



Técnicas de Computación Flexible aplicadas en la Óptica y la Ingeniería



Doctorado en Ciencias (Óptica)

Asesor: Dr. Francisco Javier Cuevas de la Rosa

Estudiante: M. en I. Carlos Hugo García Capulín

“Versión definitiva. Incluye cambios sugeridos por revisores”

Diciembre de 2014
León, Guanajuato, México

CENTRO DE INVESTIGACIONES
EN ÓPTICA A. C.

*Técnicas de Computación Flexible
aplicadas en la Óptica y la
Ingeniería*

Tesis Profesional que para obtener el grado de
Doctor en Ciencias (Óptica) presenta:

M. I. Carlos Hugo García Capulín

Asesor:

Dr. Francisco Javier Cuevas De la Rosa

León, Gto., Diciembre 2014

Agradecimientos

Quiero agradecer al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo brindado a través de la beca otorgada a mi persona bajo el número de registro 41635.



Al Centro de Investigaciones en Óptica, A. C. (CIO), que puso a mi disposición todos los recursos y medios necesarios para realizar mis estudios de doctorado.



Agradecimientos Personales

A mi familia:

Patricia, Dafne, Carlos y Martín

A mis maestros:

Dr. Raúl Sánchez, Ing. Fausto Ponce, Ing. León Hernández.

A mi asesor:

Dr. Francisco Javier Cuevas de la Rosa.

A Todos Gracias

Carlos García.

Índice general

1. Introducción	1
1.1. Antecedentes	1
1.2. Objetivo	6
1.3. Organización del trabajo de tesis	7
2. Métodos	8
2.1. Algoritmos Genéticos	8
2.1.1. Introducción	8
2.1.2. El algoritmo genético simple	10
2.1.3. Codificación	12
2.1.4. Función de aptitud	13
2.1.5. Operadores genéticos	14
2.1.6. Parámetros de control del Algoritmo Genético	16
2.1.7. Algoritmos genéticos jerárquicos	16
2.2. Base de funciones	17
2.2.1. Funciones B-splines	17
2.2.2. Superficies de producto tensorial de B-splines	18
2.2.3. Funciones de Base Radial	19
3. Algoritmos Genéticos aplicados en Óptica e Ingeniería	22
3.1. Ajuste de curvas B-spline	22
3.1.1. Definición del problema del ajuste de curvas	22
3.1.2. Ajuste de curvas B-spline usando un AGJ	23
3.1.3. Pruebas y resultados del ajuste de curvas	27
3.2. Aproximación de superficies de producto tensorial de B-splines	36
3.2.1. Definición del problema de aproximación de superficies	36
3.2.2. Aproximación de superficies mediante un AGJ	36

<i>ÍNDICE GENERAL</i>	IV
3.2.3. Pruebas y resultados de la aproximación de superficies	42
3.3. Demodulación de patrones de franjas	49
3.3.1. Definición del problema	49
3.3.2. Demodulación de patrones de franjas usando funciones de base radial y un AG	49
3.3.3. Pruebas y resultados de la demodulación de patrones de franjas	51
4. Conclusiones y Perspectivas	56
4.1. Conclusiones	56
4.2. Perspectivas	58
Información de contacto	60

Índice de figuras

2.1. Algoritmo genético simple.	11
2.2. Codificación binaria	12
2.3. Codificación real	13
3.1. Estructura General del cromosoma.	24
3.2. Diagramas de caja de los valores de MSE para ambos métodos.	30
3.3. Curvas ajustadas correspondientes la 50 ^a mejor estimación para la configuración experimental con 50 muestras y SNR=3. Las columnas corresponden de izquierda a derecha a los datos de entrada, resultados BARS y AGJ respectivamente (línea sólida: función estimada; línea interrumpida: función real). Las filas corresponden de arriba hacia abajo a las funciones de prueba 1 a 5.	35
3.4. Estructura general del cromosoma bidimensional	37
3.5. Variación de la probabilidad de mutación. Cada línea muestra el promedio de 100 conjuntos de datos de prueba.	40
3.6. Variación del tamaño de población. Cada línea muestra el promedio de 100 conjuntos de datos de prueba.	41
3.7. Evolución del mejor individuo para las 50 mejores pruebas.	42
3.8. Gráficas de las funciones de prueba.	43
3.9. Diagramas de caja de los valores de MSE obtenidos por los métodos probados.	45
3.10. Resultados para el AGJ. Las columnas corresponden de izquierda a derecha a: superficie ruidosa de entrada, nube de datos de entrada, superficies aproximadas. Las filas corresponden de arriba a abajo a las funciones de prueba de 1 a 4.	48
3.11. Patrones de franjas simulados.	53
3.12. Patrones de franjas de entrada y superficies de error del resultado	54

3.13. Patrones de franjas de entrada y superficies de error del resultado 55

Índice de cuadros

2.1. Funciones de base radial. (para $r = x - c_j $)	20
3.1. Funciones de prueba	28
3.2. Parámetros de control del AGJ	29
3.3. Media con error estándar de los valores de MSE obtenidos sobre 100 conjuntos de datos de 50 muestras.	33
3.4. Media con error estándar de los valores de MSE obtenidos sobre 100 conjuntos de datos de 100 muestras.	33
3.5. Media con error estándar de los valores de MSE obtenidos sobre 100 conjuntos de datos de 200 muestras.	34
3.6. Tiempo promedio de ejecución del AGJ.	41
3.7. Funciones bi-variadas de prueba.	43
3.8. Parámetros de control sintonizados.	44
3.9. Media con error estándar de los valores de MSE obtenidos sobre 100 conjuntos de prueba.	47
3.10. Parámetros de control del AG.	51
3.11. Funciones utilizadas para la simulación de los patrones de franjas.	52
3.12. Resultados de la demodulación de patrones de franjas cerradas	52
3.13. Resultados obtenidos para varios niveles de ruido.	53

Capítulo 1

Introducción

En los últimos años, las técnicas de computación flexible se han aplicado con gran éxito en la solución de problemas complejos pertenecientes a diversas áreas de la ciencia. Con este tipo de técnicas se ha logrado proponer nuevas soluciones a problemas típicos, así como también dar solución a otros problemas que no se pueden tratar mediante métodos tradicionales.

En este sentido dentro de esta investigación se pretende aplicar dichas técnicas en la solución de problemas complejos dentro del área de la óptica y la ingeniería. Dentro de los problemas que serán abordados en este trabajo se encuentran el ajuste de curvas y superficies a conjuntos de datos ruidosos, mediante el uso de una base de funciones polinomiales conocidas como B-spline. Además se extiende el uso del ajuste de superficies para aplicarlo en la recuperación de la fase a partir de imágenes con patrones de franjas cerradas.

Por otra parte, la exploración de la mejor codificación y configuración de las técnicas de computación flexible a utilizar es de suma importancia dentro de la investigación a fin de proponer soluciones novedosas a los problemas mencionados, las cuales obtengan excelentes resultados y sean competitivas en comparación con otros métodos existentes.

1.1. Antecedentes

Las técnicas de computación flexible tales como los Algoritmos Genéticos (AG), las Redes Neuronales Artificiales (RNA) y los Sistemas Difusos (SD), han sido aplicados en diferentes áreas de la ciencia y la ingeniería, presentándose como una buena alternativa para la solución a problemas com-

plejos en los que los métodos tradicionales no son aplicables o presentan soluciones acotadas [1].

Por ejemplo, una herramienta importante para el modelado geométrico, el análisis de datos, el procesamiento de imágenes, y muchas otras aplicaciones en ingeniería es el Ajuste de Curvas. En esta tarea, el objetivo es construir una curva que represente la mejor estimación de la relación funcional entre variables de entrada y salida dado un conjunto de datos de valores ruidosos. Entre algunos de los métodos más ampliamente usados para el ajuste de curvas se encuentran las técnicas de regresión. Dichas técnicas se pueden dividir en dos categorías: las técnicas paramétricas y las no paramétricas [2]. Las técnicas no paramétricas incluyen los métodos de suavizado con splines y de splines de regresión. Los métodos de suavizado con splines han sido utilizados por varios investigadores mostrando buenos resultados en el suavizado de datos ruidosos [3, 4, 5]. Su principal inconveniente es que ellos requieren la especificación de un parámetro global de suavidad [6]. Los métodos de splines de regresión asumen que la función puede ser aproximada por secciones continuas de polinomios (spline) de orden k con alguna secuencia de nodos. Los lugares donde las secciones de los polinomios se unen son conocidas como nodos y el orden corresponde a grado del polinomio menos uno. En estos métodos, el problema más importante es la selección del número y la posición de los nodos para poder estimar los coeficientes de la spline mediante mínimos cuadrados ordinarios. La selección del mejor modelo es realizada minimizando algún criterio estadístico de selección de modelos, tal como el criterio de información de Akaike, el criterio de información Bayesiano o el criterio generalizado de validación cruzada. Entonces el problema se convierte en un problema de optimización no lineal, multimodal y multivariable con muchos óptimos locales.

Muchos enfoques han sido propuestos para resolver este problema de optimización [7, 8, 9, 10], todos ellos basados en métodos de optimización no lineal; otros enfoques usan métodos Bayesianos en combinación con cadenas de Markov y el método de Mote Carlo [6, 11, 12, 13]. Estas técnicas, a diferencia de las anteriores, se enfocan en la forma en que se estiman los coeficientes y dejan en segundo plano la selección de los nodos [14]. Estos métodos han mostrado su éxito, pero son computacionalmente costosos y algunas veces resultan ser solo soluciones localmente óptimas [15, 16].

Debido a la complicada naturaleza del problema de optimización, varios investigadores han propuesto esquemas de búsqueda inteligente basados en algoritmos genéticos (AG) [17, 18]. Los AG, originalmente desarrollados por

Holland, han sido usados para optimizar funciones de aptitud imitando el proceso de evolución natural de los seres vivos. Los individuos de esta evolución son representaciones computacionales de soluciones potenciales para el problema. Cada individuo es representado por un código computacional conocido como cromosoma. El conjunto completo de individuos examinados en un tiempo dado es llamado población; ésta es evaluada con la función de aptitud para determinar cuáles individuos representan mejores soluciones para el problema. Los valores de aptitud son usados por el operador de selección para determinar la probabilidad de que un cromosoma sea utilizado para producir nuevos cromosomas. La producción de nuevos cromosomas a partir de los anteriores es realizada a través de la aplicación de los operadores genéticos, cruza y mutación. Una aplicación iterativa del esquema de selección y los operadores genéticos es repetida hasta alcanzar criterio de término deseado.

Típicamente, los enfoques basados en AG emplean un criterio de selección de modelos como función de aptitud para optimizar el número y la posición de los nodos. En [14, 19], los autores convierten el espacio de búsqueda en forma discreta, usando una codificación binaria para representar los nodos. En contraste, en [20, 21], una codificación real es usada para mejorar su espacio de búsqueda y desempeño de convergencia. Por otra lado, en [15], los nodos son representados por una codificación entera. En este último método, el número y la posición de los nodos es fija y solo su posición es optimizada. Finalmente, el mejor modelo es buscado para varias cantidades de nodos. En todos los enfoques basados en AG, la solución es desarrollada para cada iteración en forma de un sistema de ecuaciones lineales, de esta manera los coeficientes de las spline son calculados mediante el uso de mínimos cuadrados ordinarios.

Otra área de aplicación de estos métodos de ajuste y optimización se tiene en la aproximación de superficies. Típicamente en aplicaciones de ingeniería las técnicas de aproximación son aplicadas en problemas de ingeniería inversa, así como también para diseñar superficies que encajen con un conjunto de características deseadas [22]. En ingeniería inversa, el problema consiste con la generación automática de un modelo CAD a partir de un conjunto de puntos digitalizados pertenecientes a un objeto real.

En este contexto, el problema de la aproximación de superficies puede ser formulado de diferentes formas, obteniéndose resultados dependiendo de varias opciones. Las superficies B-spline pueden ser usadas para aproximar un conjunto arbitrario de datos, en este caso un sistema de ecuaciones lineales es

obtenido, el cual puede ser lineal respecto de los coeficientes pero es no lineal respecto de los nodos. Por lo tanto, la selección del número y la posición de los nodos de la spline es fundamental, así como también el método usado para resolver el sistema de ecuaciones. Ambas tareas son críticas pero además problemáticas, ya que se genera un problema complejo de optimización no lineal, multimodal y multivariable con muchos óptimos locales.

Para tratar con este problema, varios métodos se han propuesto en la literatura, tales como, el método de Shepard [23], el método del elemento finito [24, 25] y el método del producto tensor de splines [14, 19, 26, 27]. El método de Shepard, también conocido como el método original de interpolación de distancia inversa ponderada, trata con este problema a través de una función de interpolación continua obtenida a partir del promedio ponderado de los datos. Por otro lado, el método del elemento finito es un enfoque numérico para la solución de ecuaciones diferenciales. Este método propone el ajuste de una función continua por partes como solución, para posteriormente obtener los parámetros de la función de tal manera que se reduzca el error en la solución. Finalmente, el producto tensor de splines es otro método comúnmente usado para aproximar superficies. Es una generalización del ajuste de curvas con splines, cuyo objetivo es obtener funciones suaves a partir de puntos dispersos [28].

Al igual que para el ajuste de curvas en el caso de la aproximación de superficies también se han propuesto el uso de técnicas de computación flexible. En [29], los autores reconstruyen una superficie a partir de un conjunto de puntos dado, basados en un algoritmo de dos pasos usando una técnica de optimización por enjambre de partículas. En contraste, en [30] un AG es aplicado iterativamente para ajustar un conjunto de puntos de datos dado en dos pasos. Además, en los dos enfoques anteriores, una solución por mínimos cuadrados debe de realizarse para calcular los coeficientes de las splines [31, 32]. Con el fin de conseguir esta solución, a menudo es conveniente que la distribución de los nodos siga la regla dada por el teorema de Schoenberg-Whitney [33].

Por otra parte, en [19] los autores usan un AG para optimizar el número y la posición de los nodos con el objetivo de construir una curva B-spline que ajuste a un conjunto de puntos de datos. Una modificación de [19] es presentada en [21]. Aquí los autores usan un AG con codificación real que puede tratar datos que contengan discontinuidades. Otro ejemplo es presentado en [34]. Aquí los autores usan un algoritmo de luciérnagas para aproximar una curva B-spline a un conjunto de datos ruidosos. Aunque estos trabajos se re-

fieren a casos de curvas, estos se pueden extender para aproximar superficies mediante la aplicación repetida del método para cada una de las dimensiones de los datos para las direcciones de x e y separadamente, pero con un considerable tiempo de ejecución.

Dado que en este trabajo se consideran enfoques basados en splines, debemos resaltar el hecho de que el principal problema asociado con la aproximación de curvas y superficies usando splines es encontrar el mejor conjunto de nodos, donde el término “mejor” implica una adecuada selección del número y la posición de los nodos. Para llevar a cabo esta tarea, en [14], el autor ofrece un análisis sobre los principales algoritmos utilizados, los cuales están basados en los métodos de regresión con spline y su respectiva optimización.

En lo que respecta al área de la óptica podemos encontrar múltiples ejemplos de aplicaciones de las técnicas de computación flexible. En [35] se plantea el uso de una red neuronal multi-capa para llevar a cabo el proceso de calibración en perfilometría por proyección de franjas, en el cual no es necesario conocer los parámetros específicos de la configuración experimental usada. Similarmente en [36] una red neuronal de funciones de base radial es utilizada para estimar la información de profundidad de un objeto a partir de la información de fase recuperada usando la técnica de proyección de franjas.

En el campo de la metrología óptica [37], el análisis de interferogramas por medios digitales es de suma importancia, dado que nos permite obtener mediciones de diversas variables físicas con gran precisión [38]. Un interferograma es una imagen que contiene patrones de franjas generados por la interferencia de dos frentes de luz coherente. Dicha interferencia modula información que esta directamente relacionada con alguna variable física. Dentro de la gama de variables físicas que pueden ser medidas con este procedimiento denominado interferometría óptica se encuentran: mediciones de profundidad, análisis de esfuerzos [39], gradientes de temperatura, deformación de superficies [40, 41], topografía de superficies etc.

El objetivo de las técnicas de análisis de interferogramas es llevar a cabo el análisis automático de un interferograma obtenido a partir de un método de interferometría óptica, los cuales pueden aplicarse en eventos estáticos y dinámicos. El número, la forma y la distribución de las franjas de interferencia proporcionan datos muy útiles sobre la variable física relacionada. Para realizar dicho análisis existen diversas técnicas que están basadas en métodos matemáticos como la transformada de Fourier [42], filtros de cuadratura [43] o métodos de corrimiento de fase [44]. Por otra parte también se han aplicado técnicas de cómputo evolutivo con gran éxito [45, 46], sin embargo su

principal desventaja es la capacidad de cómputo necesaria para su ejecución.

Finalmente, un aspecto importante dentro de la técnica de demodulación de interferogramas mediante algoritmos genéticos son las técnicas de Aproximación de Superficies. Estas son importantes dado que son utilizadas por el algoritmo genético para generar las fases que son las posibles soluciones para el problema de la demodulación de los interferogramas.

1.2. Objetivo

El presente trabajo tiene como objetivo aplicar técnicas de computación flexible para la solución de problemas complejos relacionados al ajuste de superficies para la aproximación de la fase en aplicaciones de metrología óptica.

Dentro de las técnicas de computación flexible que utilizamos se encuentran los algoritmos evolutivos tales como lo son los algoritmos genéticos. Los cuales serán aplicados en problemas de aproximación de curvas y superficies obtenidas a partir de datos con ruido, así como también en la demodulación de imágenes de patrones de franjas cerradas.

Para el desarrollo de dichas aplicaciones se plantean los siguientes objetivos específicos:

- Desarrollar una librería genérica en un lenguaje de programación para la aplicación del algoritmo genético.
- Implementar los distintos métodos de codificación para los cromosomas del algoritmo genético.
- Desarrollar diversas funciones que implementen los múltiples métodos de selección.
- Desarrollar diversas funciones que implementen los múltiples operadores genéticos.
- Desarrollar la aplicación de ajuste de curvas utilizando B-spline y algoritmos genéticos.
- Desarrollar la aplicación de ajuste de superficies utilizando B-spline y algoritmos genéticos.

- Desarrollar la aplicación de demodulación de imágenes con patrones de franjas cerradas utilizando algoritmos genéticos.

1.3. Organización del trabajo de tesis

Esta tesis esta organizada de la siguiente manera. En el capítulo 2 se presentan los fundamentos de los AG y sus características más importantes como parte de la metodología utilizada en este trabajo. Además se aborda toda la base matemática acerca de las bases de funciones empleadas en el desarrollo de las aplicaciones, para continuar con el capítulo 3 donde se presenta el desarrollo de las aplicaciones de los AG en en el área de la óptica y la ingeniería, así como los resultados obtenidos; para terminar con el capítulo 4 donde se presentan las conclusiones generales del trabajo de investigación y las perspectivas del trabajo a futuro.

Capítulo 2

Métodos

2.1. Algoritmos Genéticos

2.1.1. Introducción

Un Algoritmo Genético (AG) es un método de búsqueda pseudo aleatorio basado en la teoría de evolución de las especies de Darwin [47]. Este tipo de algoritmo fue introducido por John Holland a principios de los años 70s. Desde su comienzo, estos algoritmos han tenido éxito para resolver problemas complejos con relativa facilidad y rapidez, por lo que la comunidad científica de diferentes áreas ha mostrado un enorme interés en los últimos años. Originalmente, esta técnica fue conocida como “técnica de planes reproductivos” pero se popularizó bajo el nombre de algoritmos genéticos tras la publicación del libro “Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence” escrito por Holland [17].

De manera general, podemos decir que la evolución de los seres vivos se realiza mediante un proceso de selección natural basado en los atributos del individuo para mantener su supervivencia. Esto es, que los individuos mejor adaptados, más fuertes y con mejores características son los que van desplazando a otros menos favorecidos por la naturaleza, de generación en generación. Esta selección se lleva a cabo mediante la reproducción, dado que los individuos mejor adaptados son los que tienen más posibilidades de dejar descendencia, pasando así información sobre sus características a la siguiente generación.

Las características de un individuo están codificadas en “*genes*”. Un “*gen*”

es una entidad funcional que codifica una característica específica de un individuo. Esta información genética está agrupada a su vez en cromosomas, que es donde se codifican las características del ser vivo. Dichas características almacenadas en el cromosoma varían de unas generaciones a otras mediante el intercambio genético que se da durante el proceso de reproducción. Dicho lo anterior podemos decir lo siguiente:

- La evolución es un proceso que opera directamente sobre los cromosomas y no sobre los organismos que representan.
- La selección natural es el proceso por el cual los cromosomas con buenas estructuras se reproducen más a menudo que los demás y generan una mayor cantidad de prole.
- El proceso evolutivo propiamente dicho se da durante la etapa de reproducción cuando se lleva a cabo el intercambio de material genético entre los progenitores.
- El proceso de evolución carece de memoria, es decir, que en el proceso de formación de los nuevos individuos solo se toma en cuenta el material genético de la generación anterior.

Una descripción amplia de un AG es la propuesta por John Koza [48]:

“ El AG transforma una población de individuos, cada uno con un valor de aptitud asociado, en una nueva generación de la población usando el principio Darwiniano de reproducción y supervivencia del más apto, ocurriendo naturalmente operaciones genéticas tales como la reproducción y mutación. Cada individuo en la población representa una posible solución a un problema dado. El AG busca la solución óptima o la mejor al problema, engendrando genéticamente la población de individuos.”

Los algoritmos genéticos agrupan las posibles soluciones de un problema como un grupo de individuos dentro de una población, codificando estas soluciones en estructuras de datos (p. ej. cadenas binarias en el AG simple) que representan los cromosomas, todos los elementos de la población se deben hacer evolucionar, para tal efecto cada individuo es evaluado por una función de “aptitud”, la cual nos proporciona una medida de qué tan buena es esta solución para nuestro problema. Esta función de aptitud es uno de los puntos más importantes dentro del AG, ya que de ella depende mucho el éxito o fracaso de la aproximación. En base a su nivel de aptitud, los individuos son

seleccionados para la reproducción teniendo más posibilidades aquellos con un mejor nivel, este proceso es realizado mediante la cruce. Además de este intercambio de material genético se pueden aplicar algunos otros operadores como el de mutación. Una vez realizado este proceso, los nuevos individuos se vuelven a evaluar para determinar su nuevo nivel de aptitud. Esta secuencia se repite hasta cumplir con un criterio de tolerancia que determine encontrar una aproximación adecuada al problema en cuestión. Este es otro punto importante en los algoritmos genéticos conocido como “convergencia del algoritmo”, aquí se deben de establecer los criterios adecuados para detener el proceso de evolución en un momento tal que los individuos se encuentren lo más próximos a buenas soluciones.

Los algoritmos genéticos se aplican en una amplia variedad de áreas como el diseño, música, administración de recursos, finanzas, biología, economía, aprendizaje etc. Generalmente en donde los problemas de alta complejidad se puedan enfocar y modelar como problemas de optimización [49, 50].

2.1.2. El algoritmo genético simple

Las técnicas de Computación Evolutiva (CE), en particular los algoritmos genéticos, son usados para optimizar una función de aptitud imitando la evolución natural de los organismos. Los individuos en este caso son representaciones computacionales de soluciones potenciales para resolver el problema. En un AG simple cada individuo es representado como una cadena binaria, también conocida como cromosoma computacional.

El conjunto de individuos examinados en cada iteración es llamado población. Dicha población es evaluada mediante una función de aptitud que es modelada de acuerdo al problema a resolver y que mide el desempeño de cada individuo, de esta manera los individuos con mejor aptitud son beneficiados en la evolución.

Para llevar a cabo la evolución de la población son utilizados diferentes operadores genéticos (Selección, Cruza y Mutación) los cuales se aplican sobre todos los individuos de la población actual para generar una nueva población. A la creación de poblaciones sucesivas se les llaman generaciones. Un algoritmo en donde la totalidad de la población es reemplazada entre generaciones se llama de reemplazo generacional. Un algoritmo donde solo una parte de la población es reemplazada y la otra parte pasa igual a la siguiente generación se llama un algoritmo de política de estado estable.

En la figura 2.1 se muestra el diagrama de flujo de un AG simple, en el

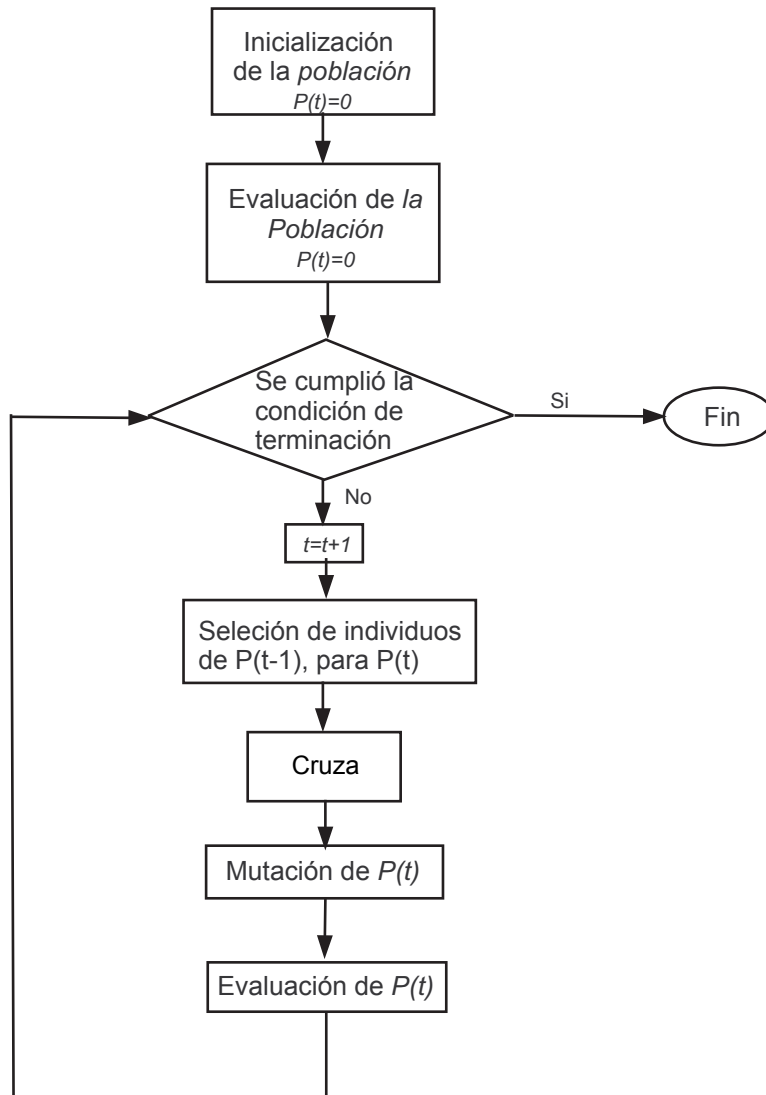


Figura 2.1: Algoritmo genético simple.

cual se incluyen los pasos de evaluación, selección, cruza y mutación. Esta es la estructura general del AG utilizado en el desarrollo de las aplicaciones de este trabajo de investigación.

2.1.3. Codificación

La codificación es el tipo y estructura de datos utilizados para representar los individuos de manera computacional dentro del AG, esta representación es agrupada en un elemento llamado cromosoma. Todos los símbolos utilizados en la representación de los individuos forman lo que se conoce como alfabeto genético. El tipo de alfabeto genético empleado depende totalmente del problema que se desea resolver. Entre los tipos de codificación más populares se encuentran la codificación binaria y la codificación real.

Codificación binaria

Este tipo de codificación tiene un gran poder de representación y se puede aplicar casi a cualquier problema. Los cromosomas se configuran mediante un alfabeto que consta de solo dos símbolos el 0 y el 1. En base a este alfabeto se pueden utilizar cadenas de estos símbolos para representar los parámetros que describen cualquier tipo de individuo, ya sean valores enteros o reales.

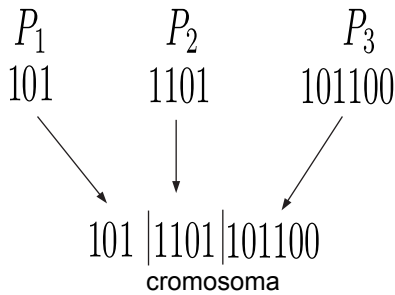


Figura 2.2: Codificación binaria

En la figura 2.2 se muestra un ejemplo de este tipo de codificación, donde se representan tres parámetros P_1 , P_2 y P_3 que componen un cromosoma que especifica a un individuo. Este tipo de codificación es utilizado en el desarrollo de las aplicaciones debido a que es adecuado para la representación del tipo de individuos que se tienen para el sistema, además de que facilita la acción de los operadores genéticos.

Codificación real

La codificación tipo real representa el cromosoma de un individuo mediante un vector de valores reales; donde cada uno de los valores representa uno de los parámetros que describen al individuo o la posible solución al problema de optimización.

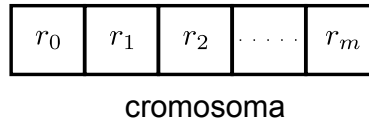


Figura 2.3: Codificación real

La estructura general de un cromosoma con codificación real se muestra en la figura 2.3, en la cual se aprecia un conjunto de m valores tipo real, cada uno de los cuales representa un gen del cromosoma.

2.1.4. Función de aptitud

El proceso de evaluación de la población se realiza mediante la función de aptitud, con el fin de determinar que individuos dentro de la población representan una mejor solución para el problema que se desea resolver. Cada individuo, representado por un cromosoma C_i , tiene asociado un valor de aptitud F_{C_i} , que se obtiene al evaluar la función de aptitud $F(C)$ en $C = C_i$.

En el AG, la función $F(C)$ es propiamente la función a optimizar. Durante la evaluación de $F(C_i)$, se decodifica el gen a lo cual se le llama “fenotipo”, convirtiéndose en una serie de parámetros del problema a tratar, se halla la solución del problema a partir de esos parámetros, y se le da una puntuación a esa solución. A esta puntuación se le llama aptitud (en inglés llamado *fitness*). El valor de aptitud determina siempre los cromosomas que se van a reproducir y aquéllos que se van a eliminar.

Una parte importante dentro de la evaluación de la función de aptitud es la función de transformación que se emplea para la decodificación del cromosoma, a fin de poder evaluar la aptitud de dicho individuo. Ésta transforma los genes del cromosoma completo en un grupo de parámetros que describen una posible solución del problema, con los cuales se evalúa realmente su aptitud.

2.1.5. Operadores genéticos

Para llevar a cabo la evolución de la población existen diversos operadores genéticos, generalmente se utilizan tres operadores, de entre los más comunes se encuentran los que se describen a continuación.

Selección

Este operador determina los individuos sobre los cuales se aplicaran los operadores de cruce y mutación. La selección esta basada en el nivel de aptitud del individuo, lo cual proporciona cierta dirección al proceso de búsqueda. La selección proporciona un método por el cual los individuos más aptos tienen más posibilidades de ser seleccionados a fin de generar individuos con mejores características que los progenitores, pero al mismo tiempo permite que los individuos menos adaptados puedan tener la posibilidad de reproducirse. Existen varios métodos de selección entre los más usados se encuentran los siguientes:

Selección por Ruleta. Este método esta basado en el juego de la ruleta. En este caso se tiene una ruleta seccionada en tantas partes como individuos existan en la población. El tamaño de cada una de estas partes es proporcional al valor de aptitud de cada individuo. Posteriormente se arroja la pelota para obtener la selección de un individuo. El número de individuos seleccionados dependerá del tipo de evolución, si se desea reemplazar a toda la población o si parte de ella pasará sin alteraciones.

Selección por Rango. La selección por rango es un método alternativo. En este caso, los individuos son ordenados de acuerdo a su valor de aptitud. Para este método el valor de aptitud no necesita ser escalado, lo cuál resulta muy útil cuando no se conoce el valor óptimo. Este método también tiene utilidad cuando la diferencia entre los valores de aptitud de los individuos es muy pequeña. En este caso se ordena la población de menor a mayor aptitud y se asigna de forma lineal el número de hijos de cada individuo.

Selección por Torneo. En este método los individuos compiten en torneos de n candidatos. Comúnmente los torneos se realizan para $n = 2$ individuos, los cuales son tomados al azar dentro de la población fijando un umbral ($U > 0,5$), siendo seleccionado para ser parte de la siguiente generación aquel que tenga mejor nivel de aptitud si el número aleatorio generado es mayor que el umbral establecido. Se deben realizar tantos torneos como sean necesarios para llenar por completo la nueva generación de individuos.

En los diferentes métodos de selección, puede usarse la idea de los individuos elite. Los individuos elite prevén la pérdida de la mejor solución en la población, por causa de la cruce y la mutación, realizando una copia del mejor individuo de forma directa a la siguiente generación de individuos.

Cruza

La cruce es un operador que trabaja sobre pares de individuos llamados padres, intercambiando material genético entre éstos para generar un nuevo individuo llamado hijo.

Los métodos de cruce son descritos a continuación. Entre los más utilizados están la cruce de *1-punto* y la cruce *multi-punto*. La cruce se realiza tomando en cuenta un factor que determina la probabilidad que tienen dichos padres para cruzarse.

Cruza de 1-punto En este mecanismo de cruce se toman dos individuos seleccionados y se escoge un punto aleatorio dentro de sus cadenas genéticas. La cruce se realiza intercambiando las secciones del padre y de la madre en dicho punto.

Cruza Multipunto Este mecanismo es igual que la cruce básica, con la diferencia de que se establece más de un punto de cruce. Este tipo de cruce es muy efectivo cuando las cadenas tienen una longitud extensa, por que se realiza un mejor intercambio de material genético.

Cruza Uniforme Al igual que para los otros dos métodos de cruce mencionados, este operador toma dos individuos. Para formar el nuevo individuo selecciona cada gen de cualquiera de los dos padres los cuales cuentan con la misma probabilidad.

Dentro de estos tipos de cruce, la menos disruptiva es la cruce de 2-puntos.

Mutación

El operador de mutación altera aleatoriamente algunos elementos del cromosoma con una baja probabilidad. Este operador es de vital importancia porque evita que el algoritmo converja prematuramente, añadiendo aleatoriedad en la búsqueda y diversidad a la población. Esto permite explorar nuevos subespacios de soluciones, por lo que, si el subespacio es bueno en términos de aptitud, se puede evolucionar a la población hacia un mejor estado. Entre los métodos de mutación más comunes están:

Mutación de bit En este mecanismo de mutación existe una única probabilidad de que se produzca una mutación de algún bit. De producirse, el algoritmo toma aleatoriamente un bit, y lo invierte.

Mutación multi-bit En este mecanismo cada bit tiene una probabilidad de mutar o no, que es calculada en cada pasada del operador de mutación multi-bit. Este método resulta más tardado, pues se tiene que calcular un evento aleatorio para cada bit. La ventaja es que se asemeja más la mutación natural y por lo tanto la exploración es más efectiva.

2.1.6. Parámetros de control del Algoritmo Genético

Una vez que se tienen seleccionados el tipo de codificación, la función de aptitud y los operadores genéticos a utilizar, también es necesario y de gran importancia realizar la selección de los parámetros de control del algoritmo genético. Estos parámetros son los valores correspondientes a ciertas cantidades que aún no están establecidas como lo son el tamaño de la población, el número máximo de generaciones, la probabilidad de cruce y mutación, así como el número de individuos de elite que tienen la mejor evaluación de la función de aptitud en la generación actual.

Cada uno de estos parámetros afecta de distinta forma el desempeño del AG, algunos en mayor medida que otros. Por ejemplo, una población pequeña afecta al algoritmo, debido a que no se cuenta con la suficiente información para realizar una búsqueda en los diferentes hiperplanos. Sin embargo, una población bastante grande puede converger en óptimos locales incluso al global. Otro problema es que se necesitan más evaluaciones por población lo que resulta en un mayor tiempo de cómputo y por lo tanto converge más lento.

La probabilidad de cruce y mutación controlan la frecuencia con que estos operadores son aplicados en el total de la población, y pueden hacer que la búsqueda converja rápidamente a una solución desfavorable o que la búsqueda sea totalmente aleatoria. La selección más adecuada de los parámetros se debe realizar mediante un proceso de sintonización del AG.

2.1.7. Algoritmos genéticos jerárquicos

En comparación con el AG simple, la principal diferencia de los Algoritmos Genéticos Jerárquicos (AGJ) es la estructura del cromosoma [51].

Desde el punto de vista biológico, la estructura genética de un cromosoma está formada por un número de variaciones de genes dispuestos de una manera jerárquica. De acuerdo con lo anterior, Man et. al. [52] propuso una estructura jerárquica del cromosoma para emular la formulación biológica de la estructura del ADN. El cromosoma computacional en el AGJ consiste de dos tipos de genes, conocidos como genes de control y genes paramétricos.

Típicamente, los genes de control son codificados como dígitos binarios, mientras que los genes paramétricos son codificados como números reales o cualquier otra representación. El propósito de los genes de control es habilitar o inhibir a los genes paramétricos, lo cual es particularmente importante para determinar la estructura genética del cromosoma. Por otra parte, los genes paramétricos son usualmente usados para representar las variables de diseño. La principal ventaja del AGJ es que tanto la estructura del sistema como las variables paramétricas pueden ser optimizadas simultáneamente, pero además los operadores genéticos estándar siguen siendo utilizados sin necesidad de realizar ningún cambio. Como ejemplo de aplicaciones de los algoritmos genéticos jerárquicos en problemas de optimización tenemos las mostradas en [53, 54].

2.2. Base de funciones

Una base de funciones esta formada por funciones con características especiales que las hacen útiles para generar bases ortogonales, que puedan ser utilizadas para generar combinaciones lineales de funciones, las cuales cumplan con criterios de suavidad, derivabilidad, continuidad y adaptabilidad. Entre las funciones base se encuentran dos que son muy populares entre la comunidad científica: las funciones spline y las funciones de base radial.

2.2.1. Funciones B-splines

Las funciones base *B-spline* son construidas de trozos de polinomios unidos en ciertos valores de x llamados “nodos”, las cuales cumplen con criterios de continuidad y suavidad determinados por el usuario. Una curva *B-spline* es modelada usando las siguientes consideraciones: Sean $\{\tau_1, \dots, \tau_m\}$ un conjunto de m puntos colocados a lo largo del dominio de la variable independiente x , llamados nodos interiores, tal que $a = x_1 < \tau_1 < \dots < \tau_m < x_n = b$; y sea

$\mathbf{t} = \{t_1, \dots, t_{m+2k}\}$ un vector de nodos definido por:

$$t_i = \begin{cases} a, & \text{si } i \leq k \\ \tau_{i-k}, & \text{si } k < i < k + m \\ b, & \text{si } i \geq k + m \end{cases} \quad (2.1)$$

Entonces, la i -ésima función base B -spline denotada como $B_{i,k}(x)$ de orden k está dada por las siguientes relaciones recursivas:

$$B_{i,k}(x) = \frac{x - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(x) + \frac{t_{i+k} - x}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(x) \quad (2.2)$$

y

$$B_{i,1}(x) = \begin{cases} 1, & \text{si } t_i \leq x < t_{i+1} \\ 0, & \text{en otro caso} \end{cases} \quad (2.3)$$

donde t_i es el vector de nodos definido sobre el dominio de la variable independiente x , y k es el orden de la B -spline que representa el grado de las funciones base más uno. Esto es, si deseamos funciones base cúbicas, el orden de la B -spline debe ser cuatro. Dadas estas definiciones, podemos expresar una función $f(x)$ mediante la siguiente combinación lineal de funciones base B -spline:

$$f(x) = \sum_{i=1}^{m+k} \alpha_i B_{i,k}(x) \quad (2.4)$$

donde α_i son los coeficientes de la B -spline y $B_{i,k}$ son las funciones base B -spline con m número de nodos interiores y orden k , definidas sobre el vector de nodos \mathbf{t} de acuerdo con las ecuaciones (2.1),(2.2) y (2.3).

Si k es especificada de antemano, podemos definir completamente una función $f(x)$ mediante $\theta = \{\tau, \alpha\}$, donde $\tau = \{\tau_1, \dots, \tau_m\}$ y además $\alpha = \{\alpha_1, \dots, \alpha_{m+k}\}$. Ahora, el problema consiste en encontrar el número y la posición de los τ_i nodos, para posteriormente calcular los coeficientes α_i . En varios métodos propuestos, los coeficientes son calculados en un paso adicional mediante el uso de mínimos cuadrados o cualquier otra técnica intensiva, tal como los métodos Bayesianos o mínimos cuadrados ponderados. Una discusión más detallada acerca de las funciones B -spline puede ser consultada en [55, 56].

2.2.2. Superficies de producto tensorial de B-splines

Las superficies B -spline son construidas mediante un producto tensorial de funciones base B -spline univariadas. La superficie B -spline es modela-

da usando las siguientes consideraciones: Definimos $\{u_1, \dots, u_m\}$ como un conjunto de m puntos colocados a lo largo del dominio de la variable x y $\{v_1, \dots, v_n\}$ como un conjunto de n puntos colocados a lo largo del dominio de la variable y , los cuales son llamados nodos interiores. Así, los vectores de nodos son definidos como sigue:

$$\begin{aligned} \mathbf{u} : u_{1-k} =, \dots, = u_0 = a < u_1 < \dots < u_m < b = u_{m+1} = \dots = u_{m+k} \\ \mathbf{v} : v_{1-l} =, \dots, = v_0 = c < v_1 < \dots < v_n < d = v_{n+1} = \dots = v_{n+l} \end{aligned} \quad (2.5)$$

con estos supuestos, una función $f(x, y)$ puede ser escrita como el producto tensorial:

$$f(x, y) = \sum_{i=1}^{m+k} \sum_{j=1}^{n+l} P_{i,j} B_{i,k}(x) B_{j,l}(y) \quad (2.6)$$

donde $P_{i,j}$ son los coeficientes de las B -splines y $B_{i,k}(x), B_{j,l}(y)$ son las funciones base B -spline de orden k and l respectivamente definidas sobre los vectores de nodos \mathbf{u} and \mathbf{v} . Las funciones base B -spline están dadas por las ecuaciones (2.2) y (2.3).

De tal forma que, sí k y l son especificados previamente, la función $f(x, y)$ puede ser completamente definida por $\theta = \{\mathbf{u}, \mathbf{v}, P\}$, donde \mathbf{u} y \mathbf{v} son los vectores de nodos y P es la matriz de coeficientes. Entonces, el problema nuevamente consiste en encontrar el número y la posición de los nodos interiores $\{u_1, \dots, u_m, v_1, \dots, v_n\}$ y entonces estimar los coeficientes $P_{i,j}$. Por el contrario del caso univariado, calcular la matriz de coeficientes no se puede realizar mediante métodos estándar, debido al hecho de que es un problema de optimización no lineal de grandes dimensiones. El tema de las superficies de producto tensorial con B -splines es tratado a detalle en [57].

2.2.3. Funciones de Base Radial

Las funciones de Base Radial (FBR), se utilizan principalmente para la interpolación de funciones [58], lo anterior debido a que el número de parámetros que este tipo de funciones requiere, hace complicada su utilización en problemas de aproximación, especialmente por métodos tradicionales.

Además de lo anterior, las FBR se utilizan como elemento principal de procesamiento dentro de un tipo específico de Redes Neuronales Artificiales conocidas como Redes Neuronales de Base Radial, [59, 60, 61]. Al igual que las redes neuronales de tipo perceptrón multicapa, estas sirven como aproximadores universales. La principal desventaja en las redes neuronales de

base radial radica en la necesidad de reducir el número de parámetros de las funciones de base radial a fin de permitir la aplicación de métodos de aprendizaje tradicionales [62].

Tabla 2.1: Funciones de base radial. (para $r = |x - c_j|$)

Función	Nombre
$\phi(r) = r$	Lineal
$\phi(r) = r^3$	Cúbica
$\phi(r) = r^2 \log(r)$	Placa delgada
$\phi(r) = \sqrt{1 + r^2}$	Multicuádrica
$\phi(r) = \exp^{-\frac{r^2}{\sigma^2}}$	Gaussiana

Definición:

Las FBR son una clase particular de funciones cuya característica principal es que su respuesta decrece (o crece) monótonamente en todas direcciones respecto de la distancia a un punto central. Si su respuesta es decreciente respecto de la distancia, entonces son consideradas como funciones con respuesta local, por el contrario si su respuesta se incrementa respecto de la distancia a su centro son consideradas como funciones con respuesta global.

Las FBR están determinadas por la siguiente expresión:

$$\Phi_j(x) = \phi(|x - c_j|) \quad (2.7)$$

donde $|\cdot|$ denota la norma de distancia utilizada para medir la separación de cualquier punto x al centro c_j de la j -ésima función base y $\phi(\cdot)$ es un tipo específico de FBR. La norma de distancia más comúnmente usada es la euclidiana, por otra parte las FBR que han sido ampliamente utilizadas y analizadas en muchos trabajos de investigación se muestran en la Tabla 2.1.

De acuerdo con la definición anterior podemos construir cualquier función como una combinación lineal de m funciones de base radial, dada por la siguiente expresión:

$$f(x) = \sum_{j=1}^m \lambda_j \Phi_j(x) \quad (2.8)$$

donde λ_j son los coeficientes y las Φ_j forman la base de funciones. Para lograr la linealidad del modelo es necesario fijar tres parámetros importantes.

Primero, el tipo de FBR a utilizar, segundo, los parámetros dependientes del tipo específico de FBR usada y finalmente el número m de funciones radiales y los centros de cada una de ellas. Todos los parámetros mencionados hacen que el uso de las FBR siempre esté bajo ciertas restricciones, para que pueda ser resuelto por métodos convencionales.

Finalmente cabe mencionar que, dado que cada punto $x_i \in R^n$, las FBR se pueden extender a n dimensiones sin que se aumente su grado de complejidad, esto debido a que sólo incrementa el calculo de la norma de distancia utilizada. Esto representa su principal ventaja, frente a otras funciones base.

Capítulo 3

Algoritmos Genéticos aplicados en Óptica e Ingeniería

En este capítulo se describen las aplicaciones de los algoritmos genéticos, implementadas para la solución de problemas del área de la óptica y la ingeniería. Entre los cuales se encuentran el ajuste de curvas y superficies [63, 64], así como también una aplicación sobre demodulación de imágenes de interferogramas que contienen franjas cerradas.

En el desarrollo de las secciones se discuten a detalle los algoritmos propuestos, los experimentos realizados y los resultados obtenidos por cada una de las aplicaciones implementadas.

3.1. Ajuste de curvas B-spline

3.1.1. Definición del problema del ajuste de curvas

Podemos definir el problema del ajuste de curvas de la siguiente manera: dados n pares de datos (x_j, y_j) , $j = 1, \dots, n$, que pueden ser modelados de acuerdo con la expresión

$$y_j = f(x_j) + \epsilon_j \quad (3.1)$$

donde $f(x)$ es una relación funcional desconocida, el término ϵ_j representa los errores de medición, y y_j es la j -ésima observación en el punto x_j . Entonces, el problema es encontrar el mejor estimado de la función $f(x)$ desconocida.

En el presente trabajo suponemos que $f(x)$ es una función suave que puede ser aproximada en el intervalo $[a, b]$ mediante una curva B-spline, y

que el ruido presente en los datos es independiente con media cero y varianza constante. En este sentido el problema consiste en encontrar el número y la posición de los nodos, así como los coeficientes de la B-spline tales que logren ajustar los mejor posible una curva B-spline al conjunto de datos dado.

3.1.2. Ajuste de curvas B-spline usando un AGJ

En esta propuesta utilizamos un AGJ para determinar de forma simultánea el número y la posición de los nodos (estructura del modelo) y los coeficientes de la B-spline (parámetros del modelo) minimizando un criterio estadístico de selección de modelos. Para ello, las principales características a considerar son: (1) la codificación del cromosoma de las soluciones potenciales, (2) la función de aptitud para la evaluación de los individuos, (3) los operadores utilizados para evolucionar los individuos y (4) un esquema de optimización multi-objetivo.

Codificación del Cromosoma

Para la codificación del cromosoma utilizamos una cadena binaria de longitud fija, la cual representa el número y la posición de los nodos interiores τ , y un vector \mathbf{r} de números reales para representar los coeficientes α de la B-spline. El cromosoma θ de un individuo se representa por la siguiente expresión:

$$\theta = \{b_1, \dots, b_m, r_{1-k/2}, \dots, r_0, \dots, r_m, r_{m+1}, \dots, r_{m+k/2}\}$$

donde cada b_i es un bit de control y r_i es un valor real (coeficiente).

Cada bit de control habilita o inhibe simultáneamente uno de los nodos interiores y uno de los coeficientes. Para lo cual se establece una correspondencia uno a uno entre los nodos interiores y los coeficientes, tal que son activados al mismo tiempo. Los valores reales definen los coeficientes de la B-spline. La estructura general del cromosoma es mostrada gráficamente en la figura 3.1. De la misma figura podemos observar claramente la naturaleza jerárquica del cromosoma, de la cual se desprende el nombre los algoritmos genéticos jerárquicos (AGJ).

Este esquema de representación no permite nodos duplicados, porque nuestro interés particular esta enfocado en funciones suaves. Sin embargo, este puede ser extendido para manejar funciones discontinuas agregando un tipo especial de gen para este propósito.

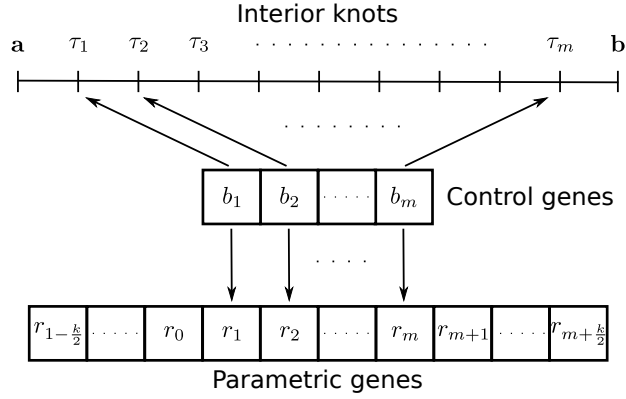


Figura 3.1: Estructura General del cromosoma.

Función de aptitud

Para determinar la estructura del modelo, así como también los parámetros del mismo, un esquema bi-objetivo es propuesto como función de aptitud. En otras palabras, la aptitud de un cromosoma es una medida basada en dos funciones objetivo: el criterio de información de Akaike y la función estructura de nodos.

Criterio de información de Akaike. El criterio de información de Akaike (AIC por sus siglas en inglés), originalmente desarrollado por Akaike [65], es una medida de la bondad del ajuste de una estimación y el número de parámetros necesarios para alcanzar ese grado particular de ajuste [4, 9]. El AIC está dado por:

$$AIC(\hat{f}) = n \log \sum_{j=1}^n \{y_j - \hat{f}(x_j)\}^2 + 2p \quad (3.2)$$

donde \hat{f} es una estimación de f (vea la ecuación (3.1)) denotada por la ecuación (2.4). El término p es el número de parámetros del modelo.

En el AIC, la suma de los cuadrados de los residuos del primer término es usada como medida de la desviación de una función estimada \hat{f} de la original f , y el segundo término es una penalización al incremento del número de parámetros p . En este trabajo el número de parámetros consiste del número de nodos interiores m y el número de coeficientes $m + k$, así $p = 2m + k$.

Función estructura de nodos. Para encontrar la mejor estructura para la posición de los nodos, usamos una función de penalización. A partir de [10],

el término de penalización es calculado como:

$$P(\tau) = \sum_{j=1}^m \log k_j + \log m \quad (3.3)$$

donde m es el número de nodos y k_j es el número de elementos x_i que satisfacen $\tau_{j-1} \leq x_i < \tau_j$.

En esta función, el primer término penaliza el número de los nodos y el segundo término penaliza la distancia entre nodos. Por lo tanto, los individuos con pocos nodos y mejor distribución son favorecidos.

Esquema de Selección y operadores genéticos

Esquema de Selección. El método de la ruleta es usado como operador de selección. En este método, cada individuo es asignado a una rebanada de la rueda de la ruleta. Esta estrategia de selección favorece a los individuos mejor adaptados pero le da oportunidad a los individuos menos adaptados de sobrevivir de acuerdo a su valor de aptitud en relación del promedio poblacional. Para prevenir la convergencia prematura, el método de escalamiento sigma [18] es utilizado. Este método trata de mantener una presión de selección relativamente constante durante todo el proceso de evolución y es calculado de acuerdo a:

$$F_{new} = \begin{cases} F_{act} - (\bar{F} - c \cdot \sigma) & \text{si } (F_{act} > \bar{F} - c \cdot \sigma) \\ 0 & \text{en otro caso} \end{cases} \quad (3.4)$$

donde F_{new} es el nuevo valor de aptitud escalado, F_{act} es el valor de aptitud actual, \bar{F} es el valor de aptitud promedio, σ es la desviación estándar de la población y c es una constante que permite controlar la presión de selección. Además, para prevenir la pérdida de la mejor solución encontrada durante el proceso de evolución, la idea del elitismo es usada. El elitismo funciona como sigue: los n mejores individuos son seleccionados para ser copiados sin cambio alguno. Estos individuos son comúnmente llamados individuos elite. Así, el resto de la población es generada por el método de selección de la ruleta.

Operador de cruza. El operador de cruza uniforme es usado para el cromosoma tipo binario y el operador de cruza binario simulado (SBX) [66] es usado para el cromosoma tipo real. Estos operadores son aplicados con la misma probabilidad de cruza. En el método de cruza uniforme, dos padres

son seleccionados para ser recombinados en un nuevo individuo. Cada bit del nuevo individuo es seleccionado de uno de los padres dependiendo de una probabilidad fija. Por otra parte, en el método SBX, dos nuevos individuos c_1 y c_2 son generados a partir de los padres p_1 y p_2 usando una distribución de probabilidad.

El procedimiento usado en SBX puede ser descrito como sigue: primero, un número aleatorio u entre 0 y 1 es generado. Entonces, el valor de β es distribuido como:

$$\beta = \begin{cases} (2u)^{\frac{1}{\eta+1}} & \text{si } u \leq 0,5 \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta+1}} & \text{en otro caso} \end{cases} \quad (3.5)$$

donde η es un número real no negativo que controla la distancia entre los padres y los nuevos individuos generados. Después de obtener β , los nuevos individuos son calculados de acuerdo a:

$$\begin{aligned} c_1 &= 0,5[(1 + \beta)p_1 + (1 - \beta)p_2] \\ c_2 &= 0,5[(1 - \beta)p_1 + (1 + \beta)p_2] \end{aligned} \quad (3.6)$$

Operador de mutación. El método de mutación de bit es aplicado sobre el cromosoma tipo binario. En este método, cada bit es invertido o no dependiendo de la probabilidad de mutación. Para el cromosoma tipo real cada valor numérico γ es cambiado también dependiendo de la misma probabilidad de mutación de acuerdo a:

$$\gamma_i = \gamma_i + \delta(rand - 0,5) \quad (3.7)$$

donde δ es el máximo incremento o decremento del valor real y $rand$ es una función que genera un valor aleatorio entre 0 y 1.

Esquema de optimización multi-objetivo

Dado que se está proponiendo un AGJ para la búsqueda del mejor modelo B-spline usando un esquema bi-objetivo. Este esquema toma en cuenta dos valores objetivo: uno es el resultado del AIC y el otro es el valor de la función de estructura de nodos. Por lo que, la función de aptitud es la suma ponderada de las dos funciones objetivo la cual puede ser expresada como:

$$F(\theta) = w_1 AIC(\hat{f}_\theta) + w_2 P(\tau) \quad (3.8)$$

donde w_1 y w_2 son factores de escalamiento para normalizar los objetivos.

En el área de los AG, la técnica de suma ponderada es usada para ajustar la búsqueda genética hacia la frontera de Pareto [67]. Dicha frontera esta compuesta por el conjunto de soluciones Pareto-óptimas. Específicamente, el uso de pesos fijos conduce la búsqueda del AG hacia un área fija en el espacio de búsqueda [68]. Así, para incrementar dicha área usamos el modelo de islas [69]. Este enfoque usa múltiples islas para encontrar varias soluciones en la frontera de Pareto [70], dado que cada isla puede seguir diferente trayectoria de búsqueda a través del espacio de búsqueda. Además, el modelo de islas puede reducir la convergencia prematura porque preserva la diversidad genética [71].

En el modelo de islas, la población es dividida en varias sub-poblaciones o islas, donde cada una de éstas evoluciona independientemente como un simple AG. Aunque cada isla es independiente una de la otra, se realiza un intercambio periódico de individuos entre ellas. Estos individuos llamados “migrantes” se intercambian en un proceso llamado migración.

El modelo de islas introduce un nuevo operador, llamado migración. Éste consiste en un intercambio de individuos entre las sub-poblaciones, permitiendo con esto cubrir el espacio de búsqueda de una mejor manera y evitar la convergencia prematura del AG. El operador de migración trabaja de la siguiente manera: n individuos son seleccionados para ser emigrados y reemplazados desde cada sub-población (tamaño de migración). Esta selección puede hacerse en cuatro formas posibles, en nuestro caso particular, seleccionamos de cada sub-población a los mejores individuos para ser migrados y los peores individuos para ser reemplazados (política de migración). El proceso de migración es aplicado cada n generaciones (intervalo de migración). Las conexiones entre las sub-poblaciones forman un anillo (topología de comunicación).

3.1.3. Pruebas y resultados del ajuste de curvas

Se aplicó el método propuesto a simulaciones numéricas de cinco diferentes funciones suaves de referencia (vea [6, 12, 14, 15, 72] para mayores detalles acerca de las funciones), las cuales son mostradas en la tabla 3.1, incluyendo una comparación con el conocido método Bayesiano de Spline de Regresión Adaptativa [6] (BARS por sus siglas en inglés).

Para este estudio se consideran diferentes niveles de ruido y número de muestras. Para ser más específico, añadimos ruido a cada función a diferentes

razones 2, 3 y 4, y entonces variamos el número de muestras en 50, 100 y 200. De tal forma que, cada configuración experimental consta de 9 diferentes combinaciones para cada función. Además, para cada combinación una colección de 100 conjuntos de datos ruidosos son generados, esto es, se probó el método sobre un total de 900 conjuntos para cada función.

Tabla 3.1: Funciones de prueba

No.	Función
1	$f(x) = \sin^3(2\pi x^3)$
2	$f(x) = (4x - 2) + 2 \exp(-16(4x - 2)^2)$
3	$f(x) = \sin(x) + 2 \exp(-30x^2)$
4	$f(x) = \sin(2(4x - 2)) + 2 \exp(-16(4x - 2)^2)$
5	$f(x) = 10(4x - 2)/(1 + 100(4x - 2)^2)$

En este trabajo, la razón de la señal-al-ruido (SNR por sus siglas en inglés) está definida como $SD(f)/\sigma$ [6]. Así, para generar un conjunto de datos ruidosos, evaluamos una función específica en n puntos de diseño x_i uniformemente distribuidos sobre el intervalo $[0, 1]$. En este punto, se añade el ruido normal con media cero y σ conocida al conjunto de datos. En la función de prueba 3, los puntos de diseño x_i son uniformemente distribuidos en el intervalo $[-2, 2]$ y posteriormente son escalados al intervalo $[0, 1]$. Por último, el método propuesto y el algoritmo BARS fueron aplicados para estimar las funciones de prueba.

Para el AGJ, usamos una función B-spline cúbica ($k = 4$) y τ está definido como un subconjunto de los puntos de diseño. En el inicio, la población es inicializada aleatoriamente; cada gen de control b_i es seleccionado aleatoriamente en el intervalo $[0, 1]$ y cada gen paramétrico r_i es calculado como un número real aleatorio definido sobre el rango $[\min(y), \max(y)]$ de la variable dependiente y . Los parámetros de control del AGJ fueron sintonizados experimentalmente y se muestran en la tabla 3.2.

El modelo de islas fue ejecutado en forma serial en lugar de en paralelo, por cuestiones del tipo de equipo de cómputo utilizado. Pero de poder ejecutarse en forma paralela este disminuiría el tiempo de ejecución entre n , dependiente de las características del equipo de cómputo utilizado. El número de islas usado en los experimentos fue seleccionado como sigue: para las funciones 1 y 5 se consideraron 10 islas, en las funciones 2 y 4, se usaron 20 islas y para la función de prueba 3, se usaron 30 islas. Estas consideraciones

Tabla 3.2: Parámetros de control del AGJ

Parámetro	Valor
Tamaño de Sub-población	60
Probabilidad de Cruza	0.85
Probabilidad de Mutación	0.008
Número de Individuos Elite	6
Número de Sub-poblaciones	10–30
Tamaño de Migración	3
Intervalo de Migración	400
Política de Migración	Mejores reemplazan peores
Topología de Comunicación	En anillo

fueron hechas para reducir el costo computacional en nuestros experimentos; sin embargo recomendamos considerar 30 islas. El algoritmo fue evolucionado durante 900 generaciones en todos los casos. El mejor individuo de cada isla fue seleccionado para al final obtener el que tenga el máximo valor de aptitud.

Para propósitos de comparación, se realizaron los mismos experimentos numéricos con el algoritmo BARS. Para estimar las funciones de todos los conjuntos de datos se usaron los parámetros por defecto que se incluyen en el código, excepto para la configuración experimental con $n = 50$ muestras, donde el parámetro para la prioridad fue fijado a $c = (1, 20)$ y el número inicial de nodos para la spline fue fijado a uno. Los archivos fuente del algoritmo BARS fueron codificados en lenguaje C y encapsulados para ejecutarse en el software R [73] y pueden ser descargados desde <http://www.stat.cmu.edu/kass/bars/bars.html>. Refiérase a [16] para una completa descripción de este algoritmo.

Para comparar los resultados, usamos el error cuadrático promedio (MSE por sus siglas en inglés) dado por:

$$MSE(\hat{f}) = \frac{1}{n} \sum_{i=1}^n \{\hat{f}(x_i) - f(x_i)\}^2 \quad (3.9)$$

donde \hat{f} es la función estimada de f , obtenida por cada método aplicado. El valor de MSE es calculado y registrado para cada función. Los diagramas de caja de los valores de $MSE(\hat{f})$ son mostrados en la figura 3.2. Los diagramas de caja muestran una representación visual de la distribución de los valores de MSE y facilita la comparación entre los resultados de ambos métodos.

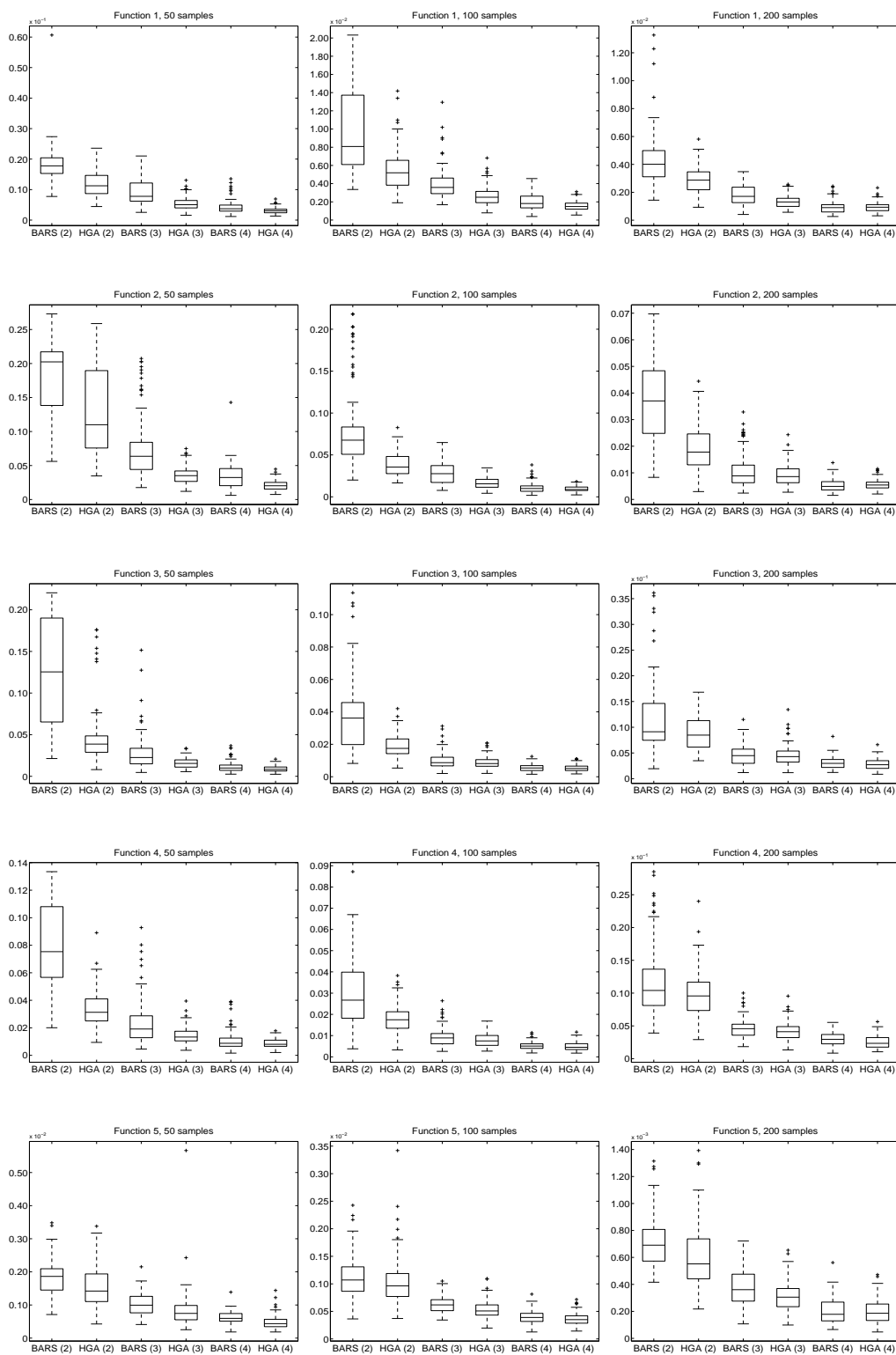


Figura 3.2: Diagramas de caja de los valores de MSE para ambos métodos.

Comparando distribuciones

Los diagramas de caja dividen y organizan una distribución de datos en cuartiles. Los cuartiles dividen los datos en cuatro partes iguales; 25 % de los datos están en cada parte. La caja central es usada para indicar la posición del cuartil superior (tercero) e inferior (primero); el interior de esta caja indica el rango intercuartil (IQR), el cual es la distancia entre el cuartil superior e inferior y consiste del 50 % de los datos. La línea horizontal dentro de la caja es el valor de la mediana de la distribución. Las líneas verticales, algunas veces referidas como bigotes, extendidas desde la orilla de la caja hasta las observaciones más baja y más alta, muestran la extensión total de la distribución; si hay observaciones que caen más allá que $1,5 \times IQR$ por encima del cuartil superior o por debajo del cuartil inferior, estas son consideradas como valores atípicos sospechosos y se representan como puntos individuales marcados con una cruz.

Los diagramas de caja son particularmente efectivos para comparar distribuciones. En la figura 3.2, se muestra un panel para cada combinación de función de prueba y número de muestras. En cada panel, los diagramas de caja corresponden a los algoritmos probados para varios valores de SNR indicados entre paréntesis de tal forma que el nivel de ruido añadido a la señal decrece de izquierda a derecha. Las columnas de paneles corresponden de izquierda a derecha a experimentos con 50, 100 y 200 muestras respectivamente. Las filas de paneles corresponden de arriba a abajo a las funciones de prueba de 1 a 5. Como podemos ver en la misma figura, en la mayoría de los casos los valores de MSE del AGJ son considerablemente más bajos que aquellos mostrados por el algoritmo BARS.

En algunos casos el cuartil superior de los valores de MSE para el AGJ está por debajo del primer cuartil de aquellos mostrados por algoritmo BARS; lo cual implica que el 75 % de los valores de MSE están por debajo del 75 % de los valores de MSE del algoritmo BARS; en otras palabras, menos del 25 % de los valores de MSE para el algoritmo BARS son comparables con el 75 % de aquellos mostrados por el AGJ. La diferencia más notable puede ser vista en la figura 3.2 para la función de prueba 4 con 50 muestras y $SNR = 2$, donde el cuartil superior del AGJ esta muy por debajo del cuartil inferior del mostrado por el algoritmo BARS.

Además, la mediana de los valores de MSE para el AGJ son virtualmente más bajas para todos los casos. La extensión de la mitad central (la caja) de los valores de MSE es marcadamente menos variable y parece ser simétrica

alrededor de la mediana en la mayoría de los casos, contrario a aquellas mostradas por el algoritmo BARS. Esto puede ser visto claramente en la figura 3.2 para la función de prueba 3 con 200 muestras, donde los diagramas de caja presentan medianas similares pero los cajas para el algoritmo BARS están más extendidas que aquellas mostradas por el AGJ y las medianas están sesgadas al menos para el SNR nivel 2 y 3.

En cuanto a los valores atípicos, aunque ambos algoritmos presentan valores inusuales de MSE, nuestro algoritmo tiene significativamente menos valores atípicos que el algoritmo BARS. Consideramos que estos valores representan casos donde el AGJ converge hacia óptimos locales o puntos arbitrarios en lugar del óptimo global; sin embargo, este comportamiento es esperado en cualquier técnica heurística.

Todo lo expuesto anteriormente es particularmente visible en experimentos bajo las peores condiciones, tales como los que tienen un mayor nivel de ruido y el menor número de muestras. También en la figura 3.2, podemos observar como el desempeño de los dos algoritmos parece ser similar cuando las condiciones del experimento son más favorables; en otras palabras, en experimentos con el menor nivel de ruido y el mayor número de muestras. Por lo que nuestro método presenta una considerable mejora sobre el algoritmo BARS.

Error estándar

Además, se calculó la media y su error estándar para los valores de MSE de ambos métodos. Dichos valores son resumidos en tres tablas, una para cada conjunto de muestras 50, 100 y 200 correspondiendo a las tablas 3.3 a 3.5 respectivamente. En estas tablas podemos apreciar que el AGJ alcanza los valores de MSE más bajos en todos los experimentos con menos incertidumbre como lo es indicado por su error estándar. Particularmente, la tabla 3 muestra la mayor diferencia estadística significativa en la función de prueba 3 y $SNR = 2$, donde la media de los valores de MSE para el AGJ está muy alejada de la media de los valores de MSE para el algoritmo BARS. La mayor diferencia estadística significativa es marcada con negritas en cada tabla.

Para evaluar y comparar visualmente el desempeño de los dos métodos en cada función, los resultados obtenidos fueron ordenados de acuerdo a sus valores de MSE. El 50° mejor resultado de la aproximación \hat{f} para cada método es mostrada gráficamente en la figura 3.3. En dicha figura solo mostramos los resultados para la configuración experimental de 50 muestras y $SNR = 3$,

Tabla 3.3: Media con error estándar de los valores de MSE obtenidos sobre 100 conjuntos de datos de 50 muestras.

Prueba	Método	SNR=2	SNR=3	SNR=4
1	BARS	0.018297 ± 0.000575	0.008954 ± 0.000384	0.004249 ± 0.000224
	AGJ	0.012040 ± 0.000432	0.005373 ± 0.000220	0.003091 ± 0.000102
2	BARS	0.182099 ± 0.005226	0.076370 ± 0.004542	0.034623 ± 0.001866
	AGJ	0.130274 ± 0.006493	0.035869 ± 0.001256	0.020783 ± 0.000753
3	BARS	0.126931 ± 0.006255	0.028209 ± 0.002249	0.011341 ± 0.000664
	AGJ	0.047211 ± 0.003380	0.015743 ± 0.000606	0.008992 ± 0.000367
4	BARS	0.079200 ± 0.003057	0.023790 ± 0.001648	0.010666 ± 0.000695
	AGJ	0.033549 ± 0.001358	0.014265 ± 0.000595	0.008651 ± 0.000318
5	BARS	0.001842 ± 0.000055	0.001014 ± 0.000033	0.000634 ± 0.000019
	AGJ	0.001560 ± 0.000062	0.000836 ± 0.000059	0.000485 ± 0.000021

Tabla 3.4: Media con error estándar de los valores de MSE obtenidos sobre 100 conjuntos de datos de 100 muestras.

Prueba	Método	SNR=2	SNR=3	SNR=4
1	BARS	0.009440 ± 0.000406	0.004002 ± 0.000175	0.002001 ± 0.000090
	AGJ	0.005437 ± 0.000235	0.002601 ± 0.000103	0.001566 ± 0.000058
2	BARS	0.081809 ± 0.004814	0.027851 ± 0.001302	0.010816 ± 0.000619
	AGJ	0.038531 ± 0.001367	0.016401 ± 0.000641	0.009406 ± 0.000339
3	BARS	0.038098 ± 0.002201	0.009833 ± 0.000508	0.005502 ± 0.000219
	AGJ	0.018831 ± 0.000689	0.008605 ± 0.000367	0.005274 ± 0.000202
4	BARS	0.030100 ± 0.001545	0.009228 ± 0.000434	0.005241 ± 0.000179
	AGJ	0.017739 ± 0.000674	0.007906 ± 0.000313	0.005021 ± 0.000198
5	BARS	0.001130 ± 0.000037	0.000619 ± 0.000015	0.000401 ± 0.000012
	AGJ	0.001042 ± 0.000045	0.000530 ± 0.000016	0.000362 ± 0.000011

Tabla 3.5: Media con error estándar de los valores de MSE obtenidos sobre 100 conjuntos de datos de 200 muestras.

Prueba	Método	SNR=2	SNR=3	SNR=4
1	BARS	0.004297 ± 0.000190	0.001827 ± 0.000074	0.000940 ± 0.000043
	AGJ	0.002904 ± 0.000100	0.001339 ± 0.000044	0.000939 ± 0.000036
2	BARS	0.036302 ± 0.001544	0.010786 ± 0.000656	0.005341 ± 0.000233
	AGJ	0.019254 ± 0.000818	0.009209 ± 0.000400	0.005560 ± 0.000199
3	BARS	0.011426 ± 0.000689	0.004636 ± 0.000199	0.003041 ± 0.000112
	AGJ	0.008967 ± 0.000332	0.004547 ± 0.000201	0.002796 ± 0.000105
4	BARS	0.011645 ± 0.000556	0.004614 ± 0.000155	0.003024 ± 0.000102
	AGJ	0.009762 ± 0.000342	0.004240 ± 0.000150	0.002553 ± 0.000098
5	BARS	0.000712 ± 0.000018	0.000373 ± 0.000014	0.000203 ± 0.000009
	AGJ	0.000603 ± 0.000023	0.000312 ± 0.000011	0.000204 ± 0.000009

junto con un ejemplo del conjunto de datos ruidosos simulado. En esta figura, podemos comparar cualitativamente el desempeño de ambos algoritmos. A partir de la figura 3.3 podemos sugerir que el AGJ tiene una mayor habilidad para adaptarse a los picos agudos en las curvas a pesar de la poca cantidad de muestras y el alto nivel de ruido en la señal, en contraste con el algoritmo BARS el cual muestra curvas mas suaves con un sobre ajuste en algunas secciones de las curvas.

Finalmente, podemos asumir que el éxito de nuestro enfoque es debido a la búsqueda simultanea global realizada sobre todos los parámetros del modelo de la B-spline. Esto muestra el potencial del AGJ para manejar la pérdida de información y un alto nivel de ruido. Por otra parte, la principal desventaja es el costo computacional requerido por el AGJ en comparación con el algoritmo BARS. El tiempo promedio de cómputo para el algoritmo BARS fue de 20 segundos para los experimentos con 200 muestras, comparado a los 166 segundos del AGJ. Sin embargo, esta desventaja puede ser solventada si consideramos una implementación en paralelo del modelo de islas. Este algoritmo fue implementado en lenguaje C++ y ambos algoritmos fueron ejecutados en una PC usando un procesador AMD Phenom II a 1.9 GHz con 4GB de memoria RAM.

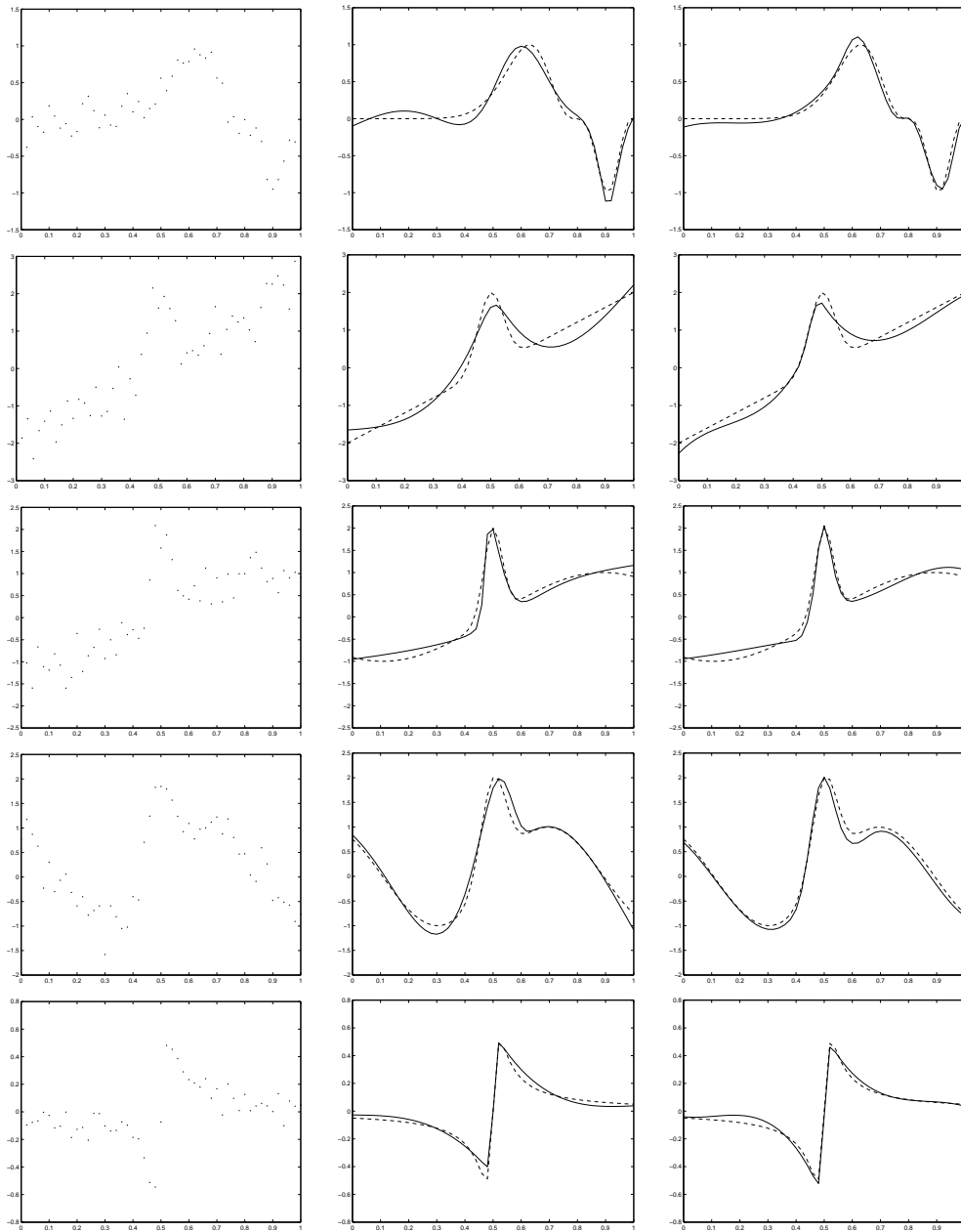


Figura 3.3: Curvas ajustadas correspondientes la 50^a mejor estimación para la configuración experimental con 50 muestras y SNR=3. Las columnas corresponden de izquierda a derecha a los datos de entrada, resultados BARS y AGJ respectivamente (línea sólida: función estimada; línea interrumpida: función real). Las filas corresponden de arriba hacia abajo a las funciones de prueba 1 a 5.

3.2. Aproximación de superficies de producto tensorial de B-splines

3.2.1. Definición del problema de aproximación de superficies

El problema de aproximación de superficies se puede describir de la siguiente manera: Dado un conjunto de mediciones $z_{i,j}$ en un dominio rectangular $i = 1, \dots, N_x, j = 1, \dots, N_y$ expresado en la siguiente forma:

$$z_{i,j} = f(x_i, y_j) + \epsilon_{i,j} \quad (3.10)$$

donde $f(x, y)$ es una superficie desconocida la cual deseamos aproximar, el término $\epsilon_{i,j}$ representa los errores de medición y $z_{i,j}$ es la muestra tomada en (x_i, y_i) . En base a lo anterior, el problema consiste en encontrar la mejor aproximación de la función $f(x, y)$. Para lo cual en este trabajo de investigación asumimos que f representa una superficie suave que puede ser bien aproximada en el intervalo $[a, b] \times [c, d]$ por un producto tensorial de funciones base B-spline.

3.2.2. Aproximación de superficies mediante un AGJ

Basándonos en un AGJ podemos determinar de manera simultanea el número y posición de los nodos, así como los coeficientes del producto tensorial de las funciones base B-spline minimizando una función de aptitud propuesta para tal fin. Las principales aportaciones del método propuesto en esta aplicación son: primero, la codificación del cromosoma que representa las soluciones potenciales del problema; segundo, la función de aptitud propuesta; tercero la configuración del AGJ.

Condificación del cromosoma bidimensional

En esta propuesta usamos una cadena binaria de longitud fija para representar el número y la posición de los nodos interiores en ambas dimensiones $\{u_1, \dots, u_m\}$, $\{v_1, \dots, v_n\}$, y una matriz de números reales para representar los coeficientes P del producto tensorial de las funciones base B-spline. Además, asumimos que u_i es un subconjunto de los puntos de diseño x_i y que v_i es un subconjunto de los puntos de diseño y_i . De esta manera, el número

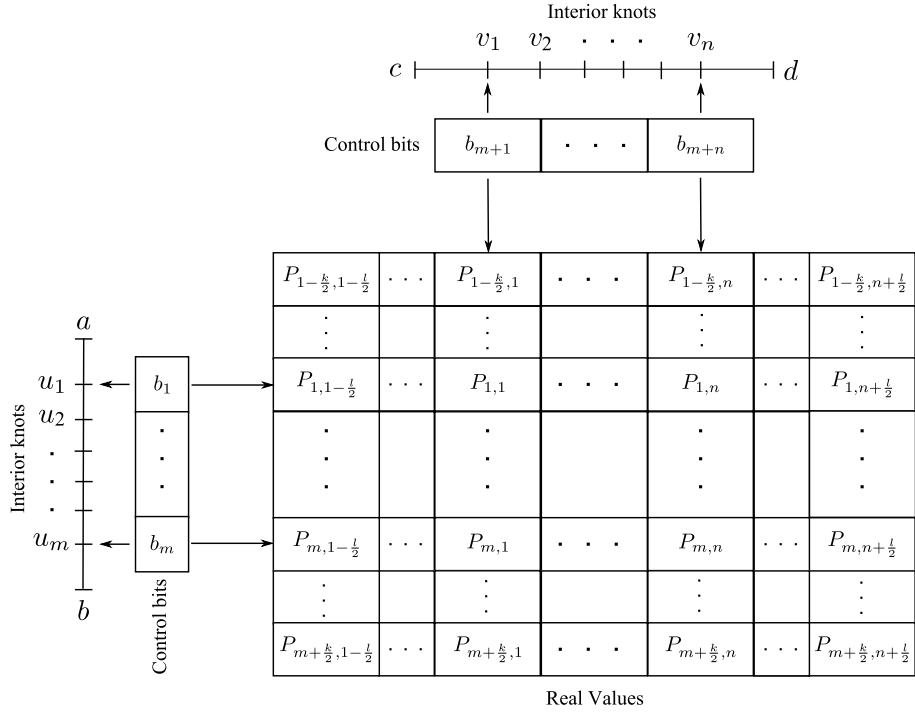


Figura 3.4: Estructura general del cromosoma bidimensional

máximo m de nodos interiores es igual al número de puntos x_i en el intervalo (a, b) y el número de coeficientes es $m + k$, así como también, el número máximo n de nodos interiores es igual al número de puntos y_i en el intervalo (c, d) y el número de coeficientes es $n + l$.

Dicho lo anterior, podemos representar el cromosoma de un individuo como:

$$\theta = \{b_1, \dots, b_m, b_{m+1}, \dots, b_{m+n}, r_{1-k/2, 1-l/2}, \dots, r_{1-k/2, n+l/2}, \dots, r_{m+k/2, 1-l/2}, \dots, r_{m+k/2, n+l/2}\}$$

donde cada b_i es un bit de control y $r_{i,j}$ es un valor real (coeficiente). Aquí, cada bit de control habilita o inhibe un nodo interior en cada dirección, y la combinación de los dos habilita o inhibe un coeficiente simultáneamente. Así se establece un correspondencia entre los nodos interiores y los coeficientes a ser activados al mismo tiempo. Los valores reales representan los coeficientes del producto tensorial de las funcione base B-spline. La estructu-

ra general del cromosoma bidimensional se muestra gráficamente en la figura 3.4.

Finalmente, el número de nodos interiores m y n está limitado por el número de puntos en cada dimensión de la superficie. Sin embargo, en [8, 13], los autores recomiendan colocar un nodo cada tres puntos de diseño x_i , esto es, el número de nodos interiores puede ser dividido por tres. Esto sería suficiente para asegurar que al menos un nodo esta en, o cercano a, las posiciones requeridas para capturar la topología de la superficie.

Función de aptitud

Para evaluar la aptitud de cada individuo θ , la función de aptitud F esta formulada como una suma de tres términos y es dada por la siguiente ecuación:

$$F(\theta) = \omega_1 RSS + \omega_2 SSD + \omega_3 PKS \quad (3.11)$$

donde cada término de la ecuación es descrito como sigue:

- (a) El primer término es la suma de residuos al cuadrado (RSS por sus siglas en inglés). Éste es usado como una medida de la desviación entre los datos observados y la función estimada \hat{f} . El RSS es calculado como sigue:

$$RSS = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \{z_{i,j} - \hat{f}(x_i, y_j)\}^2 \quad (3.12)$$

- (b) El segundo término es la suma de las diferencias cuadradas (SSD por sus siglas en inglés), la cual es una aproximación discreta del gradiente. Este término es usado para penalizar la suma alta de gradientes para generar soluciones suaves. El SSD está dado por la siguiente ecuación:

$$SSD = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \{[\hat{f}(x_i, y_j) - \hat{f}(x_{i-1}, y_j)]^2 + [\hat{f}(x_i, y_j) - \hat{f}(x_i, y_{j-1})]^2\} \quad (3.13)$$

- (c) El último término es una función de penalización para la estructura de los nodos (PKS por sus siglas en inglés). Esta es calculada de acuerdo a:

$$PKS = \sum_{i=0}^m \frac{b-a}{(u_{i+1} - u_i)^2} + \sum_{i=0}^n \frac{d-c}{(v_{i+1} - v_i)^2} \quad (3.14)$$

En la ecuación 3.11, el término PKS es utilizado para favorecer soluciones con una distribución uniforme de nodos para cada dimensión. En otras palabras, penaliza soluciones con nodos muy cercanos, los cuales generan un sobre ajuste de la función. Por lo tanto, los individuos con pocos nodos y mejores distribuciones son favorecidos.

Configuración del AGJ

El AGJ es configurado de la siguiente manera: como operador de selección se utilizó el método de la ruleta con escalamiento sigma para mantener controlada la presión de selección durante el proceso de evolución. El método de selección es combinado con elitismo, el cual toma al mejor individuo de la población actual y lo copia sin cambio a la siguiente, esto con el fin de no perder la mejor solución encontrada en ningún momento por efecto de los operadores de cruce y mutación. Para evolucionar el cromosoma codificado en binario se utiliza el operador de cruce uniforme y la mutación de bit los cuales son aplicados con una probabilidad específica. Por otra parte para evolucionar el cromosoma con codificación real se utiliza el operador de cruce SBX y el método de mutación delta. Estos últimos aplicados con las mismas probabilidades de cruce y mutación que se utilizan para los operadores del cromosoma binario.

Todos los parámetros necesarios para la ejecución del AGJ como lo son el tamaño de la población, la probabilidad de cruce y la probabilidad de mutación son sintonizados de acuerdo con el procedimiento descrito en la siguiente sección.

Sintonización de los parámetros de control del AGJ

Se llevaron a cabo una serie de experimentos con el fin de sintonizar los parámetros de control del AGJ. Para realizar la sintonización se ejecutó el AGJ sobre 100 conjuntos de datos ruidosos generados a partir de una función de prueba. Los conjuntos de datos de prueba fueron generados como se establece en la sección 3.2.3. Para analizar el proceso de convergencia, el valor de aptitud del mejor individuo es almacenado por 1500 generaciones para cada conjunto de datos probado.

Primero, analizamos la variación de la probabilidad de mutación del AGJ mientras todos los otros parámetros se mantienen fijos. En la figura 3.5, podemos ver los resultados obtenidos de la variación de la probabilidad de

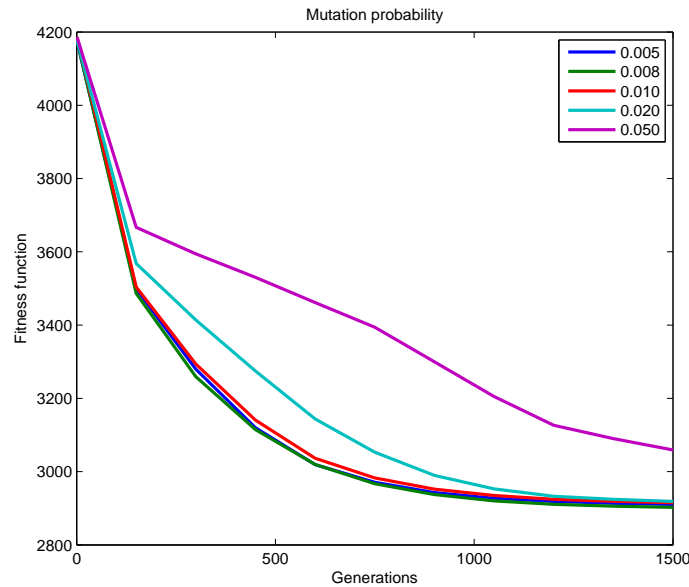


Figura 3.5: Variación de la probabilidad de mutación. Cada línea muestra el promedio de 100 conjuntos de datos de prueba.

mutación de 0.005 hasta 0.050. Como podemos ver en la misma figura, el AGJ se desempeña mejor con una probabilidad de mutación de 0.008 (línea en color verde). Es claro que mientras la probabilidad de mutación incrementa el desempeño del AGJ decrece.

Por otra parte, variamos el tamaño de la población y analizamos sus efectos en el proceso de convergencia del AGJ. La figura 3.6 muestra el desempeño del AGJ para varios tamaños de población. Podemos ver que el desempeño del algoritmo es casi similar para los tamaños de población de 90, 120 y 200 (líneas en color rojo, cyan y magenta respectivamente), pero el tiempo promedio de ejecución para cada tamaño de población que se muestra en la tabla 3.6 es diferente. Por lo tanto, la mejor opción para el tamaño de población es de 90 individuos.

Finalmente, en la figura 3.7 se muestra el desempeño del AGJ para 3000 generaciones. En esta figura, se presentan las 50 mejores corridas del AGJ. Como puede verse en esta misma figura, la convergencia del AGJ se alcanzada en la generación 1500.

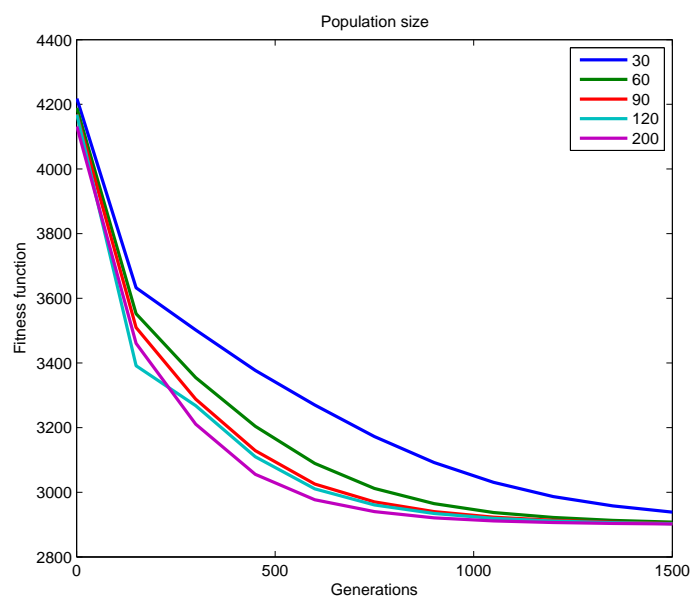


Figura 3.6: Variación del tamaño de población. Cada línea muestra el promedio de 100 conjuntos de datos de prueba.

Tabla 3.6: Tiempo promedio de ejecución del AGJ.

Tamaño de población	Tiempo (s)
30	34
60	68
90	103
120	136
200	226

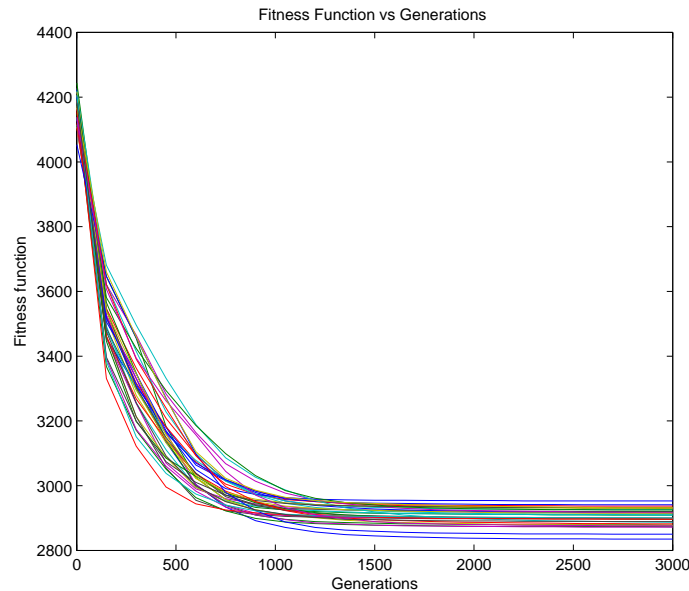


Figura 3.7: Evolución del mejor individuo para las 50 mejores pruebas.

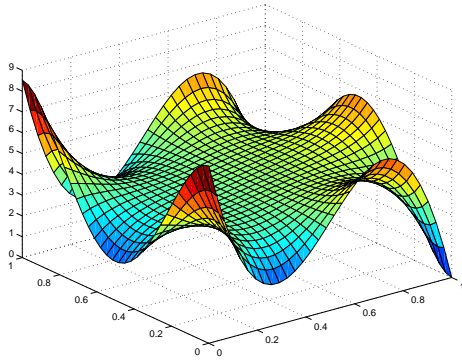
3.2.3. Pruebas y resultados de la aproximación de superficies

Para evaluar el desempeño de esta aplicación se llevaron a cabo varias simulaciones. Para lo cual se definió una configuración experimental de cuatro funciones bi-variadas, cuyas ecuaciones están dadas en la tabla 3.7 y se muestran de forma gráfica en la figura 3.8. Estas funciones de prueba fueron tomadas de trabajos previos [12, 14, 74, 75] como referencia para validar el algoritmo propuesto. También es presentada una comparación con el método de Suavizado Ponderado Local (LOWESS por siglas en inglés) [76].

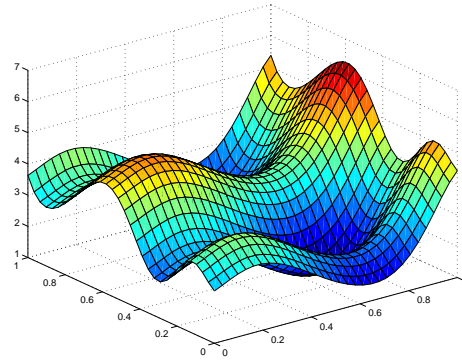
Para esta aplicación, la configuración experimental fue definida para cada función de prueba como sigue: la función es evaluada en 1024 puntos que forman una retícula rectangular de 32×32 sobre el intervalo $[0, 1] \times [0, 1]$. Entonces, los datos son trasladados para hacer el rango no negativo con el fin de facilitar la comparación entre ellas. Posteriormente, ruido normal con media cero y σ conocida es añadido al conjunto de datos. Un ejemplo de los conjuntos de datos ruidosos generados para las cuatro funciones de prueba son mostrados en la figura 3.10. Además, para cada configuración

Tabla 3.7: Funciones bi-variadas de prueba.

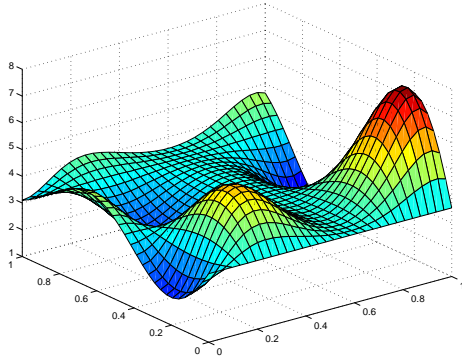
No.	Función
1	$f(x, y) = 42,659\{0,1 + \hat{x}(0,05 + \hat{x}^4 - 10\hat{x}^2\hat{y}^2 + 5\hat{y}^4)\}, \hat{x} = x - 0,5, \hat{y} = y - 0,5$
2	$f(x, y) = 1,3356[1,5(1 - x) + e^{(2x-1)}\sin\{3\pi(x - 0,6)^2\} + e^{(3(y-0,5))}\sin\{4\pi(y - 0,9)^2\}]$
3	$f(x, y) = 1,9[1,35 + e^x \sin\{13(x - 0,6)^2\}e^{-y} \sin(7y)]$
4	$f(x, y) = 3 + 2\sin(2\pi x)\sin(2\pi y) + e^{\{-15(x-1)^2 - 15(y-1)^2\}} + e^{\{-10x^2 - 15(y-1)^2\}}$



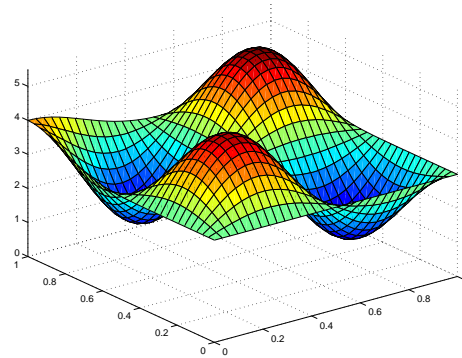
(a) Function 1



(b) Function 2



(c) Function 3



(d) Function 4

Figura 3.8: Gráficas de las funciones de prueba.

experimental, una colección de 100 conjuntos de datos ruidosos es generada con tres SNR diferentes 2, 3 y 4, respectivamente. Por lo que esta aplicación es probada sobre un total de 1200 conjuntos de datos para las cuatro funciones. Una vez que se tiene todo el conjunto experimental, el método propuesto y el método LOWESS fueron aplicados para aproximar las funciones de prueba.

En las pruebas numéricas, nuestro algoritmo fue configurado como sigue: usamos una B-spline cúbica como base de funciones, esto es, $k = 4$, $l = 4$ y los nodos interiores son definidos como un subconjunto de los puntos de diseño ($m_{max} = 32, n_{max} = 32$). La población fue inicializada aleatoriamente al inicio del algoritmo. Cada gen de control b_i es seleccionado en forma aleatoria de $[0, 1]$ y cada gen paramétrico r_i es calculado como un número real definido sobre el rango $[\min(z_{i,j}), \max(z_{i,j})]$ de las mediciones $z_{i,j}$. Los parámetros del AGJ para esta aplicación fueron sintonizados experimentalmente como se estableció en la sección 3.2.2 y son mostrados en la tabla 3.8. La población fue evolucionada durante 3000 generaciones en todos los casos.

Tabla 3.8: Parámetros de control sintonizados.

Parámetro	Valor
Tamaño de Población	90
Probabilidad de Cruza	0.85
Probabilidad de Mutación	0.008
Número de Individuos Elite	9

Para efectos de comparación, el método LOWESS se aplicó sobre los mismos conjuntos de datos de prueba. Para esto, utilizamos el “Curve Fitting Toolbox” de MATLAB. En esta simulación, los parámetros por default recomendados fueron considerados. Para comparar los resultados también utilizamos el error cuadrático promedio (MSE) dado por:

$$MSE = \frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \{\hat{f}(x_i, y_j) - f(x_i, y_j)\}^2 \quad (3.15)$$

donde \hat{f} es la función estimada dada por cada método aplicado y f es la función real. El valor de MSE es calculado y registrado para cada una de las 100 pruebas para las cuatro funciones. Los diagramas de caja de los valores de $MSE(\hat{f})$ son mostrados en la figura 3.9.

En la figura 3.9, para cada combinación de función de prueba y SNR existe un panel, en el cual se pueden comparar los diagramas de caja de

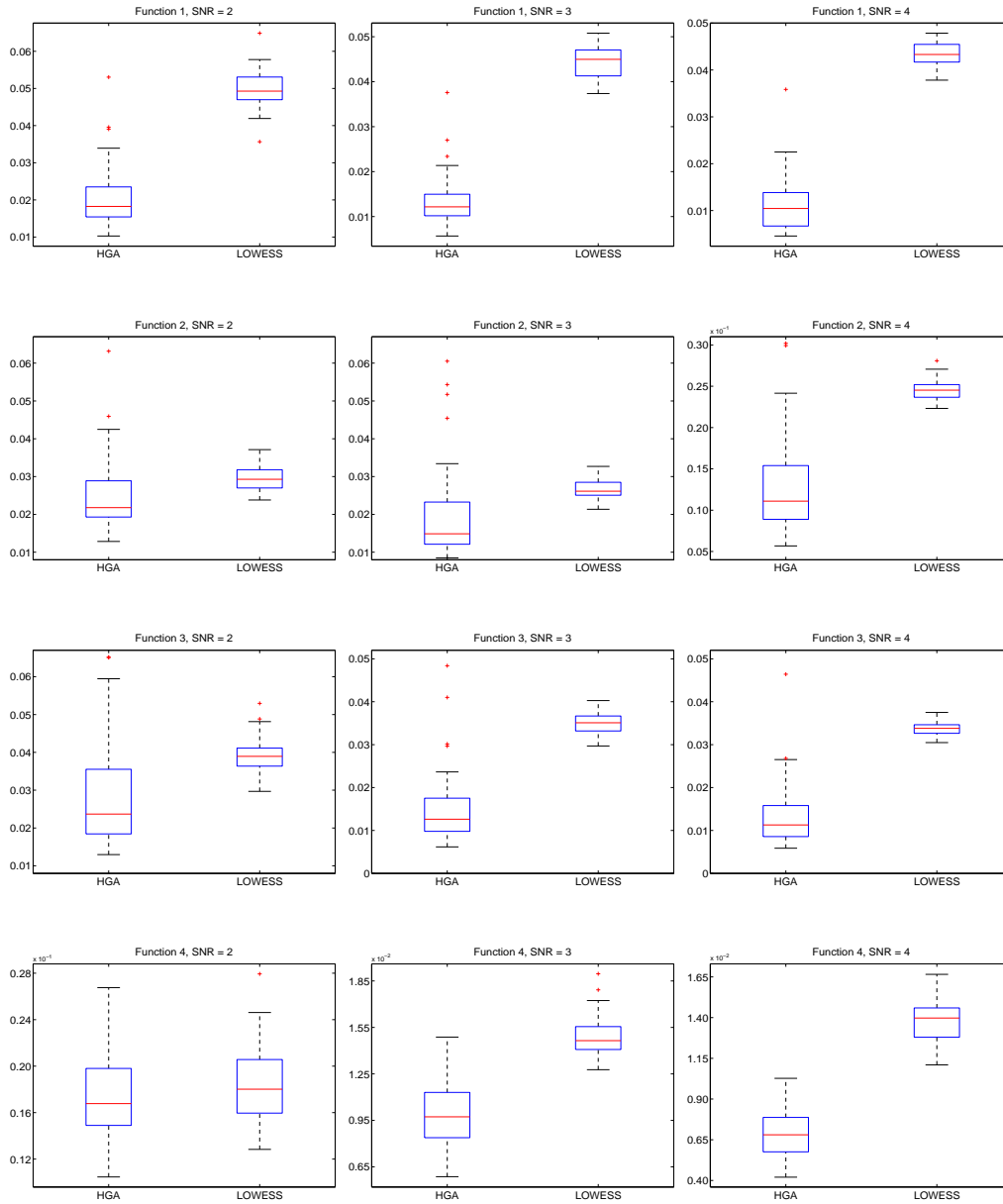


Figura 3.9: Diagramas de caja de los valores de MSE obtenidos por los métodos probados.

los valores de MSE para el AGJ y el método LOWESS. Las columnas de paneles corresponden de izquierda a derecha a niveles de SNR de 2, 3 y 4 respectivamente, de tal forma que el nivel de ruido añadido a la señal decrece de izquierda a derecha. Las filas de paneles corresponden de arriba a abajo a las funciones de prueba de 1 a 4. Como se puede apreciar en los diagramas de caja, en la mayoría de los casos los valores de MSE producidos por el AGJ son considerablemente más bajos a comparación de los generados por el método LOWESS. En algunos casos el cuartil superior de los valores de MSE producidos por el AGJ están por debajo del primer cuartil de los generados por el método LOWESS, lo cual implica que el 75 % de los valores de MSE estimados por el AGJ son menores que el 75 % de los valores de MSE producidos por LOWESS. La mayor diferencia se puede observar en la figura 3.9 para la función de prueba 4 con $SNR = 4$, donde el cuartil superior del AGJ está por muy por debajo del cuartil inferior del método LOWESS. Además la mediana de los valores de MSE producidos por el AGJ es virtualmente menor para todos los casos probados.

En cuanto a la extensión de la mitad central (la caja) de los valores de MSE, esta parece ser menos variable y simétrica al rededor de la mediana en algunos de los resultados del método de LOWESS, contrario a aquellos producidos por el AGJ. Por otra parte, aunque ambos algoritmos presentan valores de MSE atípicos, el método LOWESS muestra menos valores atípicos que el AGJ. Sin embargo, aunque el método LOWESS presenta mejores resultados en cuanto a las estas características mencionadas, el AGJ mejora en gran medida de manera global los resultados producidos por el método LOWESS.

Aparte de los diagramas de caja de los valores de MSE, también calculamos la media y el error estándar de la media de los valores de MSE para ambos métodos, los cuales se muestran en la tabla 3.9. De esta tabla podemos ver que el AGJ alcanza los menores valores de MSE en todos los experimentos. Particularmente, podemos ver la mayor diferencia estadística significativa en la función de prueba 4 y $SNR = 4$, donde la media de los valores de MSE para AGJ esta muy alejada del valor de la media para los valores de MSE producidos por el método LOWESS, la cual es marcada en negritas. En cuanto a los errores estándar, aunque el método LOWESS presenta valores menores que el AGJ, la diferencia entre las medias es significativa, por lo que a pesar de esto, los resultados del AGJ son mejores.

Para evaluar cualitativamente los resultados del método propuesto, un ejemplo obtenido para cada función es mostrado gráficamente en la figura

Tabla 3.9: Media con error estándar de los valores de MSE obtenidos sobre 100 conjuntos de prueba.

Prueba	Método	SNR=2	SNR=3	SNR=4
1	LOWESS	0.049889 ± 0.000722	0.044644 ± 0.000461	0.043435 ± 0.000349
	AGJ	0.020792 ± 0.001195	0.013381 ± 0.000757	0.011380 ± 0.000824
2	LOWESS	0.029555 ± 0.000464	0.026570 ± 0.000325	0.024656 ± 0.000180
	AGJ	0.028086 ± 0.002732	0.025059 ± 0.004591	0.017700 ± 0.002964
3	LOWESS	0.039298 ± 0.000621	0.035072 ± 0.000337	0.033791 ± 0.000218
	AGJ	0.030230 ± 0.002457	0.014920 ± 0.001158	0.013911 ± 0.001307
4	LOWESS	0.018480 ± 0.000434	0.014860 ± 0.000172	0.013765 ± 0.000163
	AGJ	0.017167 ± 0.000538	0.009926 ± 0.000295	0.006886 ± 0.000221

3.10, junto con un ejemplo de los datos ruidosos simulados. De esta misma figura, podemos apreciar que el AGJ tiene la habilidad para adaptarse de manera óptima a las superficies probadas.

En resumen, podemos asumir que el método presentado muestra resultados sobresalientes al aproximar las superficies. Por otra parte, su principal desventaja nuevamente es el costo computacional requerido por el AGJ en comparación con el método LOWESS. El tiempo promedio para realizar las aproximaciones con el método de LOWESS es de 4 segundos, mientras que para el AGJ el tiempo promedio es de 201 segundos. Sin embargo, esta problemática se puede mejorar utilizando técnicas de computación en paralelo para la creación de la aplicación. Este algoritmo fue implementado en lenguaje C++. Ambos métodos fueron ejecutados en una PC usando un procesador AMD Phenom II a 1.9 GHz con 4 GB de memoria RAM.

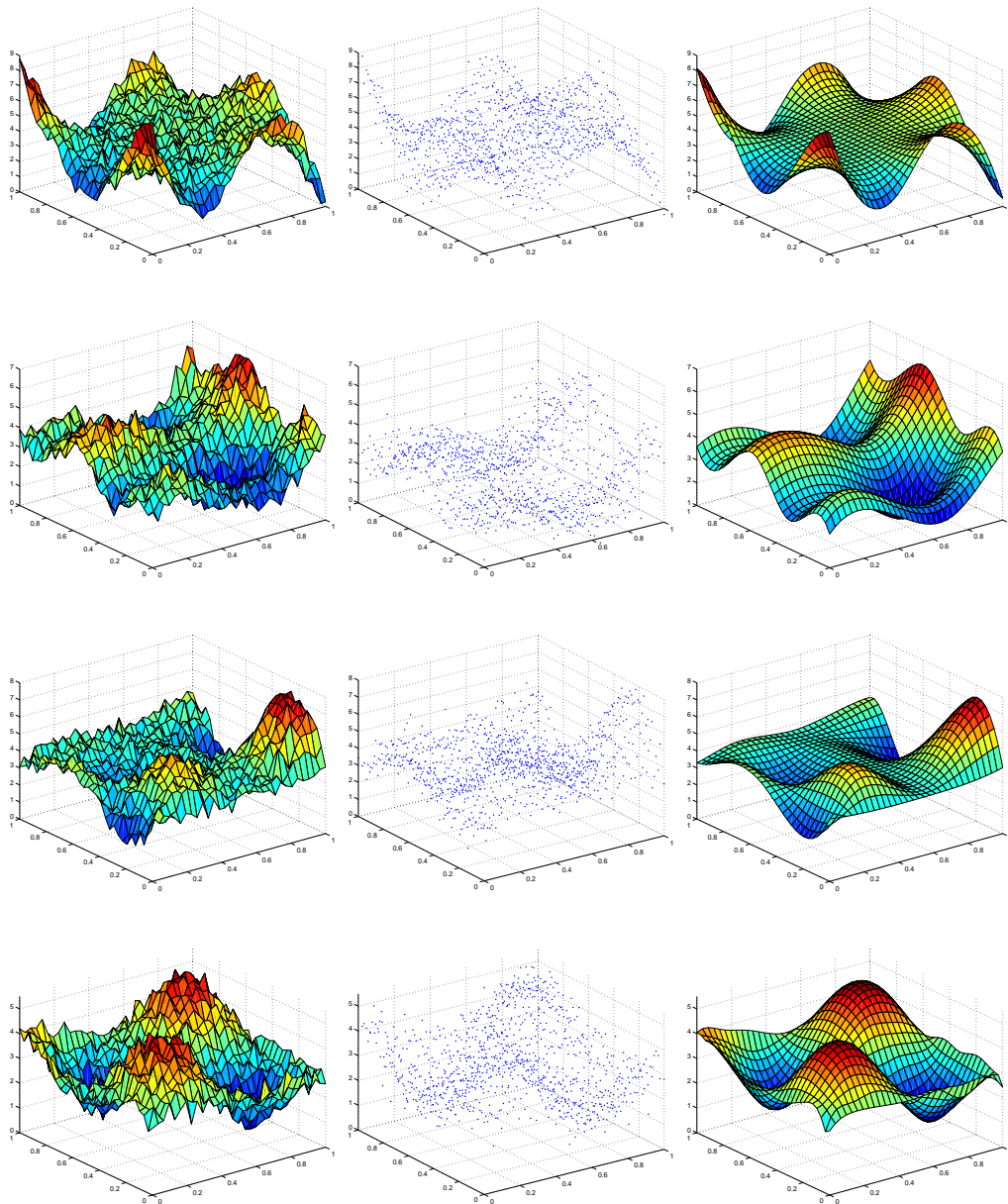


Figura 3.10: Resultados para el AGJ. Las columnas corresponden de izquierda a derecha a: superficie ruidosa de entrada, nube de datos de entrada, superficies aproximadas. Las filas corresponde de arriba a abajo a las funciones de prueba de 1 a 4.

3.3. Demodulación de patrones de franjas

3.3.1. Definición del problema

En metrología óptica un patrón de franjas se puede representar mediante la siguiente expresión:

$$I(x, y) = a(x, y) + b(x, y) \cos(\phi(x, y)) + \eta(x, y) \quad (3.16)$$

donde $a(x, y)$ representa la iluminación de fondo, $b(x, y)$ es la amplitud de la modulación, $\phi(x, y)$ es el término de fase y $\eta(x, y)$ representa el ruido del sistema. Aquí x e y representan los pixeles a lo largo de las dimensiones horizontal (columna) y vertical (fila) para un patrón de franjas $N_x \times N_y$, tenemos que $x \in [0, N_x - 1]$ y $y \in [0, N_y - 1]$.

El problema consiste entonces en determinar el término de fase el cual está relacionado a la magnitud física que se desea medir. Este problema no es sencillo de resolver debido a los siguientes factores: presencia del ruido, franjas cerradas, franjas discontinuas y el factor que puede ser el más complicado causado por la naturaleza de la función coseno la cual presenta ambigüedad del signo además de que la fase se presenta en módulos de 2π . Debido a todos los factores mencionados este es un problema desafiante del área de la metrología óptica [77, 78].

3.3.2. Demodulación de patrones de franjas usando funciones de base radial y un AG

En esta aplicación se propone resolver el problema de la demodulación de franjas haciendo uso de una base de funciones conformada por funciones de base radial y un AG para la optimización de los parámetros de las funciones base.

Considerando que una superficie cualquiera se puede expresar como una combinación lineal de funciones de base radial dadas de acuerdo con la ecuación 2.8. Para esta aplicación en particular se seleccionó el uso de la función de base radial tipo Gaussiana para representar dicha superficie, la cual queda definida como:

$$f(x, y) = \sum_{i=1}^N \lambda_i e^{-\frac{((x-c_{xi})^2 + (y-c_{yi})^2)}{\sigma_i^2}} \quad (3.17)$$

donde N es el número de funciones de base radial que se utilizan para representar la superficie, (c_{xi}, c_{yi}) representan las coordenadas del centro, σ_i el radio y λ_i la amplitud de la i -ésima función de base radial.

Por lo que se propone el uso de un AG para optimizar los centros, radios y amplitudes de las funciones de base radial de tal forma que minimicen una función de aptitud que permita encontrar la fase correspondiente a un patron de franjas dado. El algoritmo propuesto se detalla en las siguientes secciones.

Codificación del Cromosoma

Para representar cada individuo como una posible solución al problema de minimización de la función de aptitud, cada individuo contiene un cromosoma θ con codificación real compuesto por las amplitudes λ_i , los centros (c_{xi}, c_{yi}) y los radios σ_i de cada función de base radial, esto es, el cromosoma esta representado por un vector de genes tal que la cantidad de parámetros sea igual al total de funciones de base radial a utilizar en la búsqueda de la fase.

$$\theta = \{\lambda_1, \lambda_2, \dots, \lambda_N, c_{x1}, c_{x2}, \dots, c_{xN}, c_{y1}, c_{y2}, \dots, c_{yN}, \sigma_1, \sigma_2, \dots, \sigma_N\} \quad (3.18)$$

Cada gen del cromosoma es codificado por un valor de tipo real, de tal forma que el cromosoma forma un vector de $4 * N$ valores reales, donde N es el número de funciones de base radial usado, como esta dado en la expresión de la ecuación 3.18.

Función de aptitud

Para realizar la demodulación se propone encontrar una estimación de la fase representada mediante una combinación lineal de funciones de base radial mediante la minimización de la siguiente función de adaptación:

$$F(\theta) = \sum_x \sum_y \left\{ I(x, y) - \cos \left(\sum_{i=1}^N \lambda_i e^{-\frac{((x-c_{xi})^2 + (y-c_{yi})^2)}{\sigma_i^2}} \right) \right\}^2 \quad (3.19)$$

donde $I(x, y)$ es el patrón de franjas original normalizado, y el segundo término es el coseno de la función ajustada para aproximar el término de fase. Esta función de aptitud minimiza la diferencia entre el patrón de franjas de entrada y el patron de franjas calculado a partir del término de fase aproximado en función de los parámetros del cromosoma de cada individuo.

Configuración del AG

El algoritmo genético es configurado de la siguiente manera: se utiliza un cromosoma con codificación real, como operador de selección se utiliza el método de la ruleta con escalamiento sigma para controlar la presión de selección. Como operadores genéticos se utilizan el operador de cruce uniforme y el operador de mutación delta; para evitar perder la mejor solución encontrada, cada generación el mejor individuo es copiado sin cambio a la siguiente generación, el cual es conocido como individuo elite. Los operadores genéticos son aplicados con una probabilidad específica de cruce y mutación.

Todos los parámetros del AG son sintonizados experimentalmente siguiendo el procedimiento descrito en la sección 3.2.2. El valor de los parámetros sintonizados se muestran en la tabla 3.10.

Tabla 3.10: Parámetros de control del AG.

Parámetro	Valor
Tamaño de Población	300
Probabilidad de Cruza	0.80
Probabilidad de Mutación	0.05
Número de Individuos Elite	1

3.3.3. Pruebas y resultados de la demodulación de patrones de franjas

Para probar el algoritmo de demodulación de patrones de franjas se realizaron dos tipos de pruebas. La primer prueba comprueba el desempeño del algoritmo propuesto para la demodulación de patrones de franjas complejos, los cuales contienen franjas cerradas. Dichos patrones de franjas son simulados tomando como base funciones de prueba que aparecen en [79]. En la segunda prueba se pone a prueba el desempeño del algoritmo ante patrones de franjas simulados con una cantidad de ruido añadido, esta misma prueba se realiza en [80] para efectos de comparación.

Para realizar las pruebas el algoritmo fue ejecutado de la siguiente manera: se define el cromosoma para manejar individuos conformados de 20 funciones de base radial de tipo Gaussiano, esto es $N = 20$, la población es inicializada de forma aleatoria al inicio del algoritmo. Todos los valores son inicializados con un número real aleatorio calculado entre un intervalo

definido de forma manual de acuerdo a las condiciones del patron de franjas de entrada. Los parámetros de control del AG para esta aplicación son los mostrados en la tabla 3.10. La población es evolucionada durante 1200 generaciones en todos los casos.

Pruebas de demodulación de patrones de franjas cerradas

Para realizar estas pruebas se simularon tres patrones de franjas tomados de [79] cuyas funciones se muestran en la tabla 3.11. Las imágenes generadas fueron de 512x512, las cuales fueron sub-muestreadas a un tamaño de 32x32 puntos. Los patrones de franjas simulados se muestran en la figura 3.11.

Tabla 3.11: Funciones utilizadas para la simulación de los patrones de franjas.

No.	Función
1	$\phi(i, j) = 25 * \exp \left[\frac{-(i-128)^2 - (j-128)^2}{120^2} \right] + 25 * \exp \left[\frac{-(i-384)^2 - (j-384)^2}{120^2} \right]$
2	$\phi(i, j) = 0,188 * (j - 256) * \ln \left[\frac{(i-256)^2 + (j-256)^2}{246^2} \right]$
3	$\phi(x, y) = 38 * \exp \left[- \left(\frac{(x-5)^2}{2,5} + \frac{(y-5)^2}{10} \right) \right]$

Para evaluar el desempeño del algoritmo propuesto se calcula el valor de RMSE entre la fase real y la fase estimada por el algoritmo propuesto. Los resultados obtenidos se muestran en la tabla 3.12. Como se puede apreciar los valores de RMSE son bajos, lo cual muestra que los resultados obtenidos por el algoritmo son buenos.

Tabla 3.12: Resultados de la demodulación de patrones de franjas cerradas

Interferograma No.	RMSE de la fase estimada
1	0.025
2	0.039
3	0.037

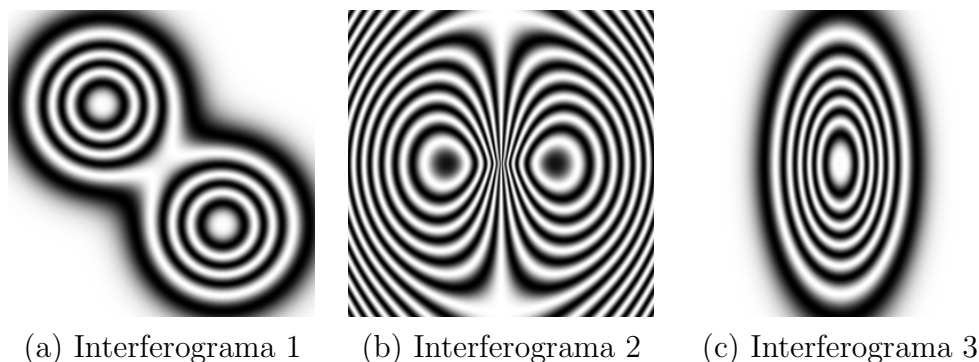


Figura 3.11: Patrones de franjas simulados.

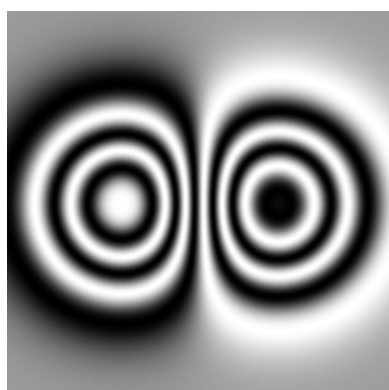
Pruebas de desempeño ante el ruido

Para realizar las pruebas de desempeño del algoritmo ante el ruido, se utilizó la misma configuración experimental descrita en [80]. Donde se simula un patrón de franjas utilizando MATLAB, al cual se le añade ruido blanco Gaussiano con media cero y desviación estándar σ conocida. Los patrones de franjas fueron generados en un tamaño de 512x512. Para efectos de comparación se obtiene el RMSE de la fase original contra la fase estimada por el AG. Los resultados obtenidos se resumen en la tabla 3.13 donde también se muestran los resultados obtenidos por el algoritmo descrito en [80].

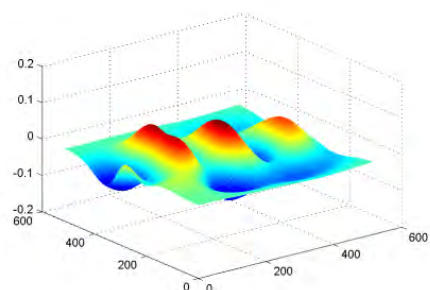
Tabla 3.13: Resultados obtenidos para varios niveles de ruido.

Prueba	Ruido(σ)	AG (RMSE)	Operador Diferencial 2D (RMSE)
1	sin ruido	0.0291	—
2	0.1	0.0300	0.0310
3	0.2	0.0279	0.0328
4	0.3	0.0392	0.0537
5	0.4	0.0423	0.0776

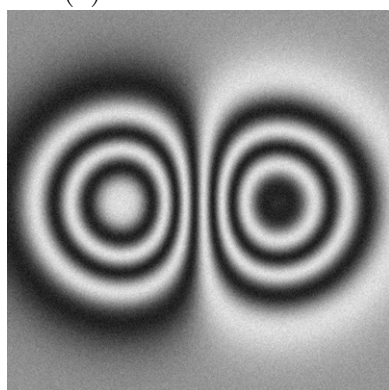
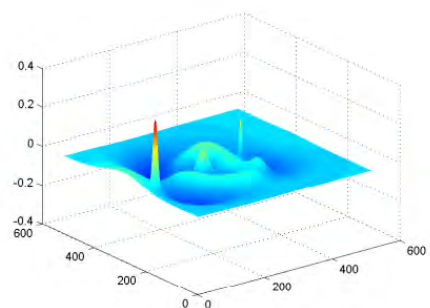
Como se puede apreciar en los valores de RMSE obtenidos por el AG, se obtienen excelentes resultados sobre todo al incrementarse la cantidad de ruido en la imagen del interferograma simulado en comparación del método propuesto en [80]. En estas pruebas se utilizó como entrada el interferograma completo, es decir, no se utilizó un sub-muestreo, lo cual provoca una mejora de los resultados en gran medida, pero por otro lado consume gran cantidad de tiempo para su evolución. En promedio para una imagen de 512x512



(a) Patron sin ruido



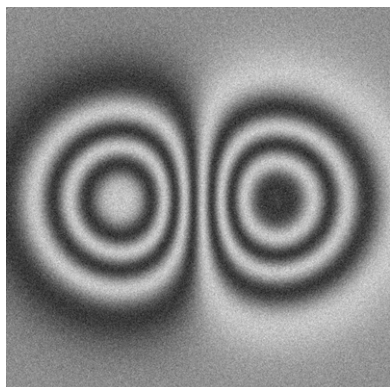
(b) Superficie de error

(c) Patron con ruido ($\sigma = 0,1$)

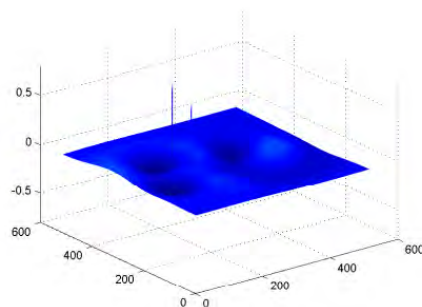
(d) Superficie de error

Figura 3.12: Patrones de franjas de entrada y superficies de error del resultado

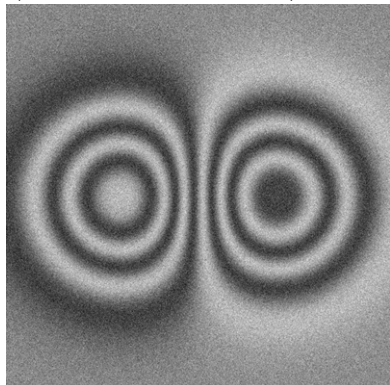
el tiempo aproximado de cómputo es de 5 hrs, lo cual representa una gran desventaja, aunque la misma se puede disminuir si consideramos la implementación en paralelo del AG. En las figuras 3.12 y 3.13 se muestran los interferogramas simulados junto con la superficie de error obtenida para las cinco pruebas mostradas en la tabla 3.13.



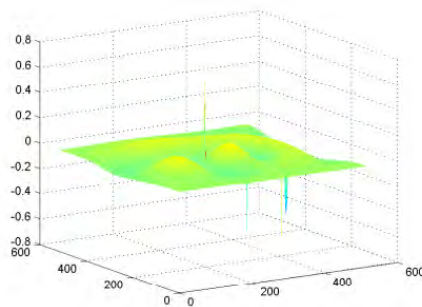
(a) Patron con ruido ($\sigma = 0,2$)



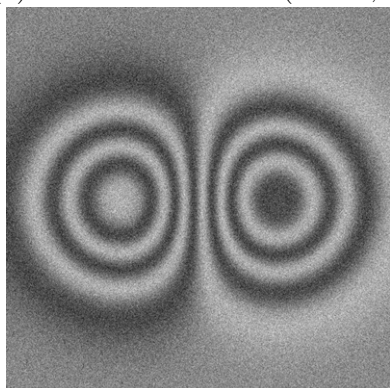
(b) Superficie de error



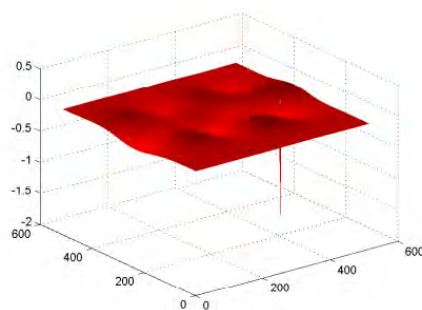
(c) Patron con ruido ($\sigma = 0,3$)



(d) Superficie de error



(e) Patron con ruido ($\sigma = 0,4$)



(f) Superficie de error

Figura 3.13: Patrones de franjas de entrada y superficies de error del resultado

Capítulo 4

Conclusiones y Perspectivas

4.1. Conclusiones

En conclusión se aplicaron técnicas de cómputo evolutivo, específicamente algoritmos genéticos convencionales y jerárquicos, en la solución de problemas en el área de la óptica e ingeniería. Se desarrollaron tres aplicaciones, un algoritmo de ajuste de curvas usando funciones B-spline, un algoritmo de aproximación de superficies de producto tensorial de funciones B-spline y la demodulación de imágenes con patrones de franjas cerradas usando funciones de base radial.

Los algoritmos presentados, son robustos y genéricos; utilizan algoritmos genéticos para la resolver los distintos problemas planteados en forma de problemas de optimización. Los resultados experimentales muestran que los algoritmos genéticos son una buena propuesta para afrontar este tipo de problemas, como se demostró en este trabajo, y otros donde se han utilizado estas mismas técnicas con otros tipos de codificaciones de individuos y parámetros a optimizar.

Cada una de las aplicaciones muestra un desempeño sobresaliente en distintos aspectos como se describe continuación:

Ajuste de Curvas. El método propuesto de ajuste de curvas B-spline usando un algoritmo genético jerárquico es capaz de buscar de manera simultanea el número y la posición de los nodos así como también los coeficientes de la curva B-spline, optimizando dichos parámetros, por lo cual es apto de encontrar soluciones con la menor cantidad de parámetros necesarios, al tiempo que minimiza el error de la función estimada en el sentido de los mínimos

cuadrados. Además, como se muestra en los resultados esta aplicación presenta gran robustez ante la presencia del ruido y la disminución del número de muestras en la entrada de los datos.

Aproximación de Superficies. En cuanto a la aproximación de superficies de producto tensorial de funciones base B-spline el cual es un problema sumamente complejo por la cantidad de parámetros necesarios para definir una superficie. Nuestro método optimiza el número y la posición de los nodos en cada una de las dimensiones de la superficie, al mismo tiempo que optimiza los coeficientes del producto tensorial de las B-splines. Tan solo considerando que el cálculo de los coeficientes es un problema que no tiene solución por métodos convencionales dado que el sistema que se forma del producto tensorial de las B-splines es no lineal, nuestro algoritmo demuestra ser una excelente propuesta de solución muy robusta la cual es capaz de optimizar todos los grados de libertad del producto tensorial de las B-spline produciendo excelentes aproximaciones aun en la presencia de datos con ruido.

Demodulación de patrones de franjas. Dentro de la aplicación de la demodulación de patrones de franjas usando funciones de base radial, el algoritmo demostró excelentes resultados en la recuperación de la fase a partir de imágenes que contienen patrones de franjas cerradas con varios niveles de ruido. La fase recuperada se obtiene en forma desenvuelta a partir de una sola imagen, aunque ésta contenga gran cantidad de ruido. La fase obtenida muestra un error muy bajo como lo muestran los resultados, en comparación con otras técnicas. Otra ventaja del algoritmo propuesto es que se obtiene una representación analítica de la fase, permitiendo obtener información a diferentes resoluciones.

Entre las principales aportaciones científicas de esta investigación se destacan las siguientes:

- Dos publicaciones en revista indexada con factor de impacto mayor de uno.
- La introducción de una nueva estructura genética jerárquica para la representación del cromosoma que permite optimizar de manera simultánea la estructura y parámetros del modelo de curvas B-spline.

- El empleo del modelo de islas para mejorar los resultados obtenidos por el algoritmo, permitiéndole encontrar soluciones en un área mayor de la frontera de Pareto.
- La propuesta de un cromosoma bidimensional basado en una estructura genética jerárquica que permite optimizar todos los grados de libertad del producto tensorial de funciones base B-spline.
- La propuesta de la función de aptitud basada en tres términos para la evaluación de los modelos de superficies dados por el producto tensorial de funciones base B-spline.
- El uso de funciones de base radial para la demodulación de patrones de franjas en combinación con un algoritmo genético para optimizar los parámetros de las funciones.

4.2. Perspectivas

Las aplicaciones propuestas presentan varios aspectos de interés en el desarrollo de nuevas investigaciones que den continuidad al trabajo realizado. La aplicación de otros tipos de funciones base para el ajuste de curvas y superficies, como lo son las funciones de base radial o las funciones de Gabor, es de mucho interés dado que los resultados mostrados para las dos propuestas descritas en este trabajo son destacados, lo que nos lleva a pensar que para otras funciones base se deberían de obtener resultados similares.

Respecto de la demodulación de franjas el algoritmo presenta muchos aspectos a mejorar como lo son la inicialización de la población la cual puede mejorarse si se detectan de antemano los centros de las franjas cerradas; además si se contabilizan el número de franjas se puede inicializar las amplitudes en rangos más adecuados, dándole al algoritmo una ventaja desde el inicio, lo cual implicaría una convergencia en un menor número de generaciones.

La principal desventaja que presentan la aplicaciones mostradas en este trabajo de investigación es el tiempo de cómputo necesario para la evolución de la población. En este sentido, para mejorar el tiempo de evolución del algoritmo, un aspecto de interés es el empleo de técnicas de cómputo paralelo basado en unidades centrales de procesamiento con múltiples núcleos (multi-core CPU por sus siglas en inglés) y unidades de procesamiento gráfico (GPU,

por sus siglas en inglés). De tal forma que una implementación totalmente paralela del algoritmo genético es idónea para aplicarse dada su naturaleza. De esta manera se puede disminuir en un gran porcentaje el tiempo de cómputo de las aplicaciones implementadas.

En resumen, el presente trabajo cumplió con los objetivos propuestos y las expectativas esperadas sobre el mismo. Proponiendo nuevas formas de abordar los problemas de ajuste de curvas y superficies usando funciones base B-spline, así como otra propuesta para la demodulación de patrones de franjas usando funciones de base radial. Impactando directamente en las áreas de la óptica y la ingeniería.

Información de contacto

Para solicitar más información diríjase a:

- M. I. Carlos Hugo García Capulín
e-mail: carlosg07@gmail.com
- Centro de Investigaciones en Óptica A. C.
Loma del Bosque #115, Col. Lomas del Campestre, C.P. 37150
León, Guanajuato, México.
Telefono: (477)-441 42 00

Bibliografía

- [1] S. Adarsh, R. Dhanya, G. Krishna, R. Merlin, and J. Tina. Prediction of ultimate bearing capacity of cohesionless soils using soft computing techniques. *ISRN Artificial Intelligence*, vol. 2012:10 pages, 2012.
- [2] Mehdi Zamani. A simple piecewise cubic spline method for approximation of highly nonlinear data. *Natural Science*, 4:79–83, 2012.
- [3] T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models*. Chapman & Hall CRC, 1990.
- [4] Thomas C. M. Lee. Smoothing parameter selection for smoothing splines: a simulation study. *Computational Statistics & Data Analysis*, 42(1-2):139–148, 2003.
- [5] Grace Wahba. *Spline Models for Observational Data*. SIAM [Society for Industrial and Applied Mathematics], 1990.
- [6] Ilaria Dimatteo, Christopher R. Genovese, and Robert E. Kass. Bayesian curve-fitting with free-knot splines. *Biometrika*, 88(4):1055–1071, 2001.
- [7] J. H. Friedman. Multivariate adaptive regression splines (with discussion). *Annals of Statistics*, 19(1):1–141, 1991.
- [8] J. H. Friedman and B. W. Silverman. Flexible parsimonious smoothing and additive modeling. *Technometrics*, 31:3–39, 1989.
- [9] J. Y. Koo. Spline estimation of discontinuous regression functions. *Journal of Computational and Graphical Statistics*, 6:266–284, 1997.
- [10] Thomas C. M. Lee. Regression spline smoothing using the minimum description length principle. *Statistics & Probability Letters*, 48(1):71 – 82, 2000.

- [11] Xiuyang Zhao, Caiming Zhang, Bo Yang, and Pingping Li. Adaptive knot placement using a gmm-based continuous optimization algorithm in b-spline curve approximation. *Computer-Aided Design*, 43(6):598–604, June 2011.
- [12] D. G. T. Denison, B. K. Mallick, and A. F. M. Smith. Automatic bayesian curve fitting. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, 60:333–350, 1998.
- [13] Michael Smith and Robert Kohn. Nonparametric regression using bayesian variable selection. *Journal of Econometrics*, 75(2):317–343, December 1996.
- [14] Thomas C. M. Lee. On algorithms for ordinary least squares regression spline fitting: a comparative study. *Journal of Statistical Computation and Simulation*, 72(8):647–663, 2002.
- [15] Jennifer Pittman. Adaptive splines and genetic algorithms. *Journal of Computational and Graphical Statistics*, 11(3):615–638, SEP 2002.
- [16] Garrick Wallstrom, Jeffrey Liebner, and Robert E. Kass. An implementation of bayesian adaptive regression splines (bars) in c with s and r wrappers. *Journal of Statistical Software*, 26(1):1–21, 2 2007.
- [17] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, April 1992.
- [18] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, January 1989.
- [19] Fujiichi Yoshimoto, Masamitsu Moriyama, and Toshinobu Harada. Automatic knot placement by a genetic algorithm for data fitting with a spline. In *Proceedings of the International Conference on Shape Modeling and Applications*, pages 162–169. IEEE Computer Society Press, 1999.
- [20] Li Pingping, Zhao Xiuyang, and Yang Bo. Automatic knot adjustment by an improved genetic algorithm. In *International Conference on Future Computer and Communication (ICFCC)*, volume 3, pages 763–768, May 2010.

- [21] Fujiichi Yoshimoto, Toshinobu Harada, and Yoshihide Yoshimoto. Data fitting with a spline using a real-coded genetic algorithm. *Computer-Aided Design*, 35(8):751–760, July 2003.
- [22] Amir Safari, Hirpa G. Lemu, Soheil Jafari, and Mohsen Assadi. A comparative analysis of nature-inspired optimization approaches to 2d geometric modelling for turbomachinery applications. *Mathematical Problems in Engineering*, vol. 2013:15 pages, 2013.
- [23] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, ACM '68, pages 517–524, New York, NY, USA, 1968. ACM.
- [24] Maxim A. Olshanskii, Arnold Reusken, and Jörg Grande. A Finite Element Method for Elliptic Equations on Surfaces. *SIAM Journal on Numerical Analysis*, 47(5):3339–3358, 2009.
- [25] Laurent D. Cohen and Isaac Cohen. Finite-element methods for active contour models and balloons for 2-D and 3-D images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1131–1147, 1993.
- [26] A. Mazroui, H. Mraoui, D. Sbibi, and A. Tijini. A new method for smoothing surfaces and computing hermite interpolants*. *BIT NUMERICAL MATHEMATICS*, 47:613–635, 2007.
- [27] Torsten Shtze and Hubert Schwetlick. Bivariate free knot splines. *BIT Numerical Mathematics*, 43:153–178, 2003.
- [28] Helmut Pottmann and Stefan Leopoldseder. A concept for parametric surface fitting which avoids the parametrization problem. *Computer Aided Geometric Design*, 20(6):343–362, September 2003.
- [29] A. Gálvez and A. Iglesias. Particle swarm optimization for non-uniform rational b-spline surface reconstruction from clouds of 3d data points. *Information Sciences*, 192:174–192, 2012.
- [30] A. Gálvez, A. Iglesias, and J. Puig-Pey. Iterative two-step genetic-algorithm-based method for efficient polynomial b-spline surface reconstruction. *Information Sciences*, 182(1):56–76, 2012.

- [31] Chun-Gang Zhu and Ren-Hong Wang. Least squares fitting of piecewise algebraic curves. *Mathematical Problems in Engineering*, vol. 2007:11 pages, 2007.
- [32] H. Haron, A. Rehman, D. I. S. Adi, S. P. Lim, and T. Saba. Parameterization method on b-spline curve. *Mathematical Problems in Engineering*, vol. 2012:22 pages, 2012.
- [33] I. J. Schoenberg and A. Whitney. On pólya frequency functions. iii. the positivity of translation determinants with an application to the interpolation problem by spline curves. *Trans. Amer. Math. Soc.*, 74:246–259, 1953.
- [34] Akemi Gálvez and Andrés Iglesias. Firefly algorithm for explicit b-spline curve fitting to data points. *Mathematical Problems in Engineering*, 2013:12 pages, 2013.
- [35] F.J. Cuevas, M. Servin, O.N. Stavroudis, and R. Rodriguez-Vera. Multi-layer neural network applied to phase and depth recovery from fringe patterns. *Optics Communications*, 181(46):239 – 259, 2000.
- [36] F.J Cuevas, M Servin, and R Rodriguez-Vera. Depth object recovery using radial basis functions. *Optics Communications*, 163(46):270 – 277, 1999.
- [37] L. E. Toledo and F. J. Cuevas. Optical metrology by fringe processing on independent windows using a genetic algorithm. *Experimental Mechanics*, 48(4):559–569, 2008.
- [38] Daniel Malacara, Manuel Servin, and Zacarias Malacara. *Interferogram Analysis for Optical Testing*. CRC Press, second edition edition, March 2005.
- [39] G. Ayala, A. L. Rivera, A. Dávila, G. Garnica, and V. M. Castaño. Analysis of mechanical vibrations through speckle interferometry: A phase-space approach. *Optik-International Journal for Light and Electron Optics*, 121(22):2028–2035, 2010.
- [40] A. Dávila, G. Garnica, J. A. López, and F. J. Carrión. Fatigue damage detection using a speckle-contrast technique. *Optics and Lasers in Engineering*, 47(3):398–402, 2009.

- [41] Alberto Aguilar, Abundio Dávila, and J. E. A. Landgrave. Displacement measurement with multi-level spiral phase filtering in speckle interferometry. *Optics and Lasers in Engineering*, 52:19–26, 2014.
- [42] M. Takeda, H. Ina, and S. Kobayashi. Fourier-transform method of fringe-pattern analysis for computer based topography and interferometry. *J. Opt. Soc. Am.*, 72(156), 1981.
- [43] J. C. Estrada, M. Servin, and J. L. Marroquin. Local adaptable quadrature filters to demodulate single fringe patterns with closed fringes. *In Opt. Express*, 15(5), 2007.
- [44] M. Servin, J. C. Estrada, and J. A. Quiroga. The general theory of phase shifting algorithms. *In Opt. Express*, 17(24), 2009.
- [45] F. J. Cuevas, J. H. Sossa-Azuela, and M. Servin. A parametric method applied to phase recovery from fringe pattern based on a genetic algorithm. *Optics Communications*, 203:213–223, March 2002.
- [46] F. J. Cuevas, F. Mendoza, M. Servin, and J. H. Sossa-Azuela. Window fringe pattern demodulation by multi-functional fitting using a genetic algorithm. *Optics Communications*, 261:231–239, 2006.
- [47] Charles Darwin. On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life., 1859. First Edition.
- [48] John R. Koza. Genetic programming: On the programming of computers by means of natural selection, 1992.
- [49] V. Ayala-Ramirez, C. H. Garcia-Capulin, A. Perez-Garcia, and R. E. Sanchez-Yanez. Circle detection on images using genetic algorithms. *Pattern Recognition Letters*, 27(6):652 – 657, 2006.
- [50] I. Cruz-Aceves, J. G. Avina-Cervantes, J. M. Lopez-Hernandez, H. Rostro-Gonzalez, C. H. Garcia-Capulin, M. Torres-Cisneros, and R. Guzman-Cabrera. Multiple active contours guided by differential evolution for medical image segmentation. *Computational and Mathematical Methods in Medicine*, vol. 2013:14 pages, 2013.

- [51] Edwin D. De Jong, Dirk Thierens, and Richard A. Watson. Hierarchical genetic algorithms. In Xin Yao, EdmundK. Burke, JosA. Lozano, Jim Smith, JuanJulin Merelo-Guervs, JohnA. Bullinaria, JonathanE. Rowe, Peter Tio, Ata Kabn, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 232–241. Springer Berlin Heidelberg, 2004.
- [52] Kim-Fung Man, Kit-Sang Tang, and Sam Kwong. *Genetic Algorithms: Concepts and Designs with Disk*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1999.
- [53] Gautam Garai and B. B. Chaudhuri. A distributed hierarchical genetic algorithm for efficient optimization and pattern matching. *Pattern Recognition*, 40(1):212 – 228, 2007.
- [54] Lai Chih-Chin and Chang Chuan-Yu. A hierarchical evolutionary algorithm for automatic medical image segmentation. *Expert Systems with Applications*, 36(1):248 – 259, 2009.
- [55] Carl De Boor. On calculating with b-spline. *Journal of Approximation Theory*, 6:50–62, 1972.
- [56] Carl De Boor. *A Practical Guide to Splines*. New York: Springer-Verlag, 1978.
- [57] Les Piegl and Wayne Tiller. *The NURBS Book (2nd Ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1997.
- [58] Stefan Jakobsson, Bjrn Andersson, and Fredrik Edelvik. Rational radial basis function interpolation with applications to antenna design. *Journal of Computational and Applied Mathematics*, 233(4):889 – 904, 2009.
- [59] H.-S. Park, W. Pedrycz, Y.-D. Chung, and S.-K. Oh. Modeling of the charging characteristic of linear-type superconducting power supply using granular-based radial basis function neural networks. *Expert Systems with Applications*, 39(1):1021 – 1039, 2012.
- [60] J.L. Valds, R. Biscay, and J.C. Jimenez. Geometric selection of centers for radial basis function approximations involved in intensive computer simulations. *Mathematics and Computers in Simulation*, 48(3):295 – 306, 1999.

- [61] Jonathan C. Carr, W. Richard Fright, and Richard K. Beatson. Surface interpolation with radial basis functions for medical imaging. *IEEE Transactions on Medical Imaging*, 16:96–107, 1997.
- [62] Bruce A. Whitehead and Timothy D. Choate. Cooperative - competitive genetic evolution of radial basis function centers and widths for time series prediction. *IEEE Transactions on Neural Networks*, 7:869–880, 1995.
- [63] C. H. Garcia-Capulin, F. J. Cuevas, G. Trejo-Caballero, and H. Rostro-Gonzalez. A hierarchical genetic algorithm approach for curve fitting with b-splines. *Genetic Programming and Evolvable Machines*, pages 1–16, 2014.
- [64] C. H. Garcia-Capulin, F. J. Cuevas, G. Trejo-Caballero, and H. Rostro-Gonzalez. Hierarchical genetic algorithm for b-spline surface approximation of smooth explicit data. *Mathematical Problems in Engineering*, vol. 2014:11 pages, 2014.
- [65] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
- [66] Kalyanmoy Deb and Ram B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148, 1995.
- [67] J. Andersson and D. Wallace. Pareto optimization using the struggle genetic crowding algorithm. *Engineering Optimization*, 34(6):623–643, December 2002.
- [68] C. C. Borges and H. J. Barbosa. Obtaining a restricted pareto front in evolutionary multiobjective optimization. *Foundations of Computing and Decision Sciences*, 26(1):5–21, 2001.
- [69] Marco Tomassini. *Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time (Natural Computing Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [70] D. G. Carmichael. Computation of pareto optima in structural design. *International Journal for Numerical Methods in Engineering*, 15:925–952, 1980.

- [71] Mourad Bessaou, Alain Pérowski, and Patrick Siarry. Island model cooperating with speciation for multimodal optimization. In *PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, pages 437–446, London, UK, 2000. Springer-Verlag.
- [72] Akemi Gálvez and Andrés Iglesias. Efficient particle swarm optimization approach for data fitting with free knot b-splines. *Computer-Aided Design*, 43(12):1683–1692, December 2011.
- [73] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. ISBN 3-900051-07-0.
- [74] Jeng-Neng Hwang, Shyh-Rong Lay, M. Maechler, R.D. Martin, and J. Schimert. Regression modeling in back-propagation and projection pursuit learning. *Neural Networks, IEEE Transactions on*, 5(3):342 – 353, may 1994.
- [75] T. X. Yue and S. H. Wang. Adjustment computation of hasm: a high-accuracy and high-speed method. *International Journal of Geographical Information Science*, 24(11):1725–1743, 2010.
- [76] W. S. Cleveland. Robust locally weighed regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:829–836, 1979.
- [77] Gannavarpu Rajshekhar and Pramod Rastogi. Phase estimation using a state-space approach based method. *Opt. Lasers Eng.*, 51(8):1004–1007, 2013.
- [78] Gannavarpu Rajshekhar and Pramod Rastogi. Multiple signal classification technique for phase estimation from a fringe pattern. *Appl. Opt.*, 51(24):5869–5875, 2012.
- [79] C. J. Tay, C. Quan, F. J. Yang, and X. Y. He. A new method for phase extraction from a single fringe pattern. *Optics Communications*, 239:251–258, May 2004.
- [80] Gannavarpu Rajshekhar and Pramod Rastogi. Fringe demodulation using the two-dimensional phase differencing operator. *Optics Letters*, 37(20):4278–4280, October 2012.