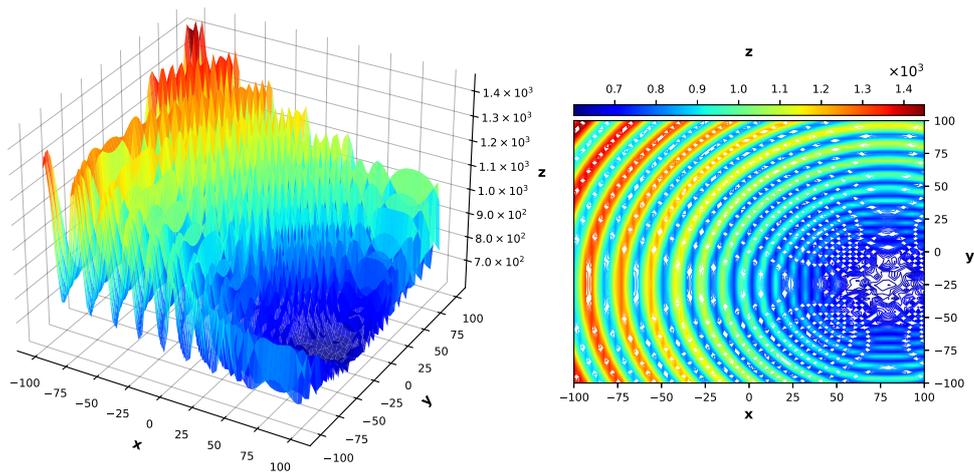
$$F_{10}(\mathbf{X}) = f_{10} \left(M \left(\frac{1000(\mathbf{X} - O_{10})}{100} \right) \right) + 1000 \quad (\text{A.3.7})$$



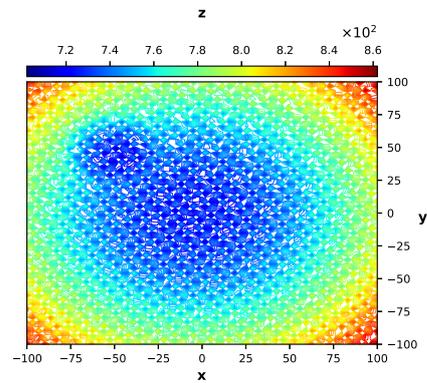
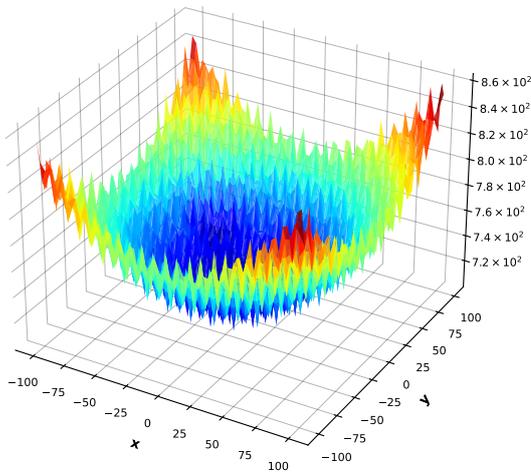
F4



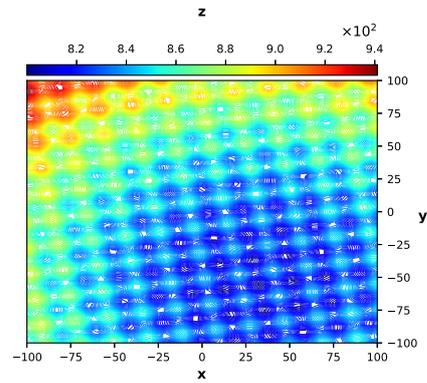
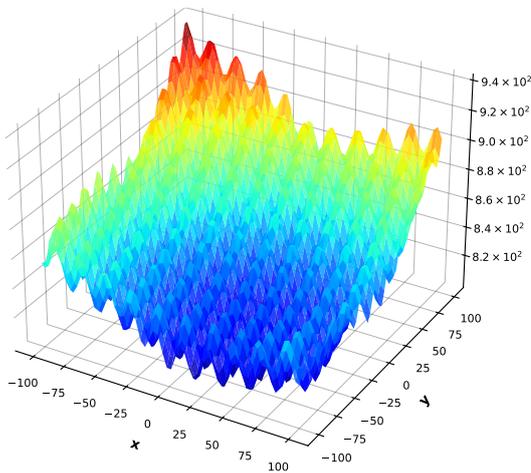
F5



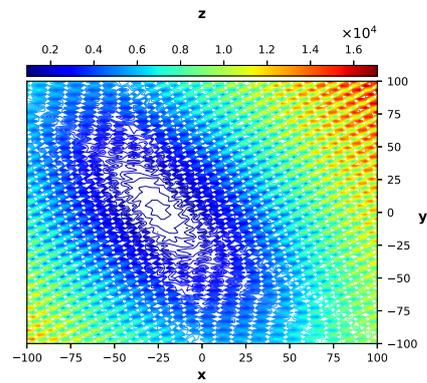
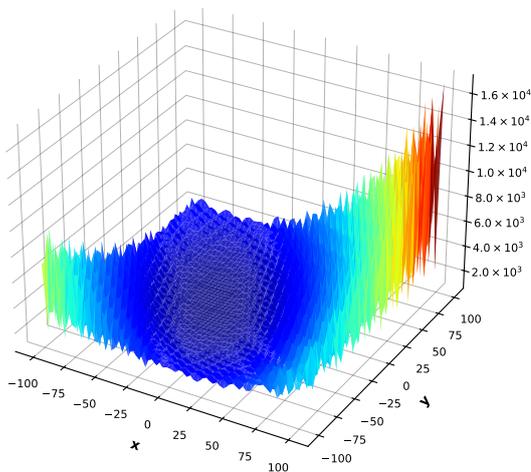
F6



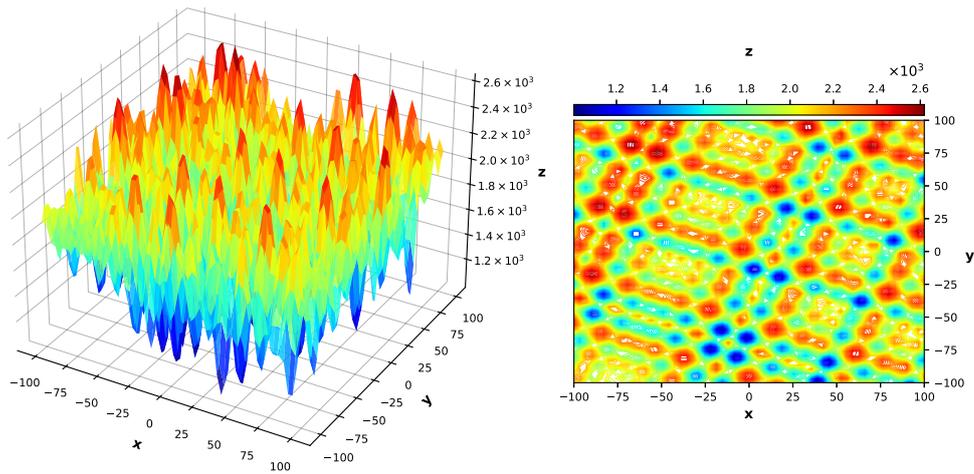
F7



F8



F9

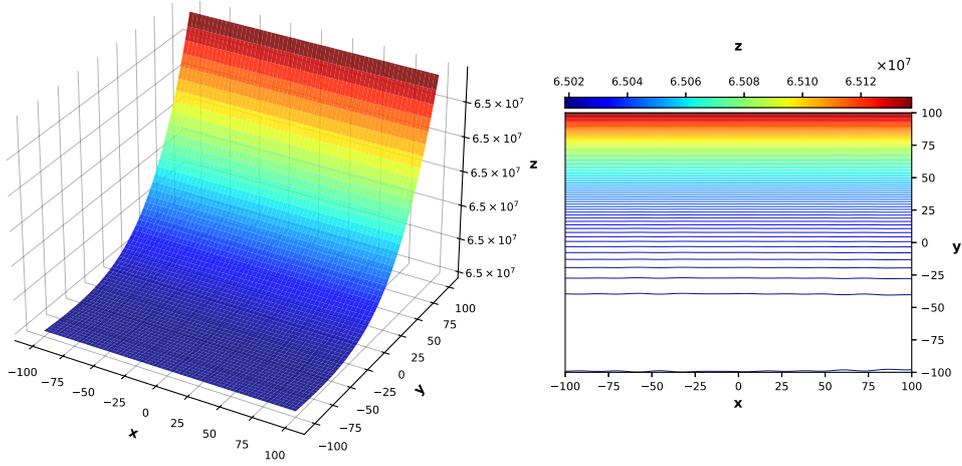


F10

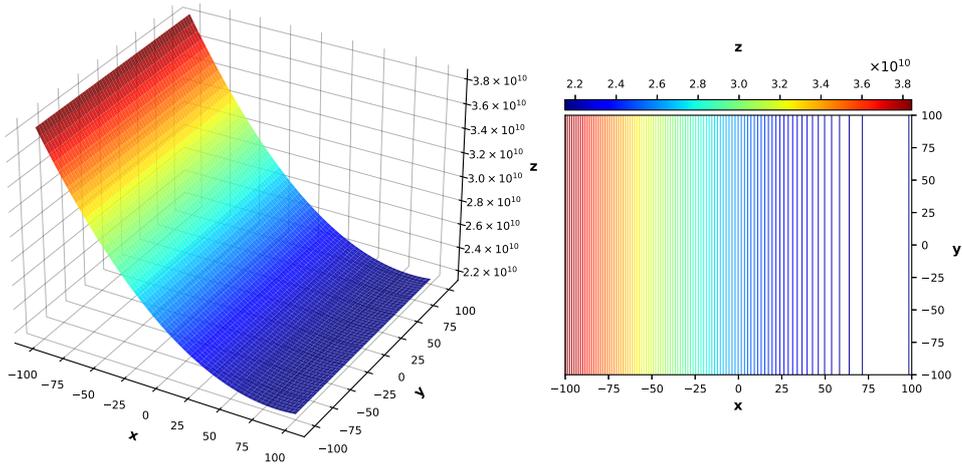
Figura A.3.1: Funciones de prueba multimodales y sus respectivas curvas de nivel ($D = 2$).

A continuación se muestran las gráficas de las funciones híbridas y compuestas, así como sus respectivas curvas de nivel. La definición detallada de cada una de estas funciones se encuentra en [117].

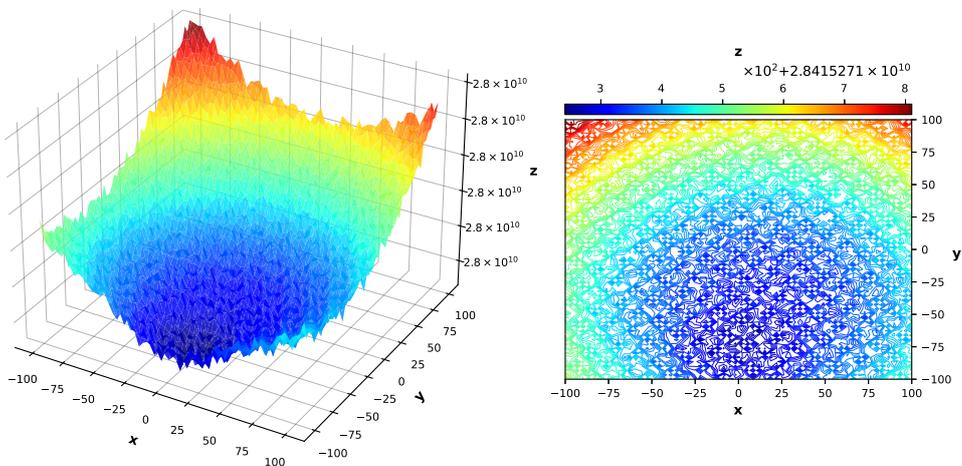
A.4. Funciones híbridas



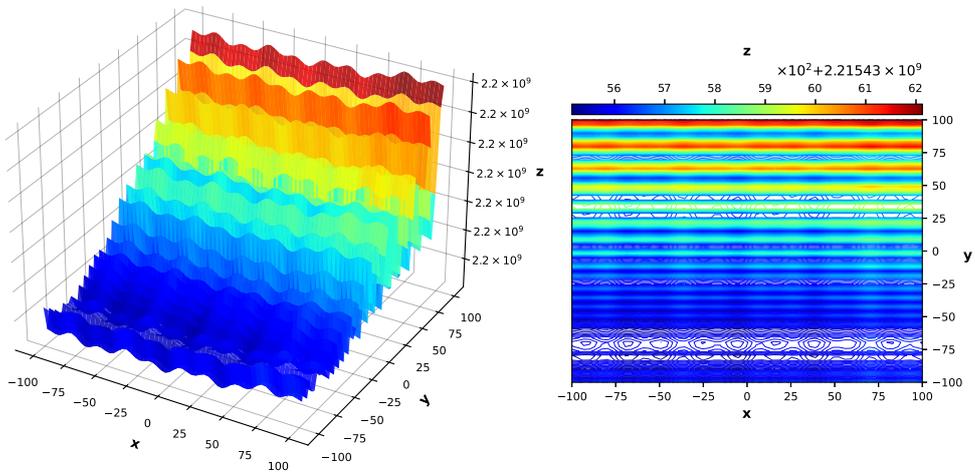
F11



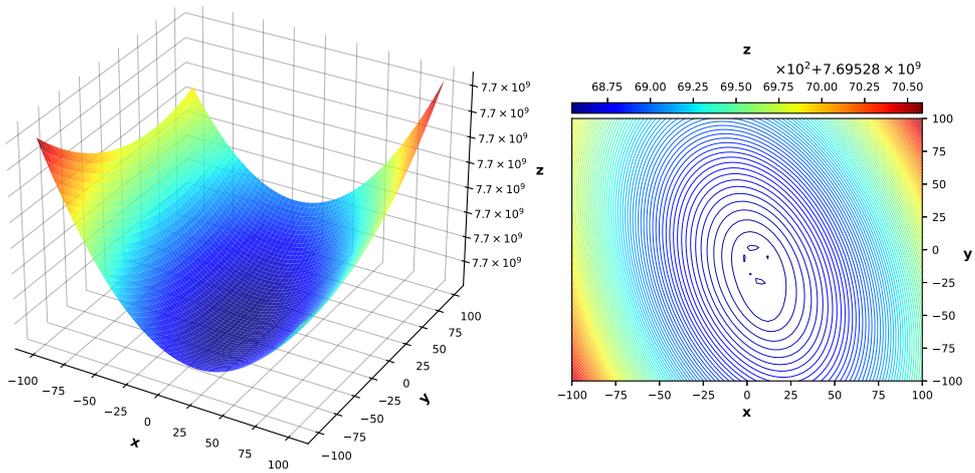
F12



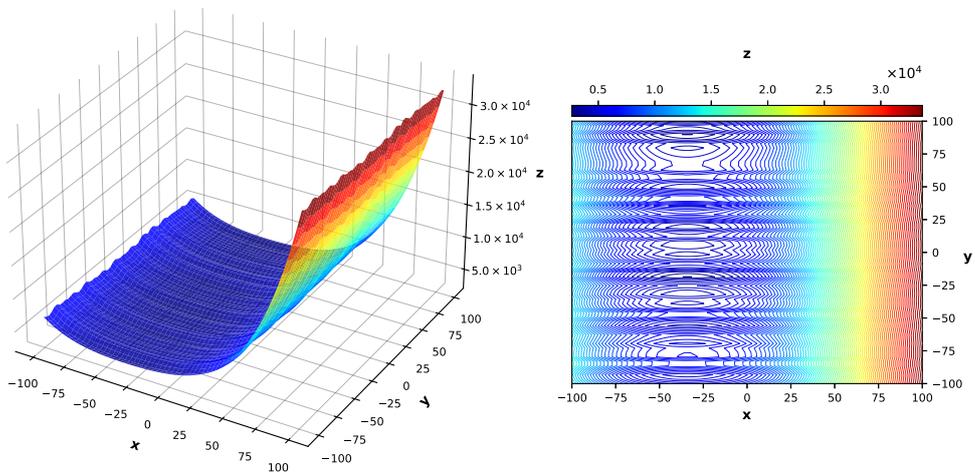
F13



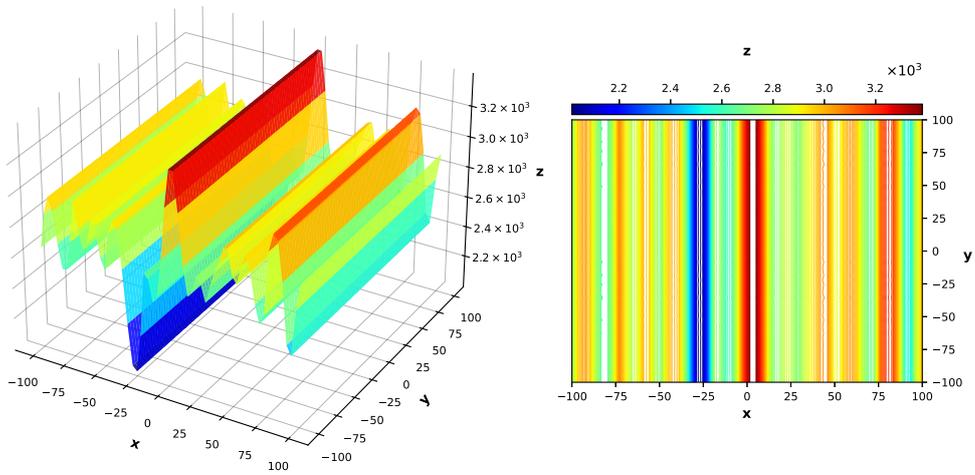
F14



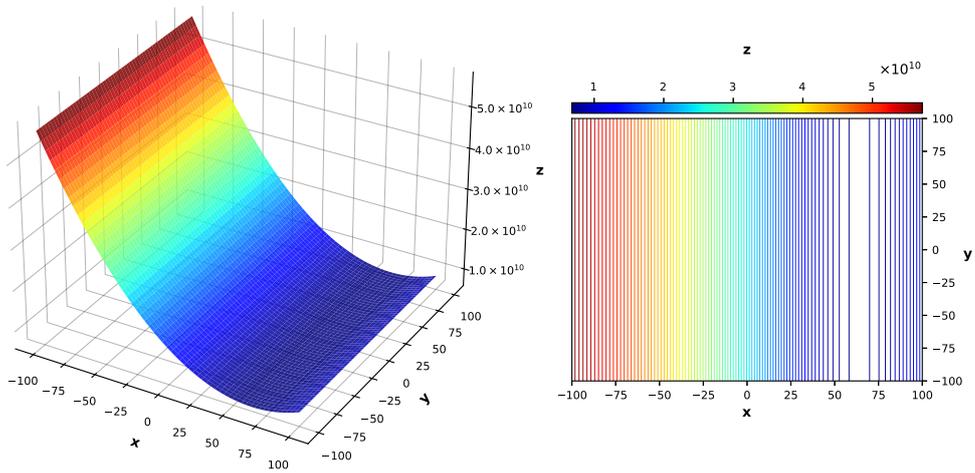
F15



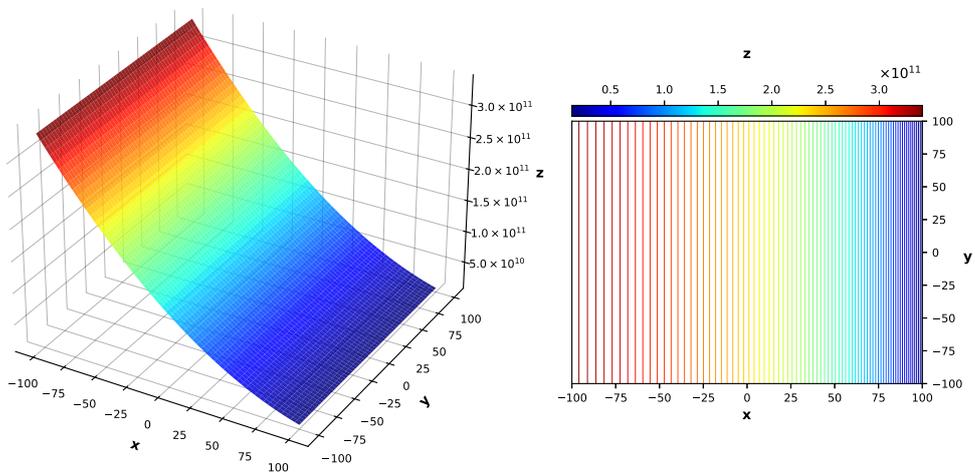
F16



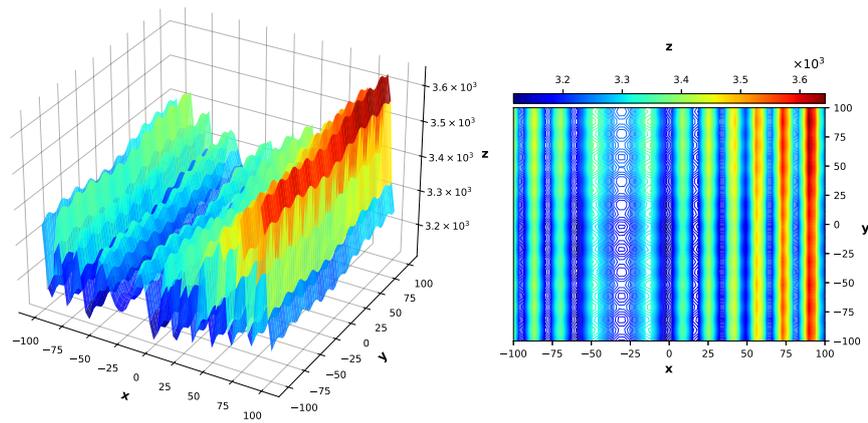
F17



F18



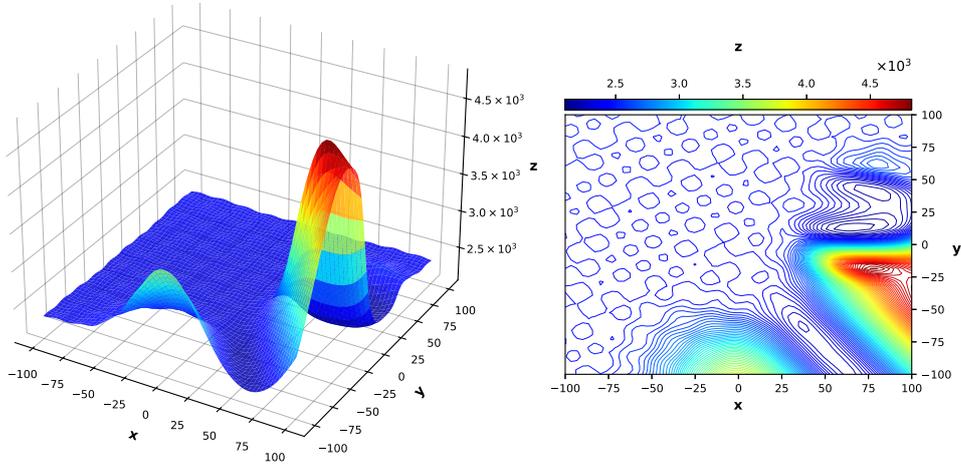
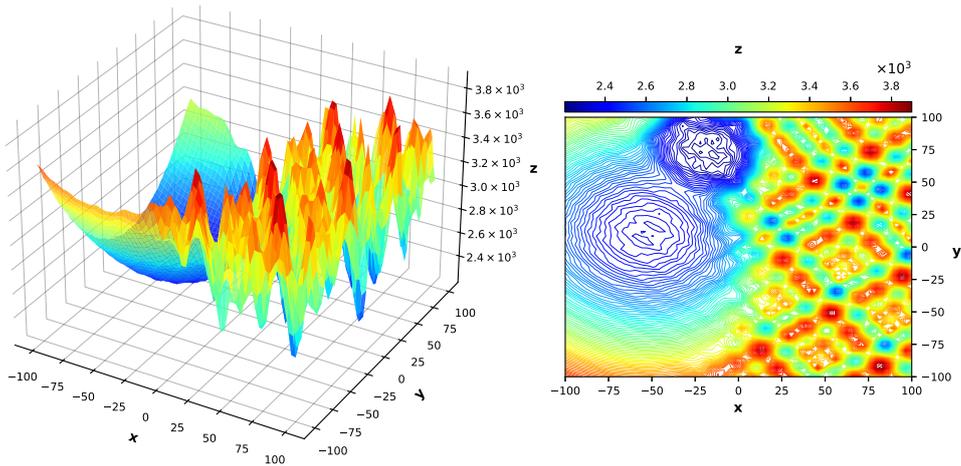
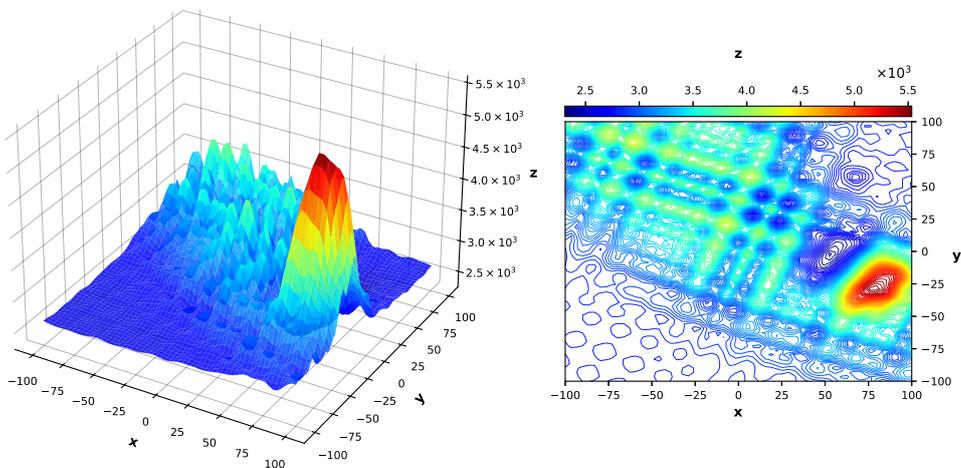
F19

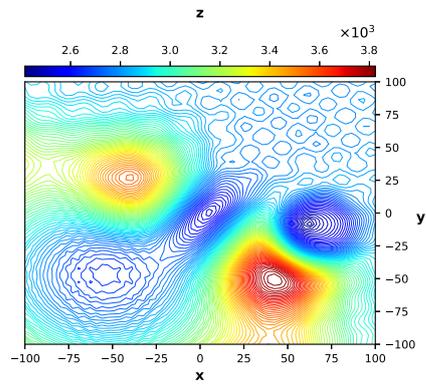
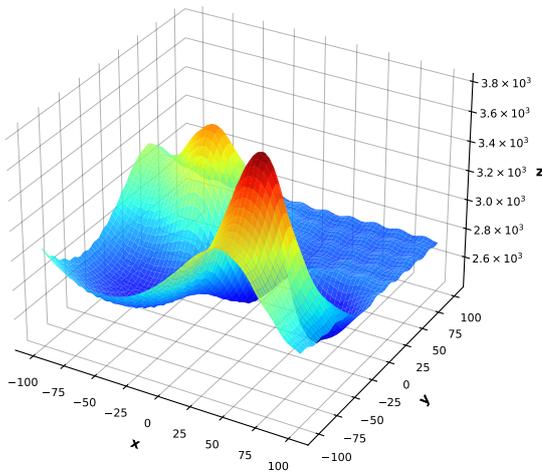


F_{20}

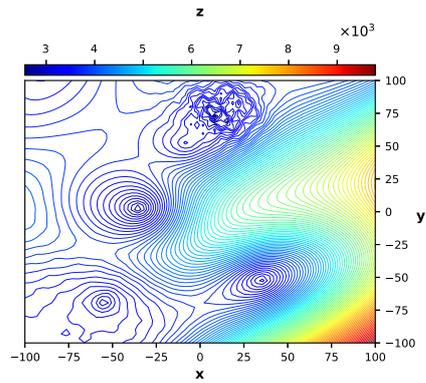
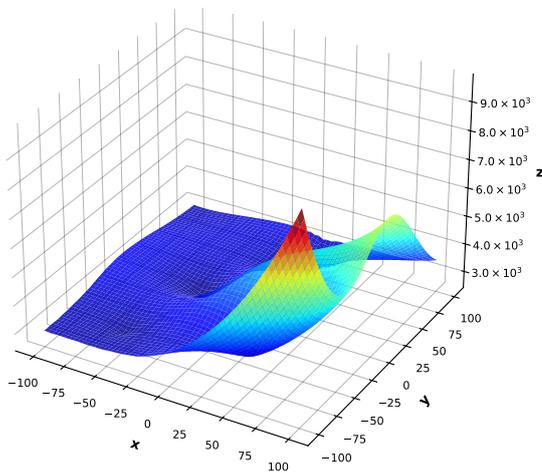
Figura A.4.1: Funciones de prueba híbridas y sus respectivas curvas de nivel ($D = 2$).

A.5. Funciones de composición

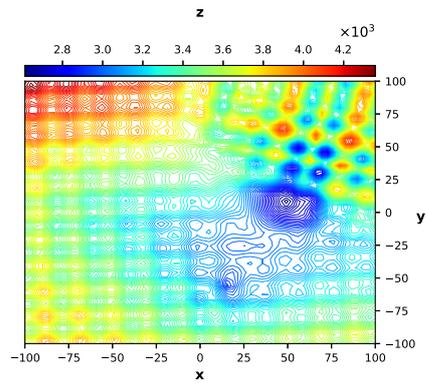
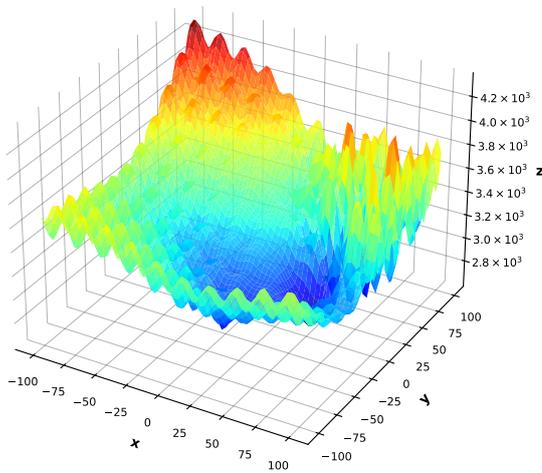
 F_{21}  F_{22}  F_{23}



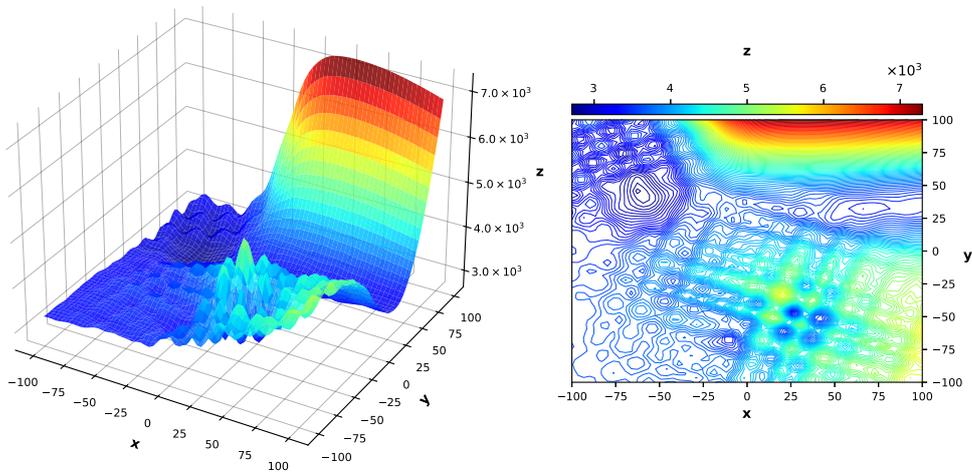
F24



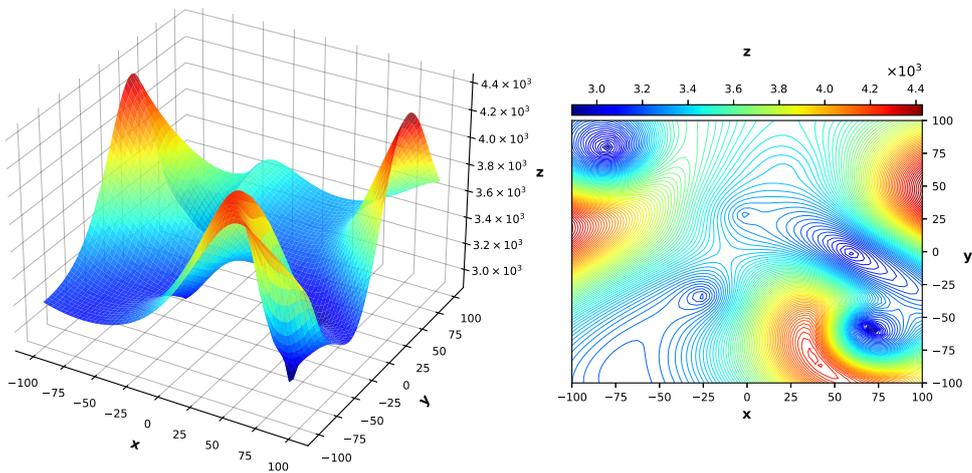
F25



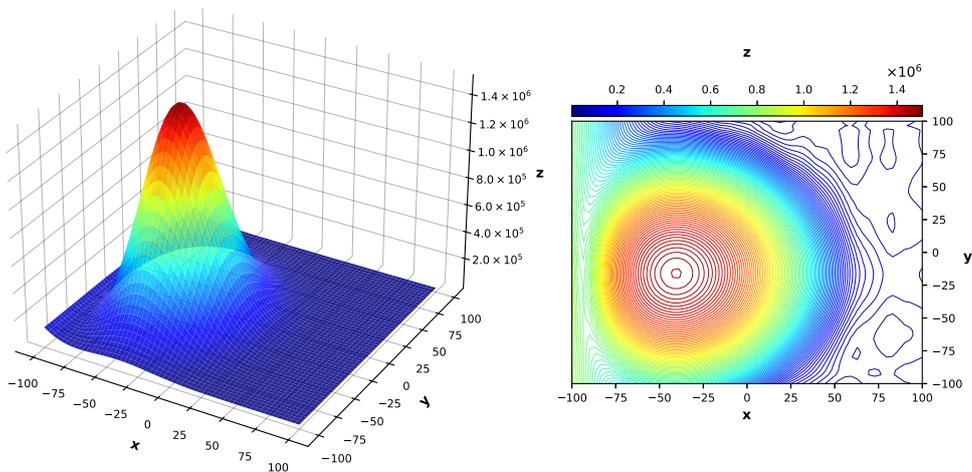
F26



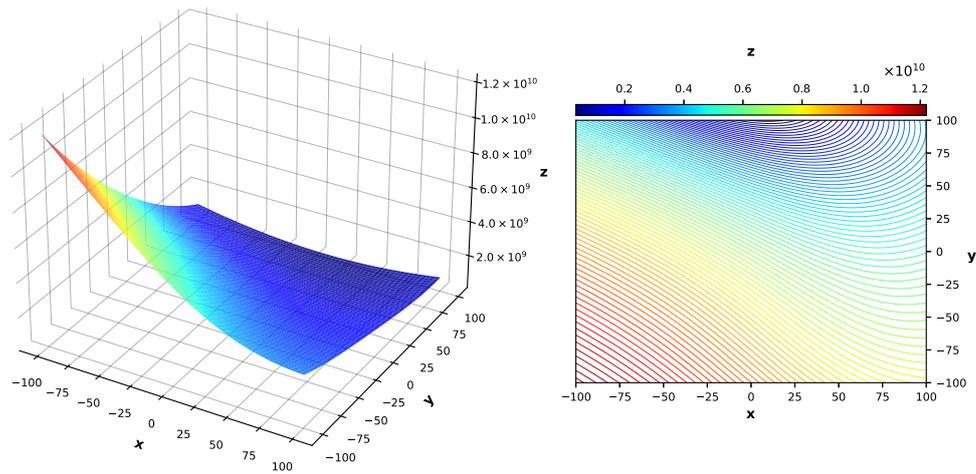
F27



F28



F29



F_{30}

Figura A.5.1: Funciones de prueba compuestas y sus respectivas curvas de nivel ($D = 2$).

Apéndice B

Curvas de convergencia para el conjunto de funciones de prueba CEC 2017

En este apéndice, se presentan las curvas de convergencia de la medida de error de solución para las 29 funciones de prueba del conjunto CEC 2017 en 10 y 20 dimensiones, obtenidas por los algoritmos HE-DEPSO, DEPSO, PSO, DE, CoDE y SHADE, como se describe en el Capítulo 4. En todas las gráficas, el eje horizontal representa las iteraciones y el eje vertical representa los valores medios de error para 31 repeticiones independientes.

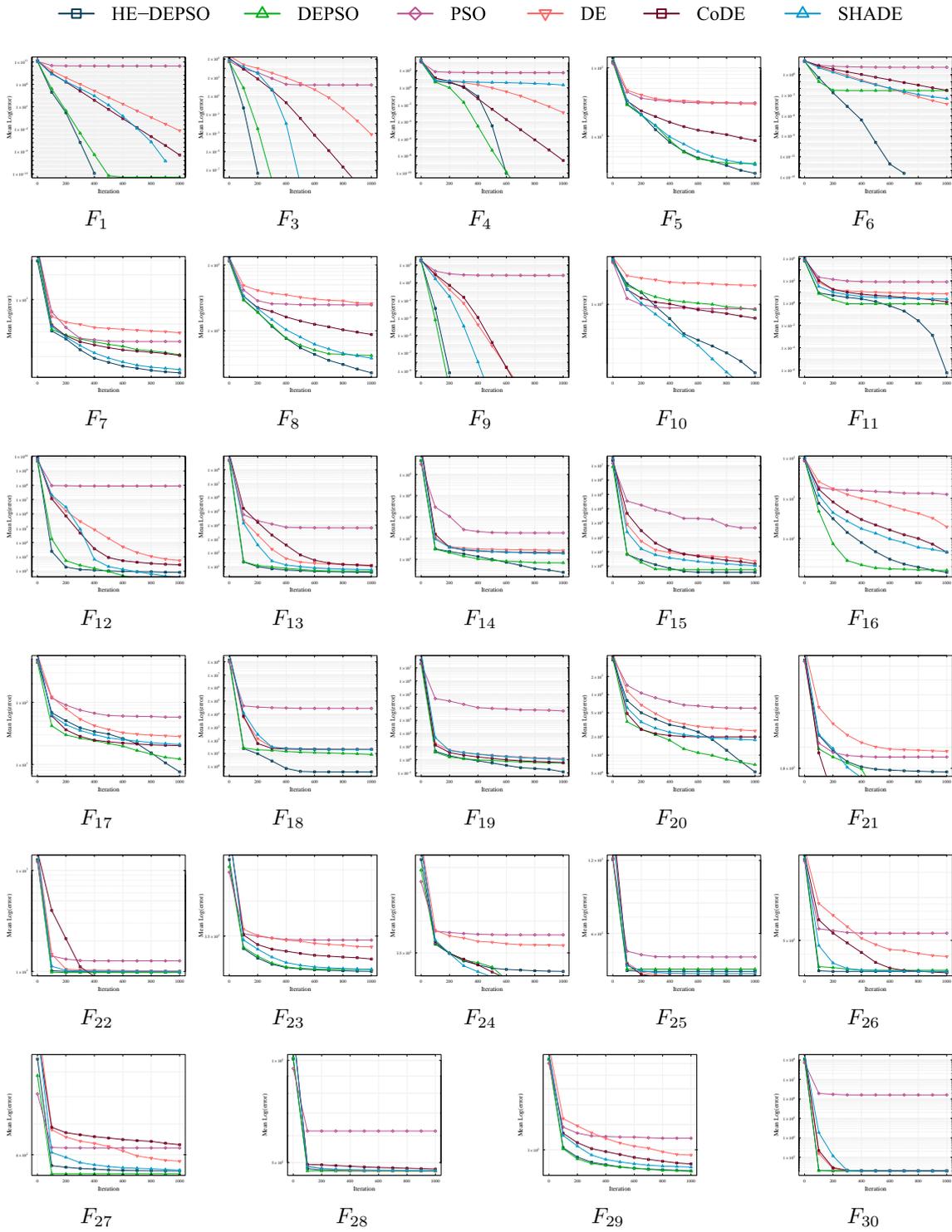


Figura B.1: Curvas de convergencia de la medida de error de solución para el conjunto CEC 2017 con $(D = 10)$. El eje horizontal y vertical representan la iteraciones y los valores de media de error para las 31 repeticiones independientes.

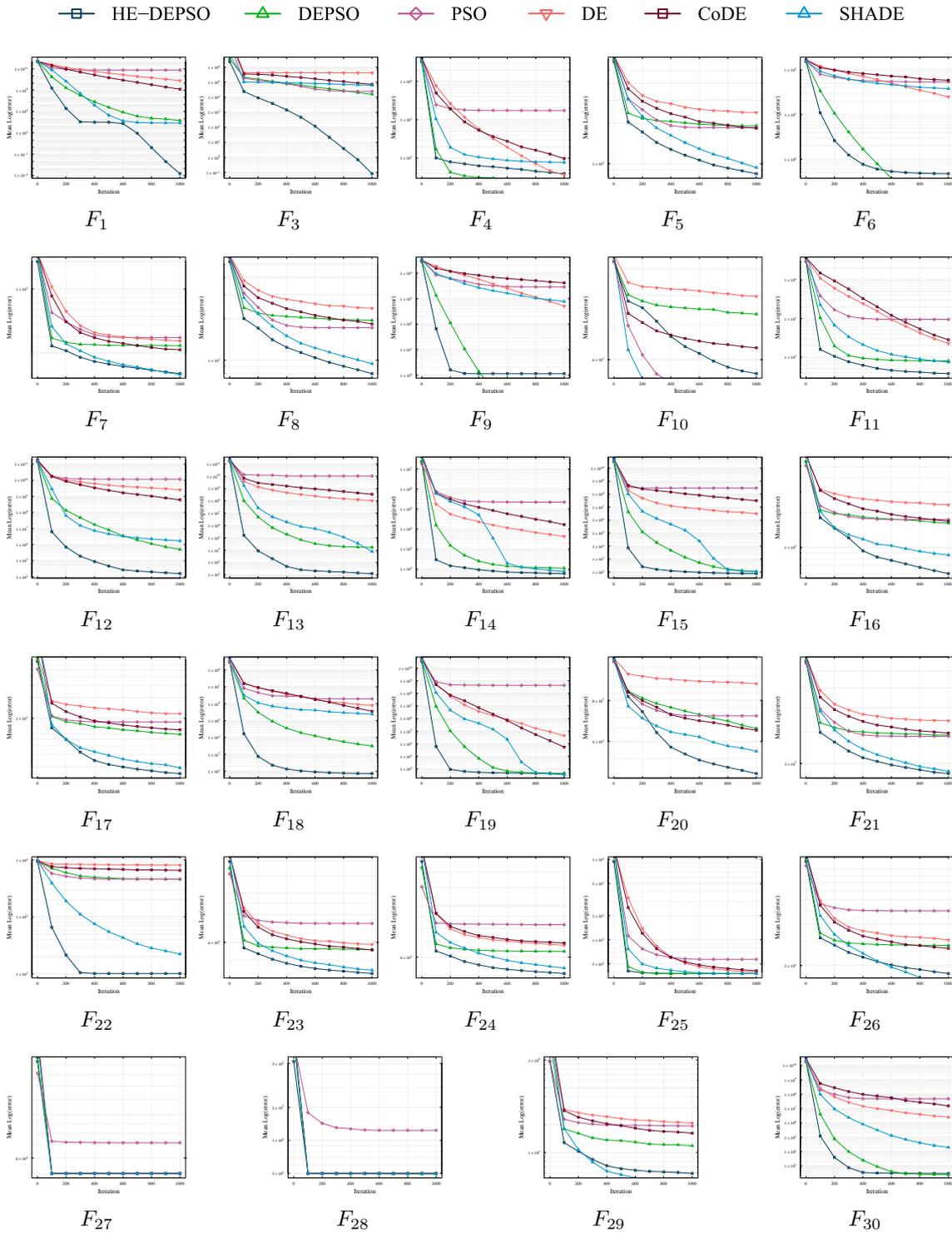


Figura B.2: Curvas de convergencia de la medida de error de solución para el conjunto CEC 2017 con ($D = 30$). El eje horizontal y vertical representan la iteraciones y los valores de media de error para las 31 repeticiones independientes.

Apéndice C

Método de ajuste χ^2

En este apéndice se presenta de forma breve el método de χ^2 , el cual permite vincular modelos teóricos con datos experimentales. Este método estadístico es ampliamente utilizado en diversas áreas de la ciencia, como física, química, biología y astronomía, para evaluar la bondad de ajuste entre un modelo teórico y los datos observados experimentalmente.

Considérese la siguiente situación común dentro de la física. Dado un conjunto de m observables O_i en la forma:

$$O_i = \bar{O}_i \pm \sigma_i, \quad (\text{C.1})$$

donde O_i son los valores centrales y σ_i su errores correspondientes.

Con el objetivo de explicar los hallazgos experimentales y observacionales, se propone un modelo teórico que provee predicciones $P_i(x)$ para las observables O_i de interés, donde:

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad (\text{C.2})$$

constituye el vector que alberga los n parámetros libres del modelo.

Una cuestión evidente en este punto es la evaluación de la concordancia del modelo con los datos experimentales. En el contexto del método de χ^2 , se utiliza una función específica, llamada función χ^2 , para cuantificar qué tan bien se ajusta el modelo con parámetros x a las observables $O_i = \bar{O}_i \pm \sigma_i$. Esta función toma la siguiente forma:

$$\chi^2(x) \equiv \sum_{i=1}^n \left(\frac{P_i(x) - \bar{O}_i}{\sigma_i} \right)^2, \quad (\text{C.3})$$

donde el vector x contiene los parámetros del modelo, $P_i(x)$ es la predicción del modelo para la observable O_i y \bar{O}_i el valor central para O_i . σ_i representa el error de O_i .

Es evidente que la función χ^2 no es negativa, es decir, $\chi^2 \geq 0$, y que su mínimo global puede ser cero si y solo si:

$$\exists_{x_{\min}} : P_i(x_{\min}) = \bar{O}_i \quad \forall_i. \quad (\text{C.4})$$

Con lo cual, para un modelo dado el mínimo global:

$$\chi^2 \equiv \min_x \chi^2(x) = \chi^2(x_{\min}) \quad (\text{C.5})$$

de la función χ^2 es una medida de la concordancia o acuerdo entre las predicciones del modelo y los datos. Cuanto menor sea el valor de χ^2 , mejor será el ajuste del modelo a los datos. Por lo tanto, la función χ^2 actúa como una función de mérito, permitiendo evaluar la calidad del ajuste del modelo y, en última instancia, determinar si el modelo es adecuado para describir los datos.

Es crucial destacar que el mínimo global x_{\min} no es necesariamente único. Incluso si fuera único, se debe considerar la presencia de varios mínimos locales x con χ^2 cerca del mínimo global χ^2 .

En resumen, el método de χ^2 es una herramienta estadística fundamental para vincular modelos teóricos con datos experimentales de manera rigurosa y cuantitativa, permitiendo evaluar la validez de las teorías y modelos propuestos en diversas áreas de la ciencia.

Apéndice D

Rangos de valores de los parámetros A_u y A_d

El espacio de búsqueda para esta función objetivo χ^2 está delimitado por los valores que pueden tomar los parámetros A_u , A_d , ϕ_1 y ϕ_2 .

Para determinar el rango de valores de los parámetros A_u y A_d , se deben considerar la condición $A_q > 0$, las ecuaciones Ecuaciones (5.1.11) a (5.1.12), las relaciones de las Ecuaciones (5.1.14) a (5.1.15), y las jerarquías de masas ($m_u < m_c < m_t$) y ($m_d < m_s < m_b$), de tal forma que se asegure que los elementos de las matrices O_u y O_d sean reales.

A partir de las Ecuaciones (5.1.11) a (5.1.12), el parámetro A_u o A_d debe cumplir con ser mayor a cero. Teniendo esto en cuenta y considerando el caso particular de A_u (aplica lo mismo para A_d), de la Ecuación (5.1.12) se tiene:

$$\frac{A_u - (-\eta_u m_u)(A_u - \eta_u m_c)(m_t - A_u)}{A_u} > 0 \quad (\text{D.1})$$

de donde se obtienen las siguientes posibilidades:

■ Posibilidad I:

$$(m_t - A_u) > 0 \wedge (A_u + \eta_u m_u)(A_u - \eta_u m_c) > 0 \quad (\text{D.2})$$

■ Posibilidad II:

$$(m_t - A_u) < 0 \wedge (A_u + \eta_u m_u)(A_u - \eta_u m_c) < 0 \quad (\text{D.3})$$

Con el propósito de mostrar el procedimiento de la obtención de los rangos para el parámetro A_u , se da continuidad a la Posibilidad I.

De acuerdo a la Expresión (D.2), para el parámetro A_u se tienen dos casos particulares, estos son:

■ Caso I:

$$(A_u + \eta_u m_u) > 0 \wedge (A_u - \eta_u m_c) > 0 \quad (\text{D.4})$$

■ Caso II:

$$(m_t - A_u) < 0 \wedge (A_u + \eta_u m_u)(A_u - \eta_u m_c) < 0 \quad (\text{D.5})$$

Teniendo en cuenta que η_u puede tomar los valores $\eta_u = +1$ y $\eta_u = -1$, al evaluar estos valores en las Expresión (D.4) y Expresión (D.5), se obtienen cuatro posibles intervalos para el parámetro A_u bajo la posibilidad I. Para determinar si un intervalo es viable, se debe verificar que se cumplan las condiciones $A_u > 0$ y la Ecuación (D.1). Los cuatro intervalos posibles para el parámetro A_u son:

$$A_u \in [m_c, m_t] \quad (\text{D.6})$$

$$A_u \in [-m_t, -m_u] \quad (\text{D.7})$$

$$A_u \in [m_u, m_t] \quad (\text{D.8})$$

$$A_u \in [-m_c, -m_t] \quad (\text{D.9})$$

Los intervalos relacionados con las Expresiones (D.6) y (D.7) se asocian con $\eta_u = +1$, mientras que los de las Expresiones (D.8) y (D.9) se relacionan con $\eta_u = -1$. Sin embargo, en los casos de la Expresión (D.7) y la Expresión (D.9), A_u sería negativo, lo que excluye directamente el uso de estos intervalos. Por otro lado, en la Expresión (D.6) y la Expresión (D.8), A_u no tendría valores negativos, por lo que se evalúan en la Ecuación (D.1) según el valor de η_u correspondiente. Al hacer esto, se observa que los intervalos de las Expresiones (D.6) y (D.8) satisfacen las condiciones mencionadas anteriormente, por lo que se consideran para formar parte de casos de estudio independientes, vinculados a las cuatro posibles combi-

naciones de valores entre η_u y η_d , generando su propio espacio de búsqueda para la función objetivo χ^2 .

Al llevar a cabo este procedimiento con el Caso II, y de igual forma, con la Posibilidad II, se obtienen un total de cuatro casos de estudio en los cuales se definen distintos intervalos para los parámetros A_u y A_d . Estos casos de estudio son los siguientes:

- Caso de estudio I: $\eta_u = +1$ y $\eta_d = +1$.

$$\begin{aligned} A_u &\in [m_c, m_t] \\ A_d &\in [m_s, m_b] \end{aligned} \tag{D.10}$$

- Caso de estudio II: $\eta_u = +1$ y $\eta_d = -1$.

$$\begin{aligned} A_u &\in [m_c, m_t] \\ A_d &\in [m_d, m_b] \end{aligned} \tag{D.11}$$

- Caso de estudio III: $\eta_u = -1$ y $\eta_d = +1$.

$$\begin{aligned} A_u &\in [m_u, m_t] \\ A_d &\in [m_s, m_b] \end{aligned} \tag{D.12}$$

- Caso de estudio IV: $\eta_u = -1$ y $\eta_d = -1$.

$$\begin{aligned} A_u &\in [m_u, m_t] \\ A_d &\in [m_d, m_b] \end{aligned} \tag{D.13}$$

Como se mencionó anteriormente, cada caso de estudio establece intervalos totales para los parámetros A_u y A_d . Estos intervalos, junto con los de los parámetros ϕ_1 y ϕ_2 , definen cuatro espacios de búsqueda de valores continuos, los cuales se asocian a cada caso de estudio.

Apéndice E

Detalles en el criterio χ_{red}^2 utilizado

En la Sección 5.2, se estableció el uso del criterio de Chi-cuadrado reducido (χ_{red}^2), expresado mediante la Ecuación (5.2.4), para evaluar la compatibilidad del modelo teórico de matrices de masa con cuatro ceros de textura y los datos experimentales actuales. Sin embargo, esta forma de proponer el criterio puede generar cierta ambigüedad, ya que para satisfacer dicho criterio, existen diferentes posibilidades de valores con los cuales puede contribuir cada uno de los cuatro términos de ajuste que constituyen la función χ^2 (ver Ecuación (5.2.3)) establecida en esta investigación. Por ejemplo, se puede dar el caso en el cual el primer término de ajuste:

$$\frac{(|(V_{CKM})_{us}^{th}(x)| - |V_{us}|)^2}{\sigma_{V_{us}}^2}, \quad (E.1)$$

denotado por simplicidad como χ_{12}^2 , tenga un valor de $\chi_{23}^2 = 1.455424$, mientras que el segundo, tercer y cuarto término denotados respectivamente por χ_{13}^2 , χ_{23}^2 y χ_{JJ}^2 , tengan valores de $\chi_{13}^2 = 1.020717$, $\chi_{23}^2 = 0.011266$ y $\chi_{JJ}^2 = 0.441887$. Si bien, estos valores al ser sumados y divididos entre cuatro de acuerdo a las Ecuaciones (5.2.4) y (5.2.5) satisfacen el criterio de χ_{red}^2 propuesto (ver Ecuación (5.2.4)), se tiene que los términos χ_{23}^2 y χ_{13}^2 son mayores que 1 (contribuyendo más que los demás términos), lo que indica que puede existir una gran discrepancia entre la predicción del modelo teórico y el valor experimental relacionado a las observables físicas χ_{23}^2 y χ_{13}^2 , y por lo tanto no ajustarse adecuadamente.

Para abordar este inconveniente, en este apéndice se presenta un estudio que analiza los efectos de restringir simultáneamente la contribución individual de cada término de ajuste (χ_{12}^2 , χ_{13}^2 , χ_{23}^2 y χ_{JJ}^2) dentro de la construcción del criterio de χ_{red}^2 utilizado en la investigación. Se analizarán los valores obtenidos para estos términos a partir de las soluciones encontradas para los parámetros A_u , A_d , ϕ_1 y ϕ_2 (ver

Subsección 5.2.3.1). Se tendrá principal interés en aquellas soluciones en las cuales la contribución de cada término sea a lo sumo 1, con el propósito de garantizar que ninguno de ellos genere un ajuste insuficiente. Esto se puede interpretar como que cada término es restringido de la siguiente forma:

$$\chi_{12}^2 : 0 < \frac{(|(V_{CKM})_{us}^{th}(x)| - |V_{us}|)^2}{\sigma_{V_{us}}^2} < 1, \quad (E.2)$$

$$\chi_{13}^2 : 0 < \frac{(|(V_{CKM})_{ub}^{th}(x)| - |V_{ub}|)^2}{\sigma_{V_{ub}}^2} < 1, \quad (E.3)$$

$$\chi_{23}^2 : 0 < \frac{(|(V_{CKM})_{cb}^{th}(x)| - |V_{cb}|)^2}{\sigma_{V_{cb}}^2} < 1, \quad (E.4)$$

$$\chi_{JJ}^2 : 0 < \frac{(J^{th}(x) - J)^2}{\sigma_J^2} < 1. \quad (E.5)$$

Este análisis se realizó en todas las soluciones encontradas, por caso de estudio (ver Tabla 5.2.1) y por precisión de χ_{red}^2 requerida (ver Ecuaciones (5.2.7) a (5.2.9)).

En la Tabla E.1, se muestra el porcentaje de soluciones encontradas para los parámetros A_u , A_d , ϕ_1 y ϕ_2 por caso de estudio y precisión de χ_{red}^2 , en las cuales se cumplen simultáneamente las Ecuaciones (E.2) a (E.5). Se observa que, en todos los casos de estudio, las soluciones que cumplen con una precisión de $1 \times 10^{-1} < \chi_{red}^2 < 1$ son las únicas en las que se ve reducido el porcentaje de soluciones en las que las contribuciones de χ_{12}^2 , χ_{13}^2 , χ_{23}^2 y χ_{JJ}^2 satisfacen ser menores que 1. Esto implica que, del total de soluciones encontradas originalmente para esta precisión, se reducen en promedio hasta un 65.3%. El 34.7% de soluciones restantes correspondientes a esta precisión, junto con aquellas para las precisiones $1 \times 10^{-2} < \chi_{red}^2 < 1 \times 10^{-1}$ y $\chi_{red}^2 < 1 \times 10^{-2}$, se denominarán como soluciones válidas bajo una restricción de χ_{red}^2 por término.

Las gráficas de la Figura E.1, muestran las regiones permitidas para A_u/m_t vs A_d/m_b (Figura E.1(a)) y ϕ_1/π vs ϕ_2/π (Figura E.1(b)) en el caso de estudio I. Se distinguen las soluciones válidas bajo dos tipos de restricciones en el criterio χ_{red}^2 :

- Soluciones válidas bajo una restricción de χ_{red}^2 por término (puntos verdes):

Tabla E.1: Porcentaje de las soluciones encontradas para los parámetros A_u , A_d , ϕ_1 y ϕ_2 por caso de estudio y precisión de χ_{red}^2 , en las cuales cada una de las contribuciones de χ_{12}^2 , χ_{13}^2 , χ_{23}^2 y χ_{JJ}^2 satisfacen ser menores que 1.

Caso de estudio	Precisión		
	$\chi_{red}^2 < 1$	$\chi_{red}^2 < 1 \times 10^{-1}$	$\chi_{red}^2 < 1 \times 10^{-2}$
I	34.9 %	100 %	100 %
II	35.4 %	100 %	100 %
III	32.9 %	100 %	100 %
IV	35.3 %	100 %	100 %

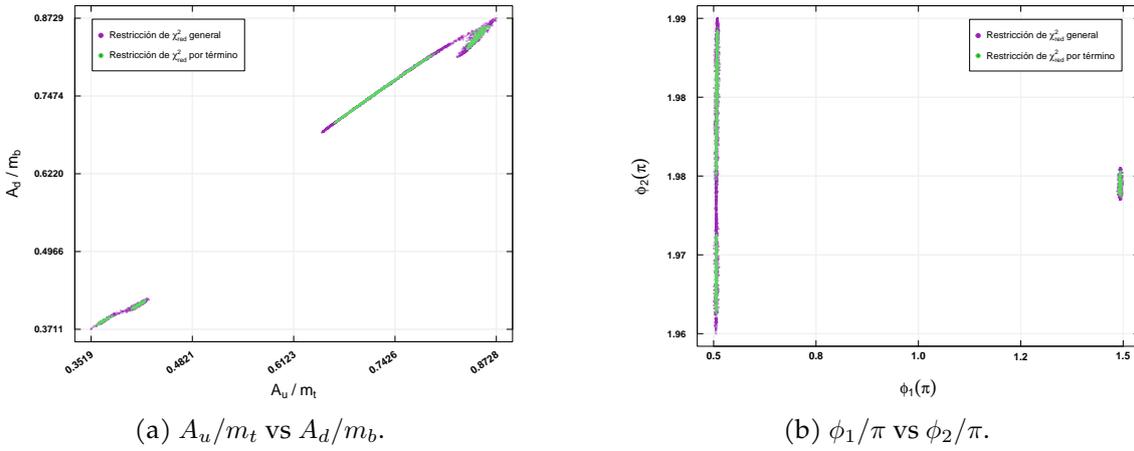


Figura E.1: Regiones permitidas para A_u/m_t , A_d/m_b , ϕ_1/π y ϕ_2/π en el caso de estudio I, mostrando soluciones válidas según restricciones de χ_{red}^2 por término (puntos verdes) y soluciones válidas según restricción general de χ_{red}^2 (puntos púrpura).

aquellas en las que todos los términos de χ^2 satisfacen simultáneamente las Ecuaciones (E.2) a (E.5).

- Soluciones válidas bajo una restricción de χ_{red}^2 general (puntos púrpura): aquellas que satisfacen la condición de la Ecuación (5.2.4) teniendo al menos un término de ajuste mayor que 1.

Al imponer restricciones en cada término de χ^2 , las regiones permitidas para A_u/m_t , A_d/m_b , ϕ_1/π y ϕ_2/π se ven reducidas, resultando en regiones más definidas. Estas observaciones se mantienen para los demás casos de estudio.

En la Figura E.2, siguiendo la estructura descrita en la Subsección 5.2.3.2, se grafican las regiones permitidas para A_u/m_t vs A_d/m_b y ϕ_1/π vs ϕ_2/π generadas con las soluciones válidas bajo una restricción de χ_{red}^2 por término, en el caso de estudio I. De forma general, se mantienen los resultados y observaciones realizadas en la

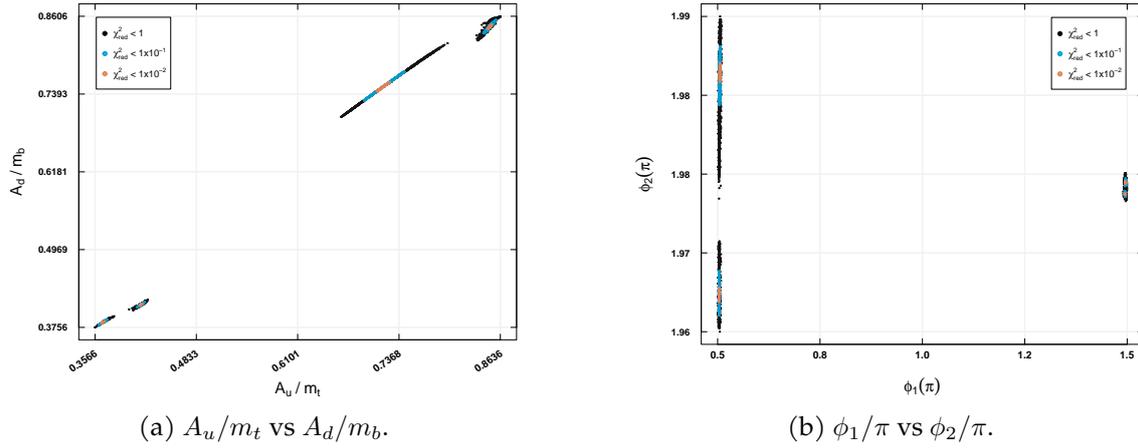


Figura E.2: Regiones permitidas para A_u/m_t , A_d/m_b , ϕ_1/π y ϕ_2/π en el caso de estudio I, restringidas según datos experimentales actuales con diferentes niveles de precisión ($\chi_{Red}^2 < 1$, $\chi_{Red}^2 < 1 \times 10^{-1}$ y $\chi_{Red}^2 < 1 \times 10^{-2}$), generadas utilizando las soluciones válidas bajo una restricción de χ_{red}^2 por término.

Subsección 5.2.3.2. Sin embargo, al restringir la contribución de cada término de la función χ^2 , se evidencia más claramente la presencia de cuatro regiones permitidas para los cuatro parámetros libres del modelo de texturas con cuatro ceros.

De acuerdo al esquema presentado en la Subsección 5.2.3.2, en la Figura E.3 se grafican las predicciones para los elementos de la matriz V_{CKM} : $|V_{cd}|$, $|V_{ud}|$, $|V_{cs}|$, $|V_{tb}|$, $|V_{td}|$ y $|V_{ts}|$, para el caso de estudio I, generadas con las soluciones válidas bajo una restricción de χ_{red}^2 por término. Estas gráficas muestran que las predicciones del modelo de texturas con cuatro ceros se ajustan mejor a los valores experimentales, reduciendo el número de puntos que sobrepasan los límites de error. Sin embargo, el modelo no logra predecir el elemento $|V_{td}|$ dentro de los límites experimentales con una precisión de $1 \times 10^{-2} < \chi_{red}^2 < 1 \times 10^{-1}$, como sí lo hace con el resto de elementos en todos los casos de estudio, limitando su alcance predictivo a soluciones con $\chi_{red}^2 < 1 \times 10^{-2}$.

Con lo discutido en este estudio, se puede establecer que restringir la contribución de los términos de ajuste en la función χ^2 utilizada para comparar el modelo teórico de la textura con cuatro ceros y los datos experimentales, evita incluir soluciones que no representan un buen ajuste a ciertas observables físicas. A pesar de las implicaciones de esta restricción, los resultados y observaciones del Capítulo 5 se mantienen, ya que es posible llegar a las mismas conclusiones. Además, la presencia más clara de cuatro regiones permitidas para los parámetros: A_u , A_d ,

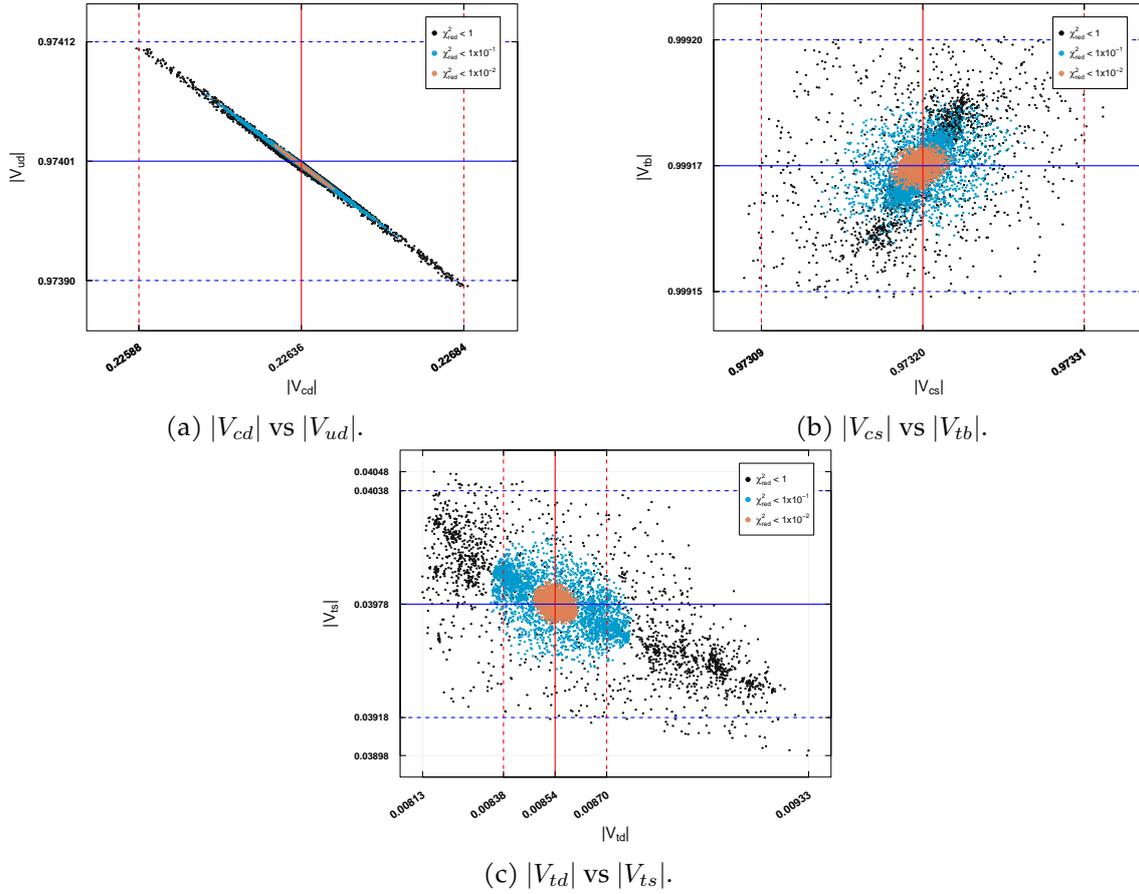


Figura E.3: Predicciones para los elementos $|V_{cd}|$, $|V_{ud}|$, $|V_{cs}|$, $|V_{tb}|$, $|V_{td}|$ y $|V_{ts}|$ en el caso de estudio I, generadas utilizando las soluciones válidas bajo una restricción de χ_{red}^2 por término.

ϕ_1 y ϕ_2 , identificadas mediante soluciones válidas bajo una restricción de χ_{red}^2 por término, puede dar pie a un análisis más profundo de sus implicaciones fenomenológicas dentro del formalismo de texturas.