



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE TELECOMUNICACIONES

**DESARROLLO DE UN APLICATIVO DIDÁCTICO PARA LA OPTIMIZACIÓN
EN EL DISEÑO DE ANTENAS TIPO PATCH EMPLEANDO ALGORITMOS
METAHEURÍSTICOS INSPIRADOS EN LA NATURALEZA**

Trabajo de titulación previo a la obtención del
título de Ingeniero en Telecomunicaciones

AUTORES: LIZBETH MICAELA NARANJO MORA
ALEX CRISTOPHER OCHOA SIAVICHAY
TUTOR: ING. JUAN PABLO BERMEO MOYANO, MGT.

Cuenca - Ecuador

2023

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Lizbeth Micaela Naranjo Mora con documento de identificación N° 0104684295 y Alex Cristopher Ochoa Siavichay con documento de identificación N° 0106636731; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 8 de marzo del 2023

Atentamente,



Lizbeth Micaela Naranjo Mora
0104684295



Alex Cristopher Ochoa Siavichay
0106636731

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO
DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Lizbeth Micaela Naranjo Mora con documento de identificación No. 0104684295 y Alex Cristopher Ochoa Siavichay con documento de identificación No. 0106636731, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos los autores del Proyecto técnico: “Desarrollo de un aplicativo didáctico para la optimización en el diseño de antenas tipo Patch empleando algoritmos metaheurísticos inspirados en la naturaleza”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Telecomunicaciones, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 8 de marzo del 2023

Atentamente,



Lizbeth Micaela Naranjo Mora
0104684295



Alex Cristopher Ochoa Siavichay
0106636731

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Juan Pablo Bermeo Moyano con documento de identificación N° 0102645884, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DESARROLLO DE UN APLICATIVO DIDÁCTICO PARA LA OPTIMIZACIÓN EN EL DISEÑO DE ANTENAS TIPO PATCH EMPLEANDO ALGORITMOS METAHEURÍSTICOS INSPIRADOS EN LA NATURALEZA**, realizado por Lizbeth Micaela Naranjo Mora con documento de identificación N° 0104684295 y por Alex Christopher Ochoa Siavichay con documento de identificación N° 0106636731, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 8 de marzo del 2023

Atentamente,



Ing. Juan Pablo Bermeo Moyano, Mgt.

0102645884

AGRADECIMIENTOS

Agradecimientos de Micaela Naranjo

Quiero agradecer a Dios por permitirme llegar a culminar esta meta. A mis padres que siempre me han estado ahí para brindarme su apoyo incondicional, gracias a ellos e salido adelante en mi vida y en la carrera. Gracias por su inquebrantable apoyo emocional y financiero. Les debo todo lo que he logrado hasta ahora, y les agradezco desde el fondo de mi corazón por ser los pilares de mi vida

A mis hermanos y familia, que siempre han sido un apoyo para mí. Siempre me han motivado a seguir adelante para cumplir mis metas. Quienes siempre me han extendido su mano en los buenos y sobre todo en los malos momentos. A esa persona quien siempre con su amor y sus palabras de aliento me motivo a no rendirme y seguir luchando por cada uno de los desafíos que se han presentado.

A mis amigos, ya que ellos son como mi familia. Siempre me han apoyado para no rendirme y me han dado un hombro donde apoyarme y seguir a pesar de las adversidades.

A los docentes, quienes compartieron sus conocimientos y que fueron parte de formación académica. A mi tutor, Ing. Juan Pablo Bermeo quien fue una de nuestras guías en este proyecto y nos brindo sus conocimientos y recomendaciones.

Al Ing. Fernando Guerrero quien siempre nos a motivado y nos a guiado para culminar este proyecto. A mi compañero de tesis, Alex por su paciencia y por su confianza, para poder culminar con éxito este proyecto.

Agradecimientos de Alex Ochoa S.

Agradezco a Dios por guiarme en este trayecto que culmina después de tan arduo esfuerzo, brindarme una gran familia, personas que me apoyaron siempre, salud, dedicación y permitirme seguir adelante.

Al Ing. Fernando Guerrero quien me a motivado, apoyado en este trayecto siendo un gran mentor, un apoyo en los distintos proyectos que he trabajado contribuyendo tanto en mi aprendizaje como desarrollo profesional. Al Ing. Juan Pablo Bermeo quien fue el tutor de este proyecto, además de contribuir en mi formación académica. A mi compañera de tesis quien fue un gran apoyo en este transcurso. A mi grupo de amigos con los que compartí muchas risas e historias. También agradezco a mis padres por haberme traído al mundo, apoyarme en mis metas por darme el apoyo necesario para cumplirlas.

A mis abuelitos que para mi son mis segundos padres quienes me criaron, me brindaron su paciencia, amor. A toda mi familia ya que fueron un pilar fundamental en mi vida, siempre me aconsejaron, jamas dejaron de apoyarme moralmente.

A la persona que me acompaño en todo este tiempo teniendo la paciencia, por brindarme el apoyo necesario para seguir en los momentos mas difíciles, escuchar mis problemas, el cariño de estar allí para mi sin buscar nada a cambio haciéndome feliz, motivándome a seguir, de corazón agradezco tantos momentos de felicidad, cariño momentos que han se han grabado en mi vida siendo una persona muy especial en mi vida, a la cual aprecio desde el fondo de mi ser.

DEDICATORIA

Dedicatoria de Micaela Naranjo

Quiero dedicar este trabajo a mis padres Miguel y Verónica, quienes con su amor, apoyo, paciencia y sabiduría, han sido mi sostén y mi guía en cada paso que he dado, sin ustedes este trabajo no sería posible. A mis hermanos y a la niña de mis ojos que siempre me han motivado y han sido mi fuente de inspiración para mi, a pesar de ser mas pequeños siempre me han enseñado algo nuevo.

A mi familia, quienes han sido mi pilar fundamental en los momentos mas difíciles y han celebrado cada logro conmigo, como si fuera el suyo mismo. Por su amor y apoyo que siempre me han brindado, por enseñarme el valor del esfuerzo y la dedicación, y por inspirarme a ser mejor cada día. Este logro también es suyo.

Con amor y gratitud.

Micaela Naranjo.

Dedicatoria de Alex Ochoa

Quiero dedicar este trabajo a Dios, a los docentes de la carrera quienes me ayudaron en mi formación, a las personas que me apoyaron, me brindaron su confianza, a las personas que fueron parte de mi vida, me ayudaron a formarme como persona, a la persona que es parte de mi vida, el motivo de mi felicidad. Todo este esfuerzo es de mi para ustedes esperando que se sientan felices y orgullosos de verme lograr una gran meta de muchas en mi vida.

Con enorme gratitud y afecto.

Alex Ochoa.

Índice General

Índice General	IX
Índice de Figuras	XI
Índice de Tablas	XIII
Resumen	XIV
Abstract	XVI
Objetivos	XVIII
Antecedentes o Problema de Estudio	XX
Justificación	XXII
Introducción	1
1. Estado del arte	3
1.1. Algoritmos metaheurísticos basados en la naturaleza	3
1.1.1. Metaheurística	5
1.1.2. Tipos de algoritmos metaheurísticos	6
1.2. Aplicación de los algoritmos de optimización para las antenas	11
1.3. Antenas tipo patch	12
1.3.1. Patch rectangular	14
2. Criterios de diseño e implementación	18
2.1. Algoritmos basados en inteligencia de enjambre.	18

2.1.1. Algoritmos implementados	19
2.2. Desarrollo de la Aplicación	35
2.2.1. App Designer en Matlab	36
3. Pruebas y análisis de resultados	41
3.1. Parámetros	42
3.1.1. Función Fitness	42
3.1.2. Criterio de paro	43
3.1.3. S11	43
3.1.4. Resonancia	44
3.2. Aplicativo	44
3.3. Pruebas	45
3.3.1. Bat Algorithm	47
3.3.2. Firefly Algorithm	50
3.3.3. Harris Hawks Optimization	52
3.3.4. Marine Predators Algorithm	55
3.3.5. Particle Swarm Optimization	57
3.3.6. Slime Mould Algorithm	59
3.4. Análisis y comparación de los resultados obtenidos.	61
4. Conclusiones y Trabajos Futuros	70
4.1. Anexo 1	73
Glosario	87
Referencias	88

Índice de Figuras

1.1. Componentes de los problemas de optimización.	5
1.2. Clasificación algoritmos metaheurísticos.	6
1.3. Clasificación algoritmos metaheurísticos.	11
1.4. Aplicaciones de algoritmos en el diseño de antenas.	12
1.5. Formas representativas de los elementos de las antenas patch (Imagen adaptada de [1]).	13
1.6. Antena patch rectangular.	15
2.1. Comportamiento del murciélago.	21
2.2. Comportamiento de las luciérnagas.	23
2.3. Comportamiento del Halcón de Harris.	25
2.4. Movimiento Browniano y movimiento de Levy.	27
2.5. Las tres fases de optimización del MPA.	30
2.6. Ilustración del algoritmo PSO.	31
2.7. morfología de forrajeo del slime mould.	33
2.8. Función fitness de todos los algoritmos empleados.	38
2.9. Simulación en HFSS en la First Phase.	38
2.10. Resultados de la primera pestaña del aplicativo.	39
2.11. Función fitness de todos los algoritmos empleados.	39
3.1. Diagrama de flujo.	46
3.2. Antena patch rectangular.	47
3.3. Algoritmo BAT, parámetros de reflexión de la antena.	48
3.4. Mejor parámetro de reflexión de la antena usando el algoritmo BAT.	48
3.5. Algoritmo BAT, Función fitness obtenida en 30 iteraciones.	49

3.6. Parámetro de reflexión de la antena usando el algoritmo Firefly.	50
3.7. Mejor parámetro de reflexión de la antena usando el algoritmo Firefly.	51
3.8. Mejor parámetro de reflexión de la antena usando el algoritmo FA.	51
3.9. Parámetro de reflexión de la antena usando el algoritmo HHO.	52
3.10. Mejor parámetro de reflexión de la antena usando el algoritmo HHO.	53
3.11. Función fitness del algoritmo HHO.	54
3.12. Parámetro de reflexión de la antena usando el algoritmo MPA.	55
3.13. Mejor parámetro de reflexión de la antena usando el algoritmo MPA.	56
3.14. Función fitness del algoritmo MPA.	56
3.15. Parámetro de reflexión de la antena usando el algoritmo PSO.	57
3.16. Mejor parámetro de reflexión de la antena usando el algoritmo PSO.	58
3.17. Función fitness del algoritmo PSO.	59
3.18. Parámetro de reflexión de la antena usando el algoritmo SMA.	60
3.19. Mejor parámetro de reflexión de la antena usando el algoritmo SMA.	60
3.20. Función fitness SMA.	61
3.21. Función fitness de todos los algoritmos empleados.	63
3.22. Resultados de convergencia obtenidos para la antena con los seis algoritmos.	65
3.23. Los mejores parámetros de reflexión de la antena usando algoritmos de optimización.	66

Índice de Tablas

1.1. Sustratos para antenas patch	15
3.1. Resultados obtenidos del algoritmo BAT.	49
3.2. Resultados obtenidos del algoritmo FA.	52
3.3. Resultados obtenidos del algoritmo HHO.	54
3.4. Resultados obtenidos del algoritmo MPA.	57
3.5. Resultados obtenidos del algoritmo PSO.	58
3.6. Resultados obtenidos del algoritmo SMA.	61
3.7. Resultados obtenidos de los algoritmos.	62
3.8. Resultados obtenidos.	68

Resumen

En este trabajo se desarrolló una recopilación sobre los algoritmos de optimización metaheurísticos inspirados en la naturaleza; específicamente, aquellos que emplean inteligencia de enjambre, por su amplia aplicación y excelentes resultados en el diseño de antenas patch. Los algoritmos incluidos en este trabajo son: Bat Algorithm (BA), Firefly Algorithm (FA), Harris Hawks Optimization (HHO), Marine Predators Algorithm (MPA), Particle Swarm Optimization (PSO) y Slime Mould Algorithm (SMA). Todos los algoritmos se emplearon en el mismo problema que consistió en optimizar las dimensiones de una antena patch rectangular (W , L y Y_0), con el objetivo de mejorar las características de acoplamiento en una frecuencia de resonancia específica.

Una vez programados y probados los algoritmos de optimización, se desarrolló un aplicativo empleando la interfaz del App Designer de Matlab. Este aplicativo pretende convertirse en una herramienta didáctica para que los estudiantes de la asignatura de teoría de antenas o asignaturas relacionadas, puedan optimizar sus diseños sin necesidad de recurrir a métodos empíricos de prueba y error. El aplicativo tiene dos fases para la optimización, la primera consiste en ingresar los parámetros iniciales de la antena (constante dieléctrica, impedancia, altura del sustrato, frecuencia de resonancia, ancho de la ventana de frecuencia, espacio de búsqueda), mientras que la segunda permite seleccionar las condiciones de optimización (seleccionar el algoritmo, cantidad de iteraciones, cantidad de agentes de búsqueda).

Finalmente, se realiza un análisis comparativo en base a los resultados de todos los algoritmos de optimización empleados y se proponen dos guías de prácticas para que los estudiantes interesados puedan emplear este recurso dentro de sus procesos de aprendizaje práctico.

Palabras clave: Algoritmo de optimización; Antena patch; función fitness; iteración; agentes de búsqueda; espacio de búsqueda; Bat Algorithm (BA); Firefly Algorithm (FA); Harris Hawks Optimization (HHO); Marine Predators Algorithm (MPA); Particle Swarm Optimization (PSO); Slime Mould Algorithm (SMA);

Abstract

In this work, a compilation of nature-inspired metaheuristic optimization algorithms was developed; specifically, those that employ swarm intelligence, due to their wide application and excellent results in patch antennas design. Algorithms included in this work are: Bat Algorithm (BA), Firefly Algorithm (FA), Harris Hawks Optimization (HHO), Marine Predators Algorithm (MPA), Particle Swarm Optimization (PSO), and Slime Mould Algorithm (SMA). All algorithms were used on the same problem which consisted of dimensions optimizing of a rectangular patch antenna (W , L and Y_0), with the objective of improving coupling characteristics at a specific resonant frequency. Once optimization algorithms were programmed and tested, an application was developed using the Matlab App Designer interface. This application is intended to become a didactic tool for students of antenna theory or related subjects, so that they can optimize their designs without resorting to empirical trial and error methods. This application has two phases for optimization, the first one consists of entering antenna initial parameters (dielectric constant, impedance, substrate height, resonance frequency, frequency window width, search space), while the second one allows selecting optimization conditions (selecting the algorithm, number of iterations, number of search agents).

Finally, a comparative analysis is made based on results of all used optimization algorithms and two practice guides are proposed so that interested students can use this resource as part of their practical learning process.

Keywords: Optimization algorithm; Patch antenna; fitness function; iteration; search agents; search space; Bat Algorithm (BA); Firefly Algorithm (FA); Harris

Hawks Optimization (HHO); Marine Predators Algorithm (MPA); Particle Swarm Optimization (PSO); Slime Mould Algorithm (SMA);

Objetivos

Objetivo General

- Desarrollar un aplicativo didáctico para la optimización en el diseño de antenas tipo patch empleando algoritmos metaheurísticos en la naturaleza para el laboratorio de Telecomunicaciones de la Universidad Politécnica Salesiana sede Cuenca.

Objetivos específicos:

- Realizar el estado del arte de algoritmos metaheurísticos inspirados en la naturaleza para determinar al menos tres algoritmos adecuados para la implementación en el aplicativo.
- Desarrollar el aplicativo con los algoritmos identificados en el objetivo 1; además de tres algoritmos metaheurísticos adicionales tomados de trabajos previos.
- Contrastar los resultados de simulación de los seis algoritmos para determinar cuál es el óptimo.

Delimitaciones

La finalidad del proyecto es el diseño y la implementación de un aplicativo de optimización de algoritmos para antenas. Con este aplicativo los estudiantes podrán optimizar las antenas tipo patch y dar paso a sus respectivas modificaciones.

El aplicativo consta de seis algoritmos inspirados en la naturaleza, los cuales serán usados en las antenas patch para optimizar sus medidas, dando mejores

resultados en el ancho de banda y mayor resonancia en la frecuencia central. El aplicativo usa como intermediario un programa realizado en Visual Basic, que toma los datos de ANSYS HFSS que es el software que simula estructuras de alta frecuencia y el comportamiento electromagnético de cualquier tipo de antena mediante el Método de Elementos Finitos (de sus siglas en inglés Finite Element Method, FEM). Posteriormente, el software ANSYS HFSS envía los resultados obtenidos de la simulación de las antenas con Matlab. Con los datos recibidos, en Matlab se encuentran programados los algoritmos que comenzarán el proceso de optimización, trabajando de manera recursiva con ANSYS HFSS. En función de las interacciones definidas por el usuario, se compara para determinar los mejores resultados, permitiendo que los estudiantes desarrollen sus propias modificaciones con las medidas propuestas para una antena patch. Con esto, lo que se busca es optimizar el tiempo del estudiante con respecto al diseño de la antena para la posterior construcción, de ser el caso. Las pruebas de funcionamiento serán simuladas y comparadas entre algoritmos para determinar cuál es el diseño óptimo.

Antecedentes

Los avances en materia de ciencia y tecnología generan aportes constantes en todos los ámbitos imaginados y la optimización de procesos es uno de ellos. Puntualmente, cuando se habla de la optimización de antenas, consiste en un problema de gran complejidad ya que no existen todavía fórmulas matemáticas exactas para su resolución. Entonces, la optimización se convierte en un proceso empírico que podría consumir considerable tiempo y recursos si no se emplea una metodología adecuada. Es por eso que se requiere seguir desarrollando o adaptando metodologías que soporten el diseño de las antenas. Este es un tema que ha recibido especial atención por parte de los investigadores debido a la importancia que representan las antenas en los sistemas de comunicaciones inalámbricas. Con el avance de la tecnología se ha buscado algún método que ayude en la complejidad de optimización de antenas mediante procesos empíricos, pues no existen fórmulas puntuales y exactas para diseñar antenas, todos son modelos aproximados.

Los algoritmos, de manera general, ayudan a resolver problemas específicos de forma más rápida, ordenada y siguiendo cierta secuencia con n interacciones según corresponda para dar una solución concreta al problema, evitando así realizar un proceso largo y tardado como es el de prueba y error. En el caso específico de las antenas, es posible implementar algoritmos metaheurísticos inspirados en la naturaleza para optimizar las dimensiones y mejorar los resultados. Estos algoritmos son procesos iterativos más complejos con ciertas restricciones que los hacen difíciles de resolver, debido a esto se aplican métodos computacionales.

Dentro del campo del diseño de antenas se tiene trabajos orientados a la optimización de este proceso; en este sentido, se tiene el trabajo de titulación denominado *“Implementación de algoritmos de optimización en el diseño de antenas*

planares” realizada por la estudiante Paola Guamán. En este trabajo se implementaron tres algoritmos para antenas circulares y rectangulares optimizados. Como se menciona en este trabajo de titulación, existe una variedad de aplicaciones para los algoritmos en el campo de la optimización, una de ellos son las antenas, donde se centra en la optimización de las dimensiones para obtener una mayor resonancia en una frecuencia central de 2.4 GHz. De acuerdo con lo mencionado, es necesario el uso de algoritmos de optimización para mejorar el diseño de las antenas, esto con el objetivo de obtener de forma eficiente una mejor aproximación en los distintos parámetros de las antenas como su resonancia, frecuencia central, ancho de banda y dimensiones. Varios de los métodos de optimización realizan n interacciones, en cada una de ellas evalúan una función objetivo la cual está en función del tamaño de la antena que se pretende optimizar (maximizar o minimizar). En este proceso se pretende encontrar una solución óptima según los parámetros de convergencia y restricciones establecidas.

Existen varios software de simulación de antenas como FEKO, MAXWELL, CST STUDIO SUITE y otros, la mayoría tienen costo por su uso; además, se necesita una capacitación para poder manejarlos, lo que implica una inversión de tiempo y dinero. También es preciso un computador con un *hardware* robusto capaz de soportar y procesar dichos *softwares*, siendo la mayoría “cajas negras” en donde no existe la facilidad para interconectarse y compartir datos directamente con los cuales implementar algoritmos de optimización. Sin embargo, el software ANSYS HFSS, aunque también es un recurso pagado que requiere licencia para su uso, tiene mayor apertura para interactuar con otros programas, como es el caso de Visual Basic. De esta manera, ANSYS HFSS se perfila como una opción adecuada para generar desarrollos que involucren la intercomunicación entre distintos programas.

Justificación

Actualmente, la tecnología siempre está en constante desarrollo para resolver distintos problemas que se presentan en la sociedad. Gracias a esto, las tecnologías de comunicación inalámbricas han hecho fácil la comunicación entre distintos dispositivos, los cuales necesitan antenas capaces de transmitir cada vez a mayores anchos de banda. Sin embargo, los tipos de modelos son infinitos y complejos para la simulación, necesitando grandes tiempos de procesamiento, por lo que se utilizan varios métodos numéricos, entre ellos el método de los momentos, elementos finitos y otros. En la mayoría de los métodos, lo que se tiene son aproximaciones sobre los cálculos y simulación de campos de las antenas. Esto implica que el proceso de diseño de antenas es altamente complejo, es decir, que no es fácil dar con un resultado deseado, siendo más un proceso de prueba y error. Por lo tanto, es necesario un complemento práctico que permita combinar la simulación y cálculo de antenas con algoritmos de optimización, con el fin de mejorar los procesos de construcción y diseño de antenas; además de mejorar la comprensión de la temática de la teoría de antenas.

En base a lo mencionado, es necesaria una herramienta que facilite la comprensión del diseño de antenas, pues con el paso de los años, se han presentado nuevas problemáticas y requerimientos de los usuarios. Esto ha impulsado el desarrollo de nuevas propuestas de aplicativos (programas de cómputo) que facilitan el diseño y la simulación de antenas. El desarrollo de un aplicativo permitirá a los estudiantes ser más productivos en el diseño de antenas, ya que la complejidad matemática será solventada mediante el aplicativo, sumado al hecho de que los estudiantes comprenden mejor la temática del diseño de antenas con resultados prácticos que pueden obtenerse mediante una herramienta didáctica de programación. Vale indicar que, la Universidad Politécnica Salesiana sede Cuenca tiene una previa

trayectoria en el estudio de la teoría de antenas, en la que se han realizados algunos artículos sobre antenas y sus arreglos como: [2, 3, 4, 5], entre otros artículos. Esto nos allana el camino para incursionar en la optimización de antenas.

Partiendo de los artículos publicados, más los avances de la tecnología en las comunicaciones inalámbricas en el campo de las antenas, se busca implementar seis algoritmos de optimización inspirados en la naturaleza para el desarrollo del aplicativo. Estos algoritmos nos permitirán encontrar las dimensiones óptimas para las antenas patch que potencialicen y optimicen ciertos parámetros de funcionamiento. Estos seis algoritmos serán incorporados a un programa (aplicativo) que facilite la optimización del diseño de antenas de tipo patch. Los estudiantes podrán ingresar los parámetros para que el programa genere las dimensiones óptimas de las antenas, de esta forma ahorrarán tiempo y recursos; debido a que no tendrán que hacer cálculos manuales y mecánicos para cada uno de los diseños, luego tenerlos que simularlos para obtener los resultados. Es decir, ya no realizarán ellos el proceso de “prueba y error”, sino que el aplicativo mediante los algoritmos llegará al resultado deseado, variando los parámetros de tamaño de la antena.

Por todo lo mencionado, es necesario un aplicativo que permita de forma didáctica, facilitar el diseño y optimización de las antenas con vistas a alcanzar un mejor desarrollo en la parte académica de los estudiantes de la carrera de Telecomunicaciones de la Universidad Politécnica Salesiana. Al no existir una herramienta didáctica asequible para la optimización de antenas durante el proceso de aprendizaje, lo que limita la capacidad de los estudiantes en el diseño de antenas, se plantea implementar seis algoritmos de optimización dentro del aplicativo, donde el usuario podrá ingresar los parámetros de la antena que desea optimizar. El aplicativo calculará las respuestas óptimas con los diferentes algoritmos, se compararán y se seleccionarán aquellos que ofrezcan la solución óptima. De esta manera el proceso de optimización será transparente para el usuario, lo que implica un menor trabajo que, de otra forma, tendría que hacerlo en un formato de prueba y error.

Introducción

En la actualidad, las nuevas generaciones de comunicaciones (5G y 6G) requieren sistemas cada vez más eficientes que permitan un mejor desempeño en diversas áreas de la tecnología. Uno de estos campos es el diseño de antenas, que busca crear dispositivos cada vez más óptimos para cumplir con las exigencias del mercado.

Para lograr este objetivo, se han desarrollado diversas técnicas de optimización que permiten mejorar el rendimiento de las antenas. Entre estas técnicas, destacan los algoritmos metaheurísticos inspirados en la naturaleza, que se han mostrado altamente eficientes en la resolución de problemas complejos de optimización.

En este contexto, el presente trabajo tiene como objetivo el desarrollo de un aplicativo didáctico para la optimización en el diseño de antenas tipo patch empleando algoritmos metaheurísticos inspirados en la naturaleza. Este aplicativo busca crear una herramienta útil y accesible para estudiantes y profesionales interesados en el diseño de antenas, y se enfoca en el uso de seis algoritmos de optimización basados en procesos naturales: Bat Algorithm (BA), Firefly Algorithm (FA), Harris Hawks Optimization (HHO), Marine Predators Algorithm (MPA), Particle Swarm Optimization (PSO) y Slime Mould Algorithm (SMA).

La motivación principal para el desarrollo de este aplicativo radica en la necesidad de contar con herramientas que permitan una optimización más eficiente en el diseño de antenas tipo patch, una de las estructuras más utilizadas en la actualidad en aplicaciones como la comunicación inalámbrica, la teledetección y la navegación.

Además, la aplicación de los algoritmos metaheurísticos inspirados en la naturaleza en este ámbito es una línea de investigación emergente, que ha demostrado una gran efectividad en la resolución de problemas complejos de optimización.

El objetivo principal de este trabajo de tesis es, por lo tanto, proporcionar una

herramienta intuitiva y fácil de usar para el diseño de antenas patch rectangulares, permitiendo a los usuarios experimentar con diferentes configuraciones y evaluar el impacto en las prestaciones de la antena.

Para lograr este objetivo, se ha utilizado el entorno de programación Matlab, que permite una gran flexibilidad en la implementación de algoritmos de optimización y la creación de interfaces gráficas de usuario amigables.

En resumen, este trabajo de tesis busca mejorar la comprensión y el conocimiento en el diseño de antenas a través de la creación de un aplicativo didáctico y accesible en Matlab que utilice algoritmos metaheurísticos inspirados en la naturaleza. Con ello, se espera contribuir al desarrollo de antenas más eficientes y optimizadas que permitan un mejor desempeño en diversas aplicaciones tecnológicas.

Capítulo 1

Estado del arte

El presente capítulo presenta una introducción a dos áreas de investigación en el campo de la inteligencia artificial: 1) algoritmos metaheurísticos basados en la naturaleza y 2) aplicación de los algoritmos de optimización para antenas patch.

Por un lado, los algoritmos metaheurísticos basados en la naturaleza se inspiran en los procesos naturales para resolver problemas complejos. Estos algoritmos han demostrado ser efectivos en la optimización de problemas que son difíciles de resolver mediante técnicas convencionales. Por otro lado, las Antenas patch son un tipo de antena plana que se utiliza en aplicaciones de radiofrecuencia. Son populares debido a sus características de bajo perfil, bajo costo y alta eficiencia en términos de radiación.

1.1. Algoritmos metaheurísticos basados en la naturaleza

La optimización es el proceso de desarrollo de una actividad que se lleva a cabo de la forma más eficiente posible, es decir, con los mínimos recursos y en el menor tiempo posible. Se puede entender un algoritmo como una secuencia de pasos finitos bien definidos que resuelven un problema. Los parámetros que suelen considerarse en la mayoría de los algoritmos se mencionan a continuación:

- Función objetivo.

En los algoritmos, la función objetivo es una medida matemática que permite evaluar la calidad de una solución para un problema dado. La función objetivo

es utilizada para guiar el proceso de búsqueda del algoritmo y ayudar a encontrar una solución óptima o aproximada.

Por ejemplo, en un problema de optimización, la función objetivo puede ser la maximización o minimización de una variable específica. En un problema de clasificación, la función objetivo puede ser la precisión o la tasa de error en la clasificación de datos.

Es importante elegir una función objetivo adecuada para el problema en cuestión, ya que esto puede tener un impacto significativo en los resultados finales del algoritmo.

- Variables.

En los algoritmos, las variables son valores que pueden tomar diferentes valores y se utilizan para representar características o elementos de un problema.

Por ejemplo, en un problema de optimización, las variables pueden representar las incógnitas que se buscan optimizar. En un problema de clasificación, las variables pueden representar las características de un objeto que se utilizan para clasificarlo.

Las variables son un componente importante de los algoritmos ya que su valor y su interacción con otros elementos del problema pueden influir en la solución final. Por lo tanto, es importante elegir cuidadosamente las variables que se van a utilizar y cómo se van a representar en el algoritmo.

- Restricciones.

En los algoritmos, las restricciones son condiciones o limitaciones que deben ser cumplidas por la solución óptima o aproximada para un problema dado. Estas restricciones pueden ser matemáticas o basadas en reglas o requisitos del problema.

Por ejemplo, en un problema de optimización, las restricciones pueden ser la limitación de los recursos disponibles o las restricciones de capacidad. En un problema de asignación, las restricciones pueden ser la limitación de los recursos o la limitación de la asignación de un trabajo a una sola persona.

Las restricciones son un componente importante de muchos problemas y pueden influir significativamente en la solución final. Por lo tanto, es importante tener en cuenta las restricciones al diseñar y ejecutar un algoritmo.

De manera resumida, se muestra en la Figura 1.1 los componentes que se deben tomar en cuenta al momento de optimizar.



Figura 1.1: Componentes de los problemas de optimización.

1.1.1. Metaheurística

Antes de revisar la clasificación de los algoritmos metaheurísticos se hará una revisión comprensiva del significado de metaheurística.

- **Significado de metaheurística:** Cuando se habla de una aplicación antes de alcanzar la mejor decisión posible se la conoce como metaheurística. Esto quiere decir que se realizan aplicaciones de experiencias previas para generar una especie de respuesta a un problema que aún no tiene solución [6].
- **Algoritmos Metaheurísticos:** Es una familia de algoritmos de aproximación de propósito general. Estos son, a menudo, procedimientos iterativos que impulsan una sub-búsqueda heurística y combinan inteligentemente diferentes conceptos para explorar y usar el espacio de búsqueda de manera adecuada [6].

El término metaheurística se introdujo en 1986 por Glover. Este término metaheurístico es definido como un método iterativo que guía un problema heurístico, combinado de

forma óptima para buscar distintos conceptos y optimiza de mejor manera el objetivo que se quiere lograr [7, 8].

1.1.2. Tipos de algoritmos metaheurísticos

Los algoritmos metaheurísticos de optimización son técnicas utilizadas para solucionar problemas de optimización complejos. Se basan en la exploración y la búsqueda de soluciones para encontrar la mejor solución posible para un problema dado. Estos algoritmos utilizan un enfoque indirecto y no garantizan encontrar la solución óptima, pero son efectivos en encontrar soluciones de alta calidad en un tiempo razonable.

En esta sección se presenta de manera general los tipos de algoritmos metaheurísticos como se ve en la Figura 1.2.

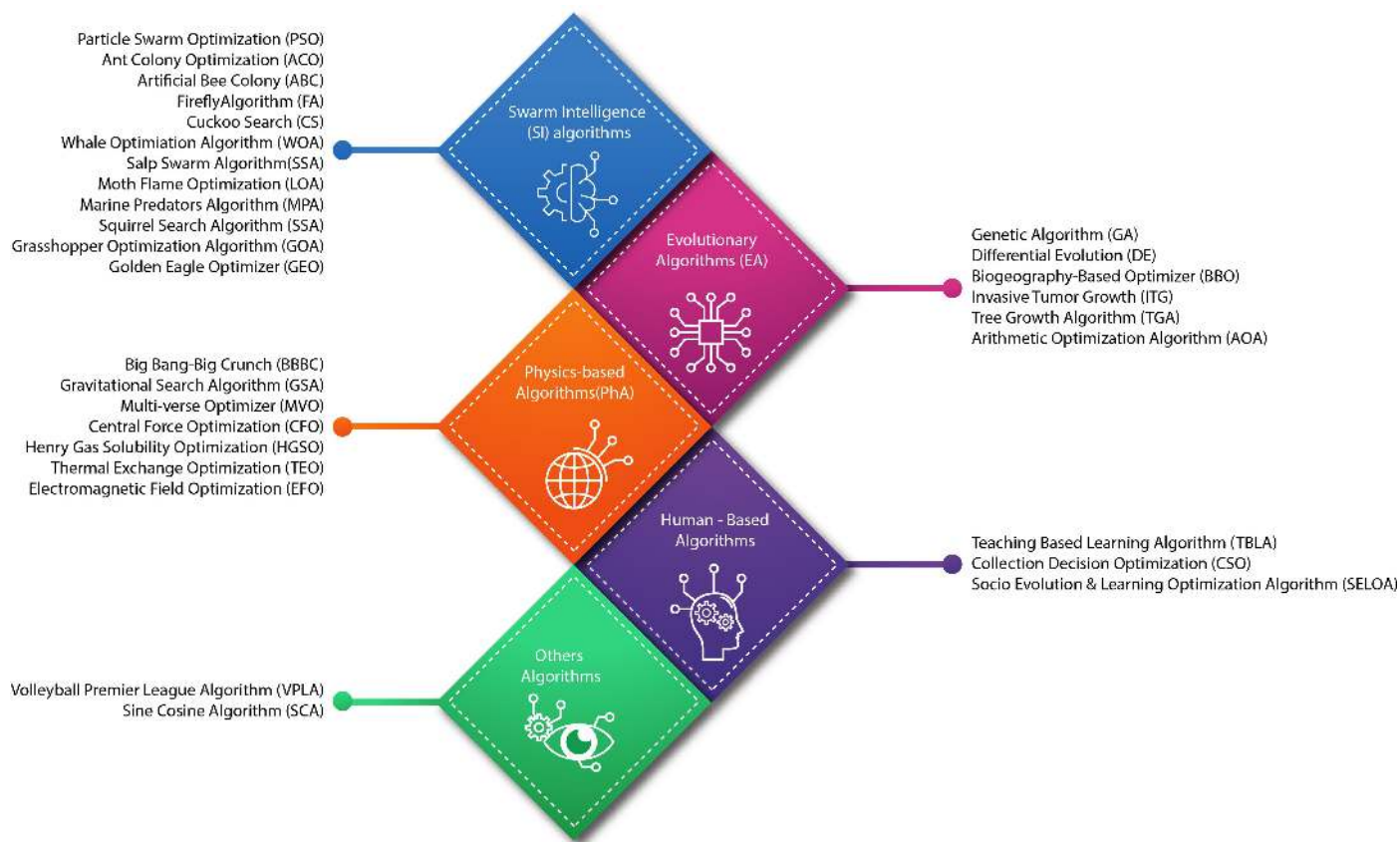


Figura 1.2: Clasificación algoritmos metaheurísticos.

Inteligencia de enjambre.

El término Inteligencia de enjambre (SI, del inglés Swarm Intelligence) hace referencia a los movimientos aleatorios de los sujetos y objetos como las hormigas, pájaros, peces, abejas, etc. [9]. SI es introducido en 1989 por Gerald y Jing Wang [10], en el ámbito de los sistemas robóticas celulares. En el caso SI se deben cumplir cinco reglas que las estableció Milonas: el principio de la proximidad, el principio de calidad, el principio de respuesta diversa, el principio de estabilidad y el principio de adaptabilidad [11, 12]. Además, de cumplir dos fases: fases de variación y la de selección, las cuales deben conservar el equilibrio de las posibles soluciones para actualizar las posiciones.

La inteligencia de enjambre corresponde a un conjunto de técnicas basadas en el estudio del comportamiento colectivo en sistemas auto organizativos y jerárquicos. Estos sistemas suelen consistir en un conjunto de agentes computacionales simples capaces de comprender y modificar parámetros locales. Esta técnica proporciona comunicación entre las personas y aquellos que reconocen cambios en el comportamiento causados por un comportamiento incorrecto. Aunque generalmente no existe una estructura de control central que gobierne el comportamiento de los agentes, las interacciones locales entre ellos a menudo conducen al surgimiento de un comportamiento global. Otra observación adicional es la falta de patrones ambientales claros.

En esta rama, las principales técnicas son la optimización de enjambre de partículas (PSO, del inglés Particle Swarm Optimization) [13] y la optimización por colonia de hormigas (ACO, del inglés Ant Colony Optimization). Existe otra variedad de algoritmos que pueden ser asociados dentro de esta rama, como lo son la optimización por enjambre de bacterias (BSFO, del inglés Bacteria Swarm Foraging Optimization), la búsqueda por difusión estocástica (SDS, del inglés Stochastic Diffusion Search) y el algoritmo de colmena de abejas artificiales (ABHA, del inglés Artificial Bee Hive Algorithm) [13].

Algoritmos Evolutivos

Los Algoritmos Evolutivos (EA, del inglés Evolutionary Algorithms) son una familia de la evolución natural que utilizan métodos sintéticos, incluidos el control de la población, la replicación, la diversidad y la selección final. Todo esto se basa en la teoría básica de la evolución darwiniana [14]. En general, los algoritmos son fáciles a primera vista, pero se vuelve difícil cargar la información central en la red [15].

Para que se realicen estos algoritmos, se debe plantear un objetivo cuantitativo, es decir, que sea medible. Para de esta manera ver el éxito de la métrica del objetivo que se planteo [15]. Esto cumple un proceso de una poblaciones, cada uno de estos va representar una punto de búsqueda en el espacio de las soluciones factibles y van a estar sometidas a un procesos de retroalimentación grupal para que vaya de generación en generación [16]. Finalmente, con el paso de las generaciones va teniendo mejores resultados en la problemática, llegando a y va a mostrar el mejor resultado que haya encontrado. Estos EA son ideales para la resolución de problemas de optimización, programación, planificación, diseño y gestión [17].

Las principales técnicas de optimización en estos algoritmos son los algoritmos genéticos (GA, del inglés Genetic Algorithm), las estrategias de evolución (ES, del inglés Evolution Strategy), la programación evolutiva (EP, del inglés Evolutionary Programming) y la programación genética (GP, del inglés Genetic Programming) [18]. A pesar de que estos algoritmos tienen diferentes aspectos entre si, todos se basan de un mismo proceso base. Cada uno de estos sostiene la idea de una población de búsqueda [19].

Algoritmos basados en la física

Algoritmos basados en la física (de sus siglas en inglés Physics-based algorithms,PhA) se refieren a técnicas de simulación que utilizan modelos y leyes físicas para resolver problemas. Estos algoritmos se utilizan en una amplia variedad de aplicaciones, incluyendo la simulación de fenómenos físicos, la resolución de ecuaciones diferenciales y la optimización de sistemas complejos. Algunos ejemplos de algoritmos basados en la física incluyen: dinámica molecular, simulación de fluidos

y resolución de ecuaciones termodinámicas.

Para estos algoritmos se pueden clasificar de la siguiente forma [20]:

- Los que son inspirados en las leyes de movimiento de Newton.
- Los que son inspirados en la fuerza gravitacional de Newton.
- Los que son inspirados en la mecánica celeste y la astronomía.
- Los que son inspirados del electromagnetismo.
- Los que son inspirados en la óptica.
- Los que son inspirados en la acústica.
- Los que son inspirados en la termodinámica.
- Los que son inspirados en la hidrología y la hidrodinámica.

Algoritmos basados en humanos

Los algoritmos basados en humanos son aquellos que se basan en el comportamiento humano para llevar a cabo tareas o solucionar problemas. Estos pueden ser algoritmos de aprendizaje automático o sistemas expertos que utilizan patrones de comportamiento humano para mejorar su desempeño [21].

Ejemplos incluyen chatbots que se comunican con humanos en lenguaje natural, sistemas de recomendación que imitan la forma en que los humanos recomiendan productos o servicios, y algoritmos de análisis de sentimientos que clasifican los sentimientos expresados en texto en base a patrones de comportamiento humano. Uno de los algoritmos de esta rama que más resalta es el de la Optimización Basada en la Enseñanza-Aprendizaje (TLBO, del inglés Physics-based algorithms), que se basa en la simulación de la comunicación e interacción entre un alumno y profesor en una aula de clases [21].

Búsqueda en inteligencia artificial en algoritmos metaheurísticos

La búsqueda en inteligencia artificial en algoritmos metaheurísticos es una técnica que combina técnicas de búsqueda y optimización para encontrar soluciones

óptimas a problemas complejos. Los algoritmos metaheurísticos son una clase de algoritmos que se utilizan para resolver problemas que son difíciles de resolver mediante técnicas de optimización convencionales. La búsqueda en inteligencia artificial se utiliza en estos algoritmos para explorar y evaluar diferentes soluciones hasta encontrar la mejor solución para el problema.

La búsqueda en inteligencia artificial puede ser guiada por una heurística, que es una función que evalúa la calidad de un estado o solución, o puede ser una búsqueda completa en la que se examinan todos los estados posibles. Los algoritmos metaheurísticos utilizan diferentes técnicas de búsqueda [22]:

- **Búsqueda en profundidad** implica explorar primero un estado o un nodo y luego profundizar en sus hijos o estados secundarios antes de examinar otros estados. Esta técnica se utiliza cuando se desea encontrar una solución en un espacio de estados con muchas ramificaciones, como un laberinto.
- **Búsqueda en anchura** implica explorar todos los hijos o estados secundarios de un estado antes de profundizar en cualquiera de ellos. Esta técnica se utiliza cuando se desea encontrar la solución más cercana a un estado inicial en un espacio de estados con muchas ramificaciones.
- **Búsqueda A*** el algoritmo asterisco o también conocido como A estrella, es una técnica de búsqueda que combina características de la búsqueda en profundidad y la búsqueda en anchura. La búsqueda A* utiliza una heurística para evaluar la calidad de los estados y elegir el camino más prometedor hacia la solución. La búsqueda A* se utiliza cuando se desea encontrar la solución óptima en un espacio de estados con muchas ramificaciones.

En resumen, estas tres técnicas de búsqueda se utilizan en diferentes situaciones para resolver problemas de búsqueda y encontrar soluciones óptimas. La elección de la técnica adecuada depende de la estructura del problema y de los objetivos de la búsqueda.

La búsqueda se realiza mediante un programa o agentes. El número de agentes necesarios en un algoritmo depende de varios factores como se ve en la Figura 1.3, tales como:



Figura 1.3: Clasificación algoritmos metaheurísticos.

En última instancia, la cantidad de agentes necesarios es una compensación entre el rendimiento deseado y los recursos computacionales disponibles. Puede requerir experimentación y pruebas para determinar la cantidad óptima de agentes para un problema y algoritmo específicos.

1.2. Aplicación de los algoritmos de optimización para las antenas

La optimización de las antenas es un aspecto fundamental en el diseño y desarrollo de sistemas de comunicaciones. La capacidad de una antena para transmitir y recibir señales de forma eficiente depende en gran medida de sus características físicas, como su tamaño, forma y material. Por lo tanto, para lograr un rendimiento óptimo en términos de cobertura, ganancia, directividad y polarización, es necesario realizar un proceso de optimización de las antenas.

En el campo de las telecomunicaciones, los algoritmos de optimización juegan un papel fundamental en el proceso de optimización de las antenas. Estos algoritmos utilizan técnicas matemáticas para analizar y mejorar las características de las antenas con el fin de lograr un rendimiento más eficiente. Algunos de los algoritmos de optimización más comúnmente utilizados son la programación genética, el algoritmo de optimización basado en enjambre de partículas y el algoritmo de optimización basado en colonias de hormigas, como otros que ya se mencionaron.

En la Figura 1.4 se describen algunos ejemplos de aplicaciones de los algoritmos de optimización en el diseño de antenas:



Figura 1.4: Aplicaciones de algoritmos en el diseño de antenas.

Estos son solo algunos ejemplos de cómo los algoritmos de optimización se utilizan en el diseño de antenas. En general, los algoritmos de optimización son una herramienta valiosa para mejorar el rendimiento y la eficiencia de las antenas, y son ampliamente utilizados en una variedad de aplicaciones.

1.3. Antenas tipo patch

Las antenas patch son un tipo de antena plana que consisten en un parche conductor en una placa dieléctrica que se monta sobre un sustrato conductor. El sustrato es un material aislante que actúa como un medio de soporte para la señal electromagnética. Estas antenas también son conocidas como antenas Microstrip, como su nombre lo dice son compactas, ligeras y fáciles de producir en masa, por lo que son populares para aplicaciones de telecomunicaciones y wireless, como en dispositivos de acceso a internet y telefonía móvil. Además, su diseño permite una alta ganancia

y direccionalidad, lo que las hace ideales para transmisiones de alta frecuencia en una sola dirección [23]. Las formas de los parches más comunes se pueden ver en la Figura 1.5 [1].

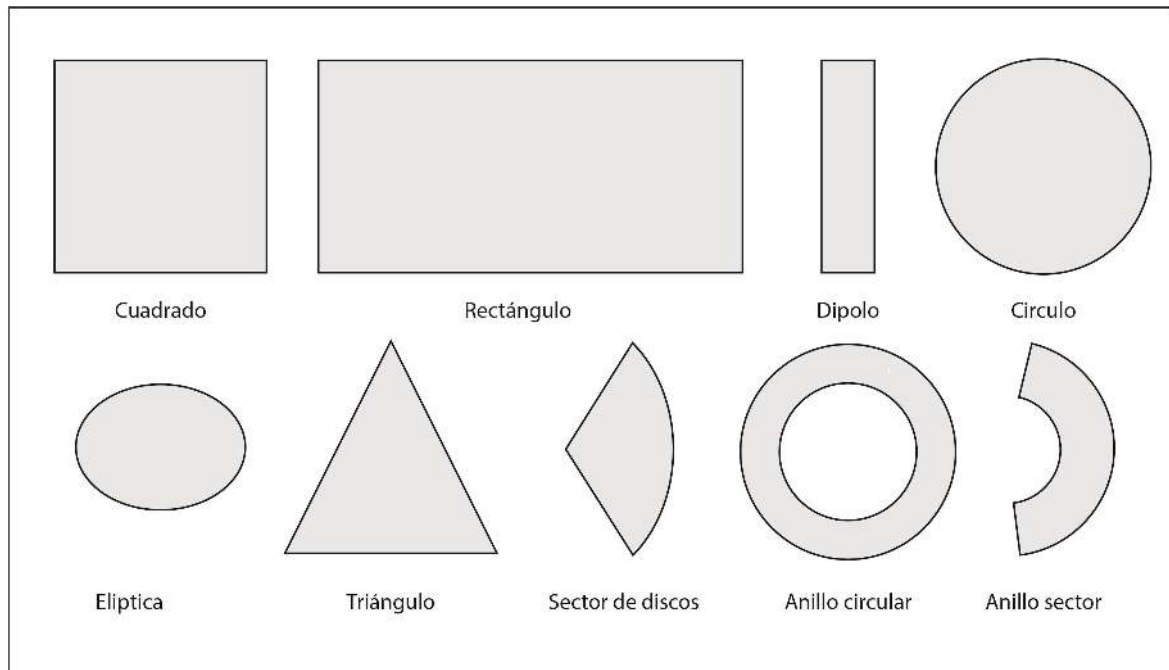


Figura 1.5: Formas representativas de los elementos de las antenas patch (Imagen adaptada de [1]).

Las antenas tipo patch tienen muchas ventajas que las hacen ideales para su uso en sistemas de comunicación. En primer lugar, son muy pequeñas y ligeras, lo que las hace muy fáciles de integrar en otros dispositivos electrónicos. Además, su tamaño compacto las hace muy discretas y poco obtrusivas, lo que las hace perfectas para aplicaciones en las que la estética es importante [24].

Otra ventaja importante de las antenas tipo patch es que su diseño les permite tener una direccionalidad muy precisa. Esto significa que la señal que transmiten o reciben es muy intensa en una dirección determinada, lo que reduce el riesgo de interferencia con otros dispositivos. Esta direccionalidad precisa también significa que las antenas tipo patch son ideales para aplicaciones en las que se requiere una señal fuerte y clara, como en sistemas de comunicación de larga distancia.

Además, las antenas tipo patch son muy fáciles de fabricar y requieren muy poco mantenimiento. Esto les permite ser muy económicas y accesibles para una amplia gama de usuarios, lo que las hace ideales para aplicaciones comerciales y

personales.

Sin embargo, también es importante mencionar que las antenas tipo patch tienen algunas desventajas. En primer lugar, su diseño les limita en cuanto a su capacidad de captar señales de frecuencias muy altas o muy bajas. Además, su pequeño tamaño también significa que su capacidad de transmisión de señal es limitada en comparación con otras antenas más grandes.

1.3.1. Patch rectangular

La antena patch rectangular es un tipo de antena plana que utiliza un parche conductor rectangular en lugar de una espiral o un dipolo. Esta antena es muy popular en aplicaciones de frecuencia ultra alta (UHF, del inglés Ultra High Frequency) y frecuencia super alta (SHF, del inglés Super High Frequency), ya que ofrece una alta ganancia, una directividad elevada y una impedancia de 50 ohmios, que es la misma que la mayoría de los dispositivos de transmisión y recepción [1].

Una de las principales ventajas de la antena patch rectangular es su facilidad de fabricación y su tamaño compacto. El parche conductor se puede fabricar con materiales relativamente baratos, y su tamaño compacto significa que es fácil de integrar en una amplia gama de dispositivos, incluyendo smartphones, laptops y dispositivos de transmisión inalámbricos.

Además, la antena patch rectangular es muy versátil, ya que se puede utilizar en una amplia gama de frecuencias y se puede diseñar para una amplia gama de aplicaciones, incluyendo transmisión y recepción de datos, comunicación móvil y sistemas de transmisión de televisión.

En términos de desventajas, la antena patch rectangular puede tener una frecuencia de resonancia limitada, lo que significa que solo es eficaz en un rango de frecuencias muy específico. También puede ser sensible a las interferencias electromagnéticas, lo que puede afectar su rendimiento.

En resumen, la antena patch rectangular es una solución popular y eficaz para aplicaciones de frecuencia UHF y SHF. Con su tamaño compacto, facilidad de fabricación y versatilidad, es una opción popular para una amplia gama de aplicaciones de transmisión y recepción de datos.

Formulación del patch rectangular

La formulación matemática de una antena patch rectangular es una combinación de cálculos electromagnéticos y de ingeniería de materiales. A continuación, en la Fig 1.6 se describen algunas de las fórmulas más importantes utilizadas en la formulación de una antena patch rectangular:

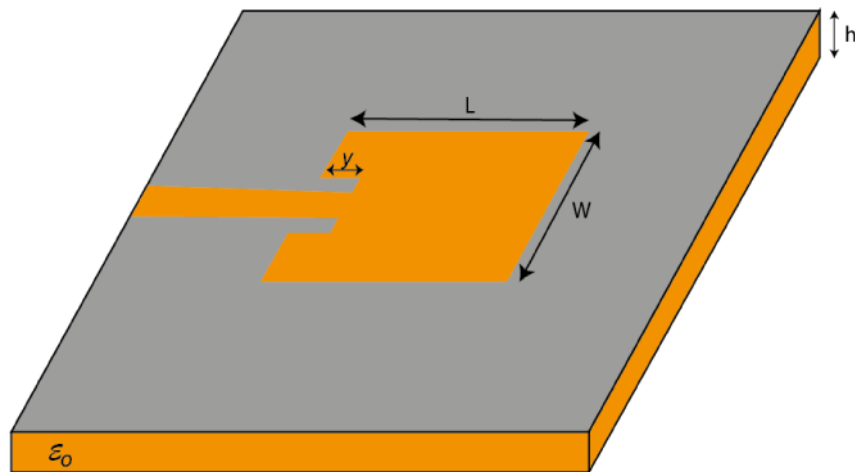


Figura 1.6: Antena patch rectangular.

Para las formulas que se van usar se deben tomar en cuenta la siguiente Tabla 1.1:

Tabla 1.1: Sustratos para antenas patch

Sustrato	Constante dieléctrica: ϵ_r
Epoxy fiberglass FR-4	4.4
Rohacell Foam	1.07
Honeycomb	1.02
Taconinc	2.33
Kapton	3.5
CuClad	2.17
RT Duroid 5880	2.2
RT Duroid 6010	10.5
GaAs	13

- Anchura del patch (W): es la medida de la longitud de una de las dimensiones de la superficie reflectora de la antena. Es un factor importante en el diseño de la antena ya que afecta directamente su frecuencia de resonancia y su ganancia. Un ancho más grande puede resultar en una mayor ganancia y una menor

frecuencia de resonancia, mientras que un ancho más pequeño puede resultar en una ganancia más baja y una frecuencia de resonancia más alta.

$$W = \frac{\lambda_o}{2} \cdot \sqrt{\frac{2}{\epsilon_o + 1}} \quad (1.1)$$

- Longitud del patch (L): La longitud de una antena patch es la medida de la otra dimensión de la superficie reflectora de la antena, perpendicular al ancho. Es un factor crítico en el diseño de la antena ya que determina su frecuencia de resonancia y su ganancia. Una longitud más larga puede resultar en una frecuencia de resonancia más baja y una mayor ganancia, mientras que una longitud más corta puede resultar en una frecuencia de resonancia más alta y una ganancia más baja. La longitud de la antena también puede ser ajustada para ajustar la banda de frecuencia de la antena.

$$L = L_{eff} - 2\Delta L \quad (1.2)$$

Donde:

La longitud eléctrica del patch L_{eff} (media onda) es:

$$L_{eff} = \frac{\lambda_g}{2} = \frac{\lambda_o}{2\sqrt{\epsilon_{ef}}} \quad (1.3)$$

El incremento de L (ΔL) esta definido por:

$$\Delta L = 0,412h \frac{(\epsilon_{ef} + 0,3)\left(\frac{W}{h} + 0,264\right)}{(\epsilon_{ef} - 0,258)\left(\frac{W}{h} + 0,8\right)} \quad (1.4)$$

- La inserción (Y_o): para una resistencia de entrada adecuada (por ejemplo 50 Ω). La formula para 50 Ω sería:

$$Y_o = \frac{L \times \arccos\left(\sqrt{\frac{50}{R_{in}(Y=0)}}\right)}{\pi} \quad (1.5)$$

Plano de tierra L_g

$$L_g = 6h + L \quad (1.6)$$

Plano de tierra W_g

$$W_g = 6h + W \quad (1.7)$$

El campo de la optimización ha experimentado un crecimiento exponencial en las últimas décadas, gracias al desarrollo de nuevas técnicas y algoritmos. Entre las diferentes metodologías de optimización existentes, los algoritmos metaheurísticos han demostrado ser altamente efectivos para resolver problemas complejos en diversas áreas.

Los algoritmos metaheurísticos son técnicas de búsqueda basadas en la naturaleza que se inspiran en los procesos biológicos y físicos observados en la naturaleza. Estos algoritmos son capaces de encontrar soluciones óptimas o casi óptimas en un tiempo razonable, incluso para problemas de gran escala.

Entre los tipos más comunes de algoritmos metaheurísticos se encuentran los algoritmos evolutivos, los algoritmos de colonia de hormigas, los algoritmos de enjambre de partículas, y los algoritmos de búsqueda tabú, por nombrar algunos.

En cuanto a la aplicación de los algoritmos de optimización en el campo de las antenas, se han utilizado para diseñar antenas de forma más eficiente. Los algoritmos metaheurísticos pueden optimizar los parámetros de las antenas, como su tamaño y forma, para maximizar su rendimiento.

En particular, las antenas patch son una opción popular en aplicaciones de comunicación, ya que son livianas, de bajo costo y pueden ser diseñadas para funcionar en una amplia gama de frecuencias. La antena patch rectangular es una de las formas más comunes de antena patch y se puede diseñar utilizando diversas formulaciones matemáticas, como la teoría de circuitos o la teoría electromagnética.

En resumen, los algoritmos metaheurísticos son una herramienta poderosa para resolver problemas complejos de optimización, y su aplicación en el diseño de antenas ha mejorado significativamente su rendimiento. La antena patch rectangular es una de las formas más populares de antenas patch, y su formulación matemática puede ser determinada utilizando varias técnicas de optimización.

Capítulo 2

Criterios de diseño e implementación

El presente capítulo se enfoca en los criterios de diseño e implementación de diferentes algoritmos de optimización en una aplicación de MATLAB. Los algoritmos que se discuten incluyen PSO, Firefly Algorithm, MPA, HHO, SMA y Bat Algorithm. Se explora sus características, fortalezas y debilidades, así como las técnicas y métodos que se deben seguir para implementarlos de manera efectiva en una interfaz de MATLAB. El objetivo de este capítulo es proporcionar una guía clara y precisa para los lectores interesados en utilizar estos algoritmos para resolver problemas de optimización en un entorno MATLAB.

2.1. Algoritmos basados en inteligencia de enjambre.

La inteligencia de enjambre es una técnica de optimización inspirada en el comportamiento de las colonias de insectos, como las abejas y las hormigas. Estos algoritmos funcionan mediante la imitación de los procesos de toma de decisiones en un enjambre, donde cada individuo o "partícula" toma decisiones basadas en la información obtenida de sus compañeros.

El estado del arte de los algoritmos basados en inteligencia de enjambre es muy activo, con una gran cantidad de investigaciones en curso en diversas áreas, incluyendo la optimización, la resolución de problemas, la inteligencia artificial y la automatización.

Una de las áreas de investigación más prometedoras en el campo de la

inteligencia de enjambre es la optimización global. Los algoritmos de inteligencia de enjambre son muy eficaces para encontrar soluciones óptimas a problemas complejos, gracias a su capacidad para explorar y explotar diferentes soluciones simultáneamente. Esto ha llevado a una gran cantidad de aplicaciones en áreas como la ingeniería, la economía y la finanzas.

Otra área de investigación importante es la aplicación de la inteligencia de enjambre a la resolución de problemas en tiempo real. Estos algoritmos son muy eficaces para solucionar problemas con una gran cantidad de variables y restricciones, lo que los hace ideales para aplicaciones en sistemas de control y automatización.

Además, la inteligencia de enjambre se está aplicando cada vez más a la inteligencia artificial y la aprendizaje automático. Por ejemplo, se están desarrollando algoritmos basados en inteligencia de enjambre que pueden utilizarse para clasificar grandes cantidades de datos y mejorar la precisión de las predicciones.

Entonces, los algoritmos basados en inteligencia de enjambre es muy prometedor, con una amplia gama de investigaciones en curso y una gran cantidad de aplicaciones en diferentes áreas. A medida que se desarrollan nuevas técnicas y se mejoran los algoritmos existentes, es probable que la inteligencia de enjambre siga siendo una técnica clave en la optimización y la resolución de problemas en el futuro.

2.1.1. Algoritmos implementados

Bat Algorithm

Es un algoritmo de optimización basado en la población que utiliza las ondas de sonido producidas por los murciélagos para localizar presas y navegar en el entorno. En el algoritmo BAT, cada murciélago representa una solución candidata al problema de optimización y su posición se actualiza en función de una combinación de información local y global. El algoritmo utiliza un enfoque probabilístico para equilibrar la exploración y la explotación del espacio de soluciones y se ha aplicado a varios problemas de optimización, como la optimización de funciones, la optimización multiobjetivo y la optimización de restricciones. El algoritmo BAT es conocido por su simplicidad, robustez y eficiencia en comparación con otros algoritmos de

optimización.

Una de las ventajas del algoritmo BAT es que utiliza una función de evaluación global y una función de actualización para mejorar la búsqueda de soluciones. Además, el algoritmo puede adaptarse a diferentes tipos de problemas y es capaz de manejar problemas con múltiples restricciones y objetivos. En general, el Algoritmo de los Murciélagos es una herramienta útil para la optimización de problemas complejos y ha sido aplicado con éxito en una amplia variedad de áreas, incluyendo ingeniería, ciencias de la computación y biología.

Las siguientes son las fases generales del algoritmo BAT:

1. **Inicialización:** se genera una población inicial de soluciones aleatorias para el problema de optimización.
2. **Evaluación:** se evalúa la calidad de cada solución de la población.
3. **Actualización:** se actualiza la posición de cada murciélago en el espacio de búsqueda en función de la calidad de su solución.
4. **Selección:** se seleccionan las soluciones de mejor calidad para crear una nueva población.
5. **Reproducción:** se utiliza un proceso de reproducción para crear nuevas soluciones basadas en las soluciones existentes.
6. **Criterio de parada:** el algoritmo se detiene cuando se alcanza un criterio de parada específico, como un número máximo de iteraciones o cuando se encuentra una solución satisfactoria.

El algoritmo BAT es un enfoque de optimización eficiente y flexible que puede ser utilizado para resolver una amplia variedad de problemas de optimización. Además, es un enfoque computacionalmente eficiente y puede ser paralelizado para resolver problemas más grandes y complejos.



Figura 2.1: Comportamiento del murciélago.

Algoritmo 1 BAT algorithm**Entrada**

Población de murciélagos $x_i = (x_{i1}, \dots, X_{iD})^T$ for $i = 1 \dots N_p$, MAX FE.

Output:

La mejor solución x_{best} y esta corresponde al valor $f_{min} = \min(f(x))$.

iniciar bat();

evaluar = evaluar la nueva población;

f_{min} = encuentra la mejor solución (x_{best}); inicialización

while no se cumple la condición de finalización **do**;

for $j=1$ to N_p **do**

y = generar una mejor solución (x_i);

if $\text{rand}(0,1) > r_i$ **then**

y = mejorar la mejor solución x_{best}

end if etapa de búsqueda local

$f_{new} = \text{Evalualamejorsolucion}(y)$;

 evaluar = evaluar + 1;

if $f_{new} \leq f_i$ **and** $N(0,1) < A$ **then**

$x_i = y$; $f_i = f_{new}$;

end if guardar la mejor solución condicionalmente

f_{min} = encuentra la mejor solución (x_{best});

end for

end while

$$v_i(t+1) = v_i(t) + \omega(t)(x_i(t) - x_i(t-1)) + \gamma(t)(x_{rand} - x_i(t)) \quad (2.1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2.2)$$

$$T(t+1) = \frac{1}{2}(T(t) \cdot \alpha(t)) \quad (2.3)$$

$$f_{best}(t+1) = \min_{1 \leq i \leq n} \{f_i(t+1)\} \quad (2.4)$$

$$x_{best}(t+1) = \operatorname{argmin}_{1 \leq i \leq n} \{f_i(t+1)\} \quad (2.5)$$

Donde $v_i(t)$ es la velocidad de la partícula i en el tiempo t , $\omega(t)$ es la frecuencia de la vibración de la partícula, $\gamma(t)$ es la amplitud de la vibración, x_{rand} es un punto aleatorio en el espacio de búsqueda, $x_i(t)$ es la posición de la partícula i en el tiempo t , $T(t)$ es la temperatura inversa, $\alpha(t)$ es la tasa de enfriamiento, $f_{best}(t)$ es la función objetivo mínima en el tiempo t , y $x_{best}(t)$ es la posición correspondiente a la función objetivo mínima en el tiempo t .

Fireflys Algorithm

Algoritmo de las luciérnagas (FA, del inglés Firefly Algorithm) es un algoritmo de optimización metaheurística inspirado en el comportamiento intermitente de las luciérnagas. Es un algoritmo de optimización basado en la población que utiliza la emisión de luz de las luciérnagas para atraer parejas y localizar presas. En el algoritmo Firefly, cada luciérnaga representa una solución candidata al problema de optimización y su posición se actualiza en función de su brillo. El brillo de cada luciérnaga está determinado por el valor de la función objetivo y se usa para comparar y clasificar las soluciones candidatas. El algoritmo utiliza un enfoque de recorrido aleatorio para equilibrar la exploración y la explotación del espacio de la solución y se ha aplicado a varios problemas de optimización, como la optimización de funciones, la optimización multiobjetivo y la optimización de restricciones. El algoritmo Firefly es conocido

por su simplicidad, flexibilidad y eficacia en comparación con otros algoritmos de optimización.

La formulación del algoritmo incluye los siguientes pasos:

1. **Inicialización:** se genera una población inicial de soluciones aleatorias para el problema de optimización.
2. **Evaluación:** se evalúa la calidad de cada solución de la población.
3. **Actualización:** se actualiza la posición de cada luciérnaga en el espacio de búsqueda en función de la calidad de su solución.
4. **Reproducción:** se utiliza un proceso de reproducción para crear nuevas soluciones basadas en las soluciones existentes.
5. **Criterio de parada:** el algoritmo se detiene cuando se alcanza un criterio de parada específico, como un número máximo de iteraciones o cuando se encuentra una solución satisfactoria.

El algoritmo de las luciérnagas es un enfoque de optimización eficiente y flexible que puede ser utilizado para resolver una amplia variedad de problemas de optimización. Además, es un enfoque computacionalmente eficiente y puede ser paralelizado para resolver problemas más grandes y complejos.

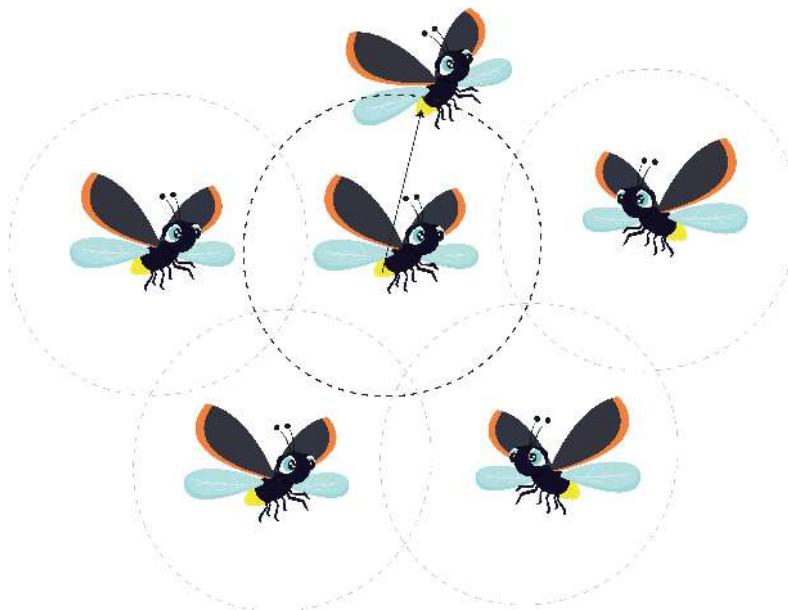


Figura 2.2: Comportamiento de las luciérnagas.

Algoritmo 2 Firefly algorithm

Función objetivo $f(x)$, $x = (x_1, \dots, x_d)^T$
 Población inicial genérica de luciérnagas $x_i (i = 1, 2, \dots, n)$
 Intensidad de la luz I_i en x_i es determinada por $f(x_i)$
 Definir el coeficiente de absorción de luz γ
while ($t < MaxGeneration$)
for $i = 1 : n$ al n luciérnagas
 for $j = 1 : n$ todas n luciérnagas (bucle interior)
 if ($I_i < I_j$), Mover luciérnaga i hacia j ;
 end if
 Variar la atracción con la distancia r vía $exp[-\lambda r]$
 Evaluar nuevas soluciones y actualizar la intensidad de la luz
 end for j
end for i
 Clasifica las luciérnagas y encuentra la mejor global g_*
end while
 Visualización y resultados posteriores al proceso

$$\text{firefly lighth} = \beta_0 e^{-\gamma r} \quad (2.6)$$

$$\text{Best Pos} = x_i(t+1) = x_i(t) + \beta(t) (\text{best firefly} - \text{firefly lighth}) \quad (2.7)$$

$$\beta(t+1) = \beta(t) \cdot \alpha \quad (2.8)$$

$$\gamma(t+1) = \gamma(t) \cdot \alpha \quad (2.9)$$

Donde β_0 es la intensidad inicial de la luz de la luciérnaga, γ es la constante de atenuación, r es la distancia entre la luciérnaga y la luciérnaga más brillante, $\beta(t)$ es la intensidad de la luz de la luciérnaga en el tiempo t , α es la tasa de decaimiento, y $x_i(t)$ es la posición de la luciérnaga i en el tiempo t .

Harris Hawk Optimization

Optimización Harris Hawks (HHO, del inglés Harris Hawk Optimization) es un algoritmo de optimización metaheurística inspirado en el comportamiento de caza cooperativa de Harris Hawks. Es un algoritmo de optimización basado en la población donde cada ave representa una solución candidata al problema de optimización. La posición y la velocidad de cada ave se actualizan iterativamente en función de su posición actual, la mejor posición que ha encontrado hasta el momento y la mejor

posición encontrada por toda la población. El algoritmo utiliza una combinación de búsqueda local y global para equilibrar la exploración y la explotación del espacio de la solución. El algoritmo de optimización de Harris Hawk se ha aplicado a varios problemas de optimización, como la optimización de funciones, la agrupación de datos y la selección de funciones. El algoritmo de optimización de Harris Hawk es conocido por su simplicidad, versatilidad y eficacia en comparación con otros algoritmos de optimización.

HHO tiene 3 fases:

- **Fase de inicialización:** en esta fase se definen los parámetros iniciales, como el tamaño de la población, el número de generaciones, el criterio de selección y la función de evaluación.
- **Fase de evolución:** en esta fase se lleva a cabo la selección y la mutación de los individuos en la población. La selección se basa en la función de evaluación y la mutación en el cambio aleatorio de los valores de los genotipos de los individuos.
- **Fase de evaluación:** en esta fase se evalúa la solución óptima obtenida a través de la función de evaluación. Si la solución es satisfactoria, se detiene el algoritmo y se presenta la solución óptima, de lo contrario se vuelve a la fase de evolución y se repiten los procesos hasta que se encuentre la solución óptima.

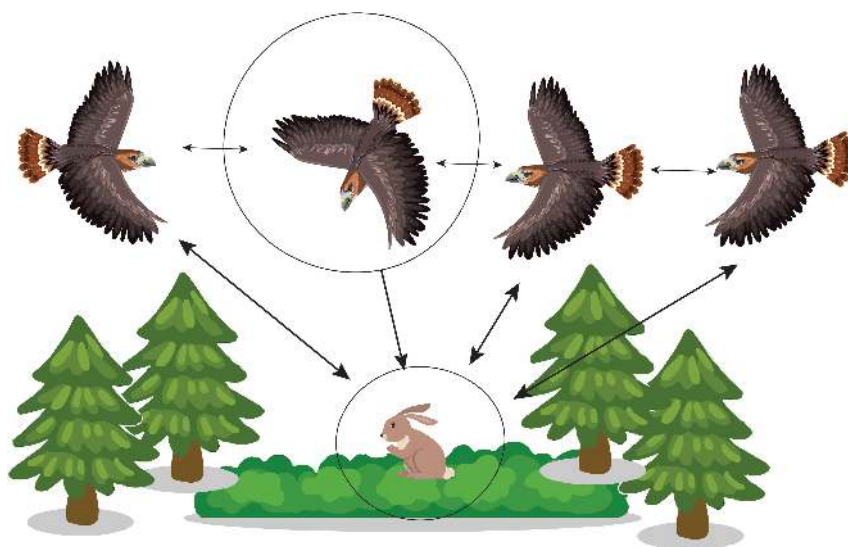


Figura 2.3: Comportamiento del Halcón de Harris.

Algoritmo 3 Pseudocódigo Harrys Hawk Optimization**Definiciones****Entradas:**

Tamaño de Población N y el numero máximo de iteraciones T .

Salida:

Ubicación del conejo con su valor fitness

Inicializa la población aleatoria $X_i (i = 1, 2, \dots, N)$

while (El criterio de paro no cumple) **do**

Calcular valor del fitness de los halcones

Mover X_{rabbit} a la mejor locación del conejo (best location)

for (cada halcón X_i) **do**

Actualizar energía inicial E_0 y saltar a J

Actualizar E

if $|E| \geq 1$ **then**

Actualizar la ubicación del vector

if $|E| < 1$ **then**

if ($r \geq 0,5$ and $|E| \geq 0,5$)

Actualizar la ubicación del vector

elseif ($r \geq 0,5$ and $|E| < 0,5$) **then**

if ($r \geq 0,5$ and $|E| \geq 0,5$)

Actualizar la ubicación del vector)

elseif ($r < 0,5$ and $|E| \geq 0,5$) **then**

Actualizar la ubicación del vector)

elseif ($r < 0,5$ and $|E| < 0,5$) **then**

Actualizar la ubicación del vector

Return X_{rabbit}

$$\text{Actualización de la posición} = x_i(t+1) = x_i(t) + \frac{\Delta_i(t)}{\|\Delta_i(t)\|} \quad (2.10)$$

$$\Delta_i(t) = x_{best}(t) - x_i(t) + \frac{\|x_{best}(t) - x_i(t)\|}{\|x_{global_best}(t) - x_i(t)\|} \quad (2.11)$$

$$\text{Selección de líder} = x_{best}(t+1) = \operatorname{argmax}_{1 \leq i \leq n} f(x_i(t+1)) \quad (2.12)$$

$$\text{Actualización global} = x_{global_best}(t+1) = \operatorname{argmin}_{1 \leq i \leq n} f(x_i(t+1)) \quad (2.13)$$

Donde $x_i(t)$ es la posición de la partícula i en el tiempo t , $\Delta_i(t)$ es el vector de cambio en la posición de la partícula i en el tiempo t , $x_{best}(t)$ es la posición de la partícula líder en el tiempo t , $x_{global_best}(t)$ es la posición global mínima en el tiempo t , y f es la función objetivo.

Marine Predator Algorithm

Marine Predator Algorithm (MPA) es un algoritmo de optimización metaheurística inspirado en el comportamiento de caza de los depredadores marinos. Es un algoritmo de optimización basado en la población que utiliza los principios de la interacción depredador-presa para buscar la solución óptima a un problema determinado. En el algoritmo del depredador marino, los depredadores representan las soluciones candidatas y las presas representan los valores de la función objetivo. Los depredadores actualizan sus posiciones en función de la posición de la presa, con el objetivo de maximizar el valor de la función objetivo. El MPA utiliza una combinación de exploración y explotación para equilibrar la búsqueda del espacio de solución. El MPA se ha aplicado a varios problemas de optimización, como la optimización de funciones, la agrupación de datos y la selección de características. El MPA es conocido por su capacidad para superar la convergencia prematura y su capacidad para encontrar soluciones óptimas globales de manera eficiente.

Antes de ver las Fases que tiene el MPA debemos tener en cuenta lo que son los Movimiento Brownianos y de Levy, ya que en base a estos movimientos; se mueven las presas. En la Figura 2.4, se define los dos movimientos.

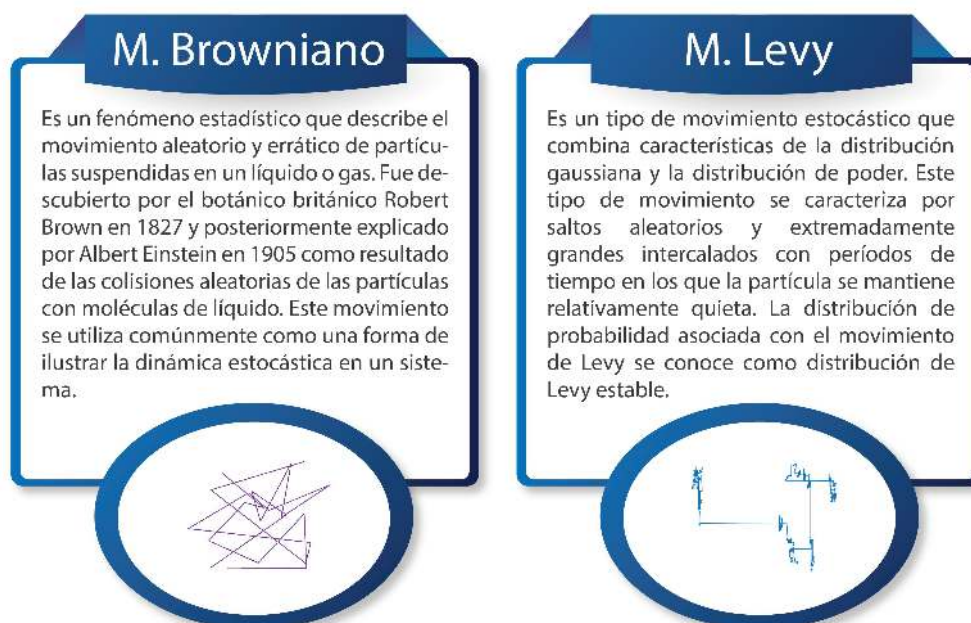


Figura 2.4: Movimiento Browniano y movimiento de Levy.

Fases de exploración y explotación del MPA, como se ve en la Figura 2.5 consta de 3 fases:

- **Fase 1** Durante la primera etapa de optimización, las presas se mueven siguiendo el movimiento browniano. Debido a que las presas están distribuidas uniformemente en el área de búsqueda en las primeras iteraciones y la distancia entre el depredador y la presa es relativamente amplia, este movimiento aleatorio puede permitir a las presas explorar su entorno de manera individual, lo que resulta en una buena exploración del área. Luego, se evalúa la aptitud de la presa en una nueva posición y se reemplaza si es más ajustada que la anterior. Las posiciones ajustadas de las presas se pueden entender como áreas con alimento abundante, y el proceso de almacenamiento es equivalente a la memoria de las presas para recordar estos lugares. Si una presa tiene más éxito en la búsqueda de alimento, puede ser considerada un depredador. Esto significa que se calcula su aptitud y se reemplaza al depredador superior si es más ajustado.
- **Fase 2** Ahora es el momento en que los depredadores comienzan a buscar alimento mientras que las presas continúan haciendo lo mismo. Este es el inicio de la segunda etapa de optimización. La fase 2 mostrada en la figura muestra esquemáticamente esta fase de optimización en la que el depredador empieza a dar pasos brownianos en trayectoria azul para buscar mejor en el dominio mientras que presa se mueve en la estrategia de Lévy mostrada por los paseos verdes. Dado que las ubicaciones del depredador y la presa y la longitud de los pasos será menor que en la fase anterior, El efecto FAD junto con los pasos largos de la estrategia de Lévy ayuda a MPA a evitar el estancamiento de los óptimos locales y a mejorar el rendimiento. y mejorar el rendimiento del método.
- **Fase 3** Cuando la optimización llega a su fase final, el algoritmo necesita una gran capacidad de explotación. En esta fase, el depredador empieza a cambiar su comportamiento de la estrategia browniana a la de Lévy para buscar de forma más eficiente en un determinado vecindario. El factor de convergencia (CF) definido de forma adaptativa en esta fase ayuda en gran medida a los depredadores a limitar las áreas de búsqueda dentro de un determinado

vecindario para su explotación y también a evitar malgastar parte del esfuerzo de búsqueda derivado de los largos tamaños de paso de la estrategia Lévy en regiones no prometedoras del dominio. La última fase se representa en la figura como Phase 3.

Algoritmo 4 Pseudocódigo Marine Predators Algorithm

Inicio

Inicializar la población de agentes de búsqueda (Prey) $i = 1, 2, \dots, n$

while (El criterio de paro no cumple)

Calcular valor fitness y construir la matriz Elite

if $iter < Iter_{max}$

Actualizar prey

elseif ($Iter_{max}/3 < Iter < 2 * Iter_{max}/3$)

for Mitad de poblaciones ($i = 1, 2, \dots, n/2$)

Actualizar prey

for Resto de poblaciones ($i = n/2, \dots, n$)

Actualizar prey

elseif ($Iter > 2 * Iter_{max}/3$)

Actualizar prey

endif

Guardar la memoria y actualizar el Elite

Aplicar efecto FADs y actualizar

Guardar la memoria y actualizar el Elite

end while

$$\text{Actualización de la posición} = x_i(t+1) = x_i(t) + \text{Función de aceleración} \quad (2.14)$$

$$\text{Función de aceleración} = \sum_{j=1}^n w_{ij}(t) (x_{pbest}(j, t) - x_i(t)) \quad (2.15)$$

$$w_{ij}(t) = \frac{1}{\|x_j(t) - x_i(t)\|^2 + \text{Función de penalización}} \quad (2.16)$$

$$w_{ij}(t) = \frac{1}{\|x_j(t) - x_i(t)\|^2 + \text{Función de penalización}} \quad (2.17)$$

$$\text{Selección de líder} = x_{pbest}(i, t+1) = \operatorname{argmin}_{1 \leq j \leq n} f(x_j(t)) \quad (2.18)$$

$$\text{Actualización global} = x_{gbest}(t+1) = \operatorname{argmin}_{1 \leq i \leq n} f(x_{pbest}(i, t)) \quad (2.19)$$

Donde $x_i(t)$ es la posición de la partícula i en el tiempo t , $w_{ij}(t)$ es el peso en el tiempo t que se aplica a la partícula j por la partícula i , $x_{pbest}(j, t)$ es la posición personal mejor de la partícula j en el tiempo t , $x_{gbest}(t)$ es la posición global mejor en el tiempo t , y f es la función objetivo.

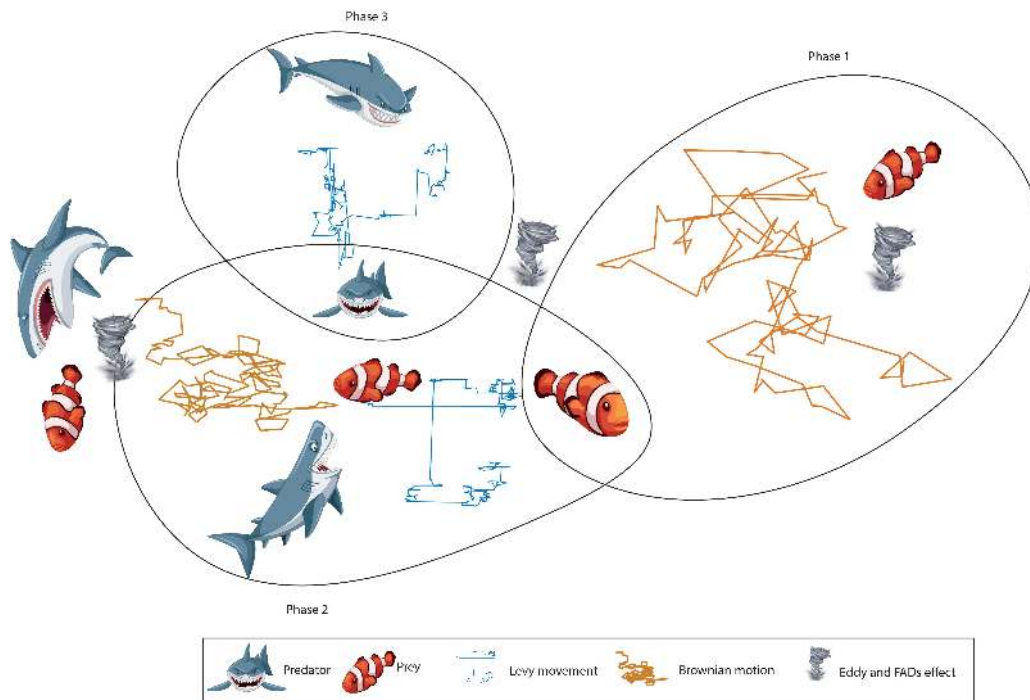


Figura 2.5: Las tres fases de optimización del MPA.

Particle Swarm Optimization

Particle Swarm Optimization (PSO) es un algoritmo de optimización basado en inteligencia de enjambre inspirado en el comportamiento social de las aves. Es un algoritmo de optimización basado en la población que imita el comportamiento de un grupo de aves en busca de alimento. En PSO, cada partícula en la población representa una solución candidata al problema de optimización. La posición de cada partícula se actualiza en función de su propia mejor solución (pbest) y la mejor solución encontrada hasta el momento por toda la población (gbest). La fórmula de actualización de posición para cada partícula es un promedio ponderado de su velocidad actual, su propia mejor solución y la mejor solución global. El algoritmo repite el proceso de actualización de la posición de cada partícula hasta que se cumple un criterio de parada, como un número máximo de iteraciones o una tolerancia de error mínima.

PSO se ha aplicado a una amplia gama de problemas de optimización, incluida la optimización de funciones, el aprendizaje automático y los sistemas de control.

Según el algoritmo PSO, el conjunto de partículas de un proceso puede presentarse según las siguientes ecuaciones:

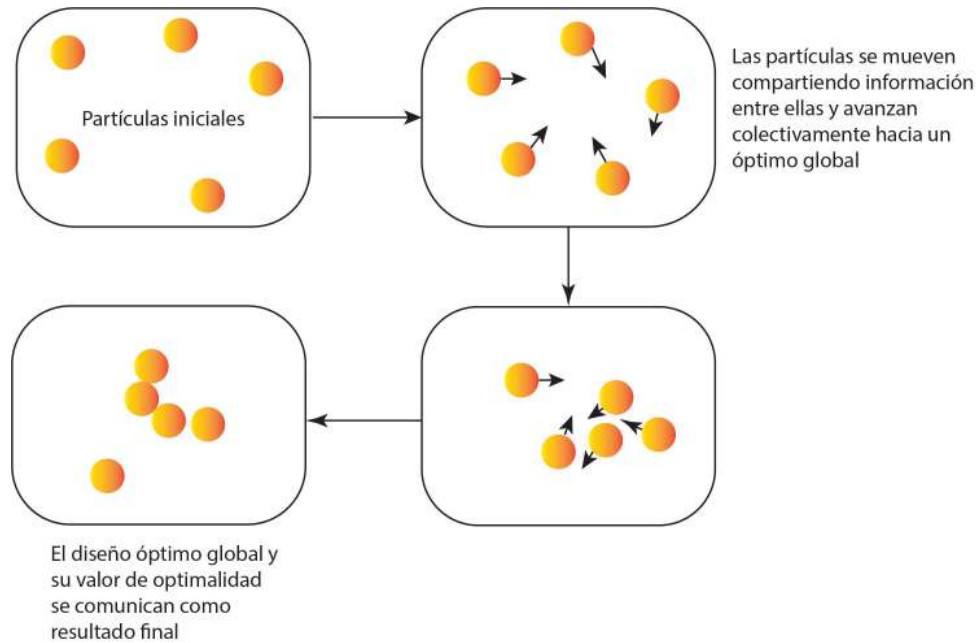


Figura 2.6: Ilustración del algoritmo PSO.

Algoritmo 5 Pseudocódigo Particle Swarm Optimization

Inicio

for ($t = 1$: máxima generación)

for ($i = 1$: tamaño de población)

if $f(x_{i,d}(t)) < f(p_i(t))$ **then** $p_i(t) = x_{i,d}(t)$

$f(p_g(t)) = \min(f(p_i(t)))$

end

for $d = 1$: dimensión

$v_{i,d}(t+1) = \omega v_{i,d}(t) + c_1 r_1 (p_i - X_{i,d}(t)) + c_2 r_2 (p_g - x_{i,d}(t))$

$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1)$

if $v_{i,d}(t+1) > v_{max}$ **then** $v_{i,d}(t+1) = v_{max}$

else if $v_{i,d}(t+1) < v_{min}$ **then** $v_{i,d}(t+1) = v_{min}$

end

if $x_{i,d}(t+1) > x_{max}$ **then** $x_{i,d}(t+1) = x_{max}$

else if $x_{i,d}(t+1) < x_{min}$ **then** $x_{i,d}(t+1) = x_{min}$

end

end

end

end

$$\text{Actualizar velocidad} = v_i(t+1) = wv_i(t) + c_1r_1 (pbest_i(t) - x_i(t)) + c_2r_2 (gbest(t) - x_i(t)) \quad (2.20)$$

$$\text{Actualizar posición} = x_i(t+1) = x_i(t) + v_i(t+1) \quad (2.21)$$

$$\text{Selección de líder} = pbest_i(t+1) = \operatorname{argmin}_{x_i(t)} f(x_i(t)) \quad (2.22)$$

$$\text{Actualizar global} = gbest(t+1) = \operatorname{argmin}_{1 \leq i \leq n} f(pbest_i(t)) \quad (2.23)$$

Donde $v_i(t)$ es la velocidad de la partícula i en el tiempo t , w es una constante de inercia, c_1 y c_2 son constantes de aceleración, r_1 y r_2 son valores aleatorios en el intervalo $[0, 1]$, $pbest_i(t)$ es la posición personal mejor de la partícula i en el tiempo t , $gbest(t)$ es la posición global mejor en el tiempo t , y f es la función objetivo.

Slime Mould Algorithm

El algoritmo Slime Mould (SMA, del inglés Slime Mould Algorithm) es un algoritmo de optimización que se inspira en el comportamiento del moho limoso. Es un algoritmo de optimización basado en la población que utiliza los principios de crecimiento y agregación de moho para buscar la solución óptima a un problema dado.

La idea básica detrás del SMA es utilizar modelos matemáticos para simular el movimiento y crecimiento del moho líquido, y luego optimizar la solución al problema utilizando el comportamiento del moho líquido. Las ecuaciones específicas utilizadas en el SMA pueden variar dependiendo de la implementación, pero generalmente implican el modelamiento del movimiento y agregación del moho líquido utilizando funciones matemáticas.

Por ejemplo, el movimiento del moho líquido puede ser modelado mediante ecuaciones que describen el crecimiento y la propagación del moho líquido en respuesta a diversos factores ambientales, como la luz, las fuentes de alimento y otros estímulos físicos y químicos. La agregación del moho líquido puede ser modelada mediante ecuaciones que describen la atracción y repulsión entre los mohos líquidos individuales a medida que entran en contacto unos con otros.

Una vez que se han modelado el movimiento y la agregación del moho líquido,

el problema de optimización puede resolverse utilizando el comportamiento del moho líquido para encontrar la solución óptima. Por ejemplo, la función objetivo que se debe optimizar puede ser asignada al movimiento y agregación del moho líquido, y el algoritmo puede diseñarse para encontrar la solución que resulte en el valor máximo o mínimo de la función objetivo.

En general, el SMA es un algoritmo de optimización flexible y eficiente que se ha aplicado a una amplia variedad de problemas de optimización, incluyendo problemas en los campos de la ingeniería, la física, la química y la biología.

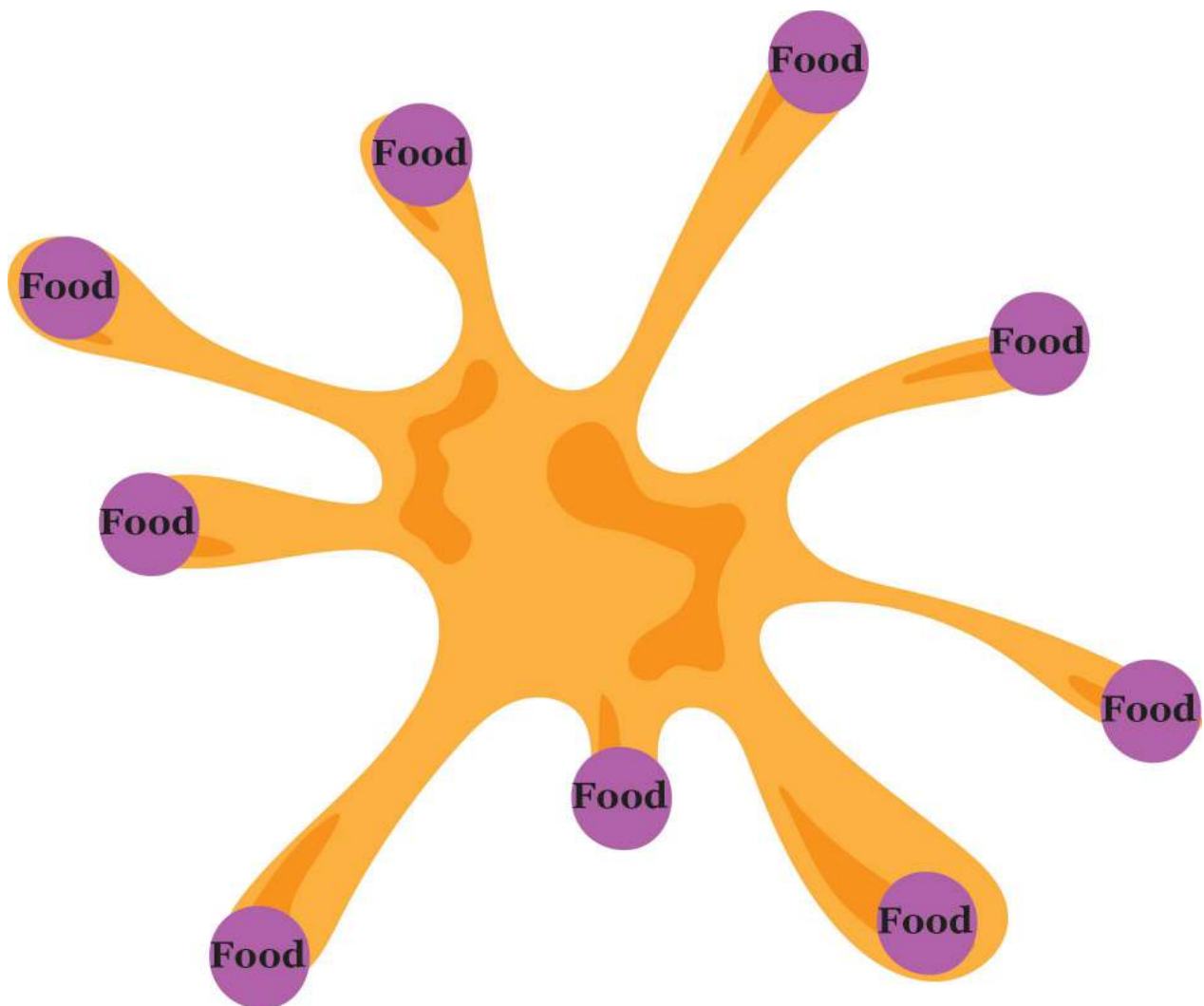


Figura 2.7: morfología de forrajeo del slime mould.

Las ecuaciones que describen el comportamiento del moho de limo en el algoritmo SMA incluyen:

Ecuación de movimiento: describe la dirección y velocidad de la masa de moho

Algoritmo 6 Pseudocódigo Slime Mould Algorithm**Definiciones****Entradas:**

Tamaño de Población N y el numero máximo de iteraciones T .

Posiciones del slime $X_i (i = 1, 2, \dots, N)$

while ($t \geq$ Iteración Máxima)

 Calcular valor del fitness de todos los slimes

 Actualizar el mejor valor Fitness X_b

 Calcular valor de W

for (cada sección de búsqueda)

 Actualizar p, vb, vc

 Actualizar las posiciones

End For

$t = t + 1$

End While

Return mejor Fitness X_b

de limo en función de la concentración de nutrientes en su entorno.

$$v_i^{t+1} = v_i^t + c_1 \cdot \text{rand}_1(\mathbf{p}_i^t - \mathbf{x}_i^t) + c_2 \cdot \text{rand}_2(\mathbf{p}_g^t - \mathbf{x}_i^t) \quad (2.24)$$

Donde:

v_i^{t+1} es la velocidad de la masa de moho de limo en el tiempo $t + 1$ v_i^t es la velocidad de la masa de moho de limo en el tiempo t c_1 es una constante de ajuste rand_1 es un número aleatorio generado en el intervalo $[0,1]$ \mathbf{p}_i^t es la posición más cercana de una fuente de alimento para la masa de moho de limo en el tiempo t \mathbf{x}_i^t es la posición actual de la masa de moho de limo en el tiempo t c_2 es otra constante de ajuste rand_2 es otro número aleatorio generado en el intervalo $[0,1]$ \mathbf{p}_g^t es la posición más cercana de una fuente global de alimento para la masa de moho de limo en el tiempo t

Ecuación de crecimiento: describe la forma en que la masa de moho de limo aumenta su tamaño y peso en función de la cantidad de nutrientes disponibles.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2.25)$$

Donde:

\mathbf{x}_i^{t+1} es la posición de la masa de moho de limo en el tiempo $t + 1$ \mathbf{x}_i^t es la posición de la masa de moho de limo en el tiempo t \mathbf{v}_i^{t+1} es la velocidad de la masa de moho de limo en el tiempo $t + 1$

Ecuación de difusión: describe la dispersión de la concentración de nutrientes en el entorno en función del tamaño y la posición de la masa de moho de limo.

$$\mathbf{v}_i^{t+1} = \beta \sum_{j=1}^m \frac{\mathbf{x}_j^t - \mathbf{x}_i^t}{\|\mathbf{x}_j^t - \mathbf{x}_i^t\|} \quad (2.26)$$

Donde:

β es un parámetro de control que ajusta la tasa de difusión \mathbf{x}_i^t es la posición de la masa de moho de limo en el tiempo t \mathbf{x}_j^t es la posición de la masa de moho de limo en el tiempo t m es el número total de masas de moho de limo $\|\mathbf{x}_j^t - \mathbf{x}_i^t\|$ es la norma euclidiana de la diferencia entre las posiciones de las masas de moho de limo.

Estas ecuaciones se utilizan en conjunto para modelar la dinámica del moho de limo y, por lo tanto, para resolver un problema de optimización dado.

2.2. Desarrollo de la Aplicación

La aplicación de algoritmos metaheurísticos inspirados en la naturaleza ha sido una tendencia cada vez más popular en el campo de la optimización de sistemas. En particular, estos algoritmos se han aplicado en la optimización de antenas patch rectangulares, un tipo de antena que se utiliza ampliamente en sistemas de comunicación inalámbrica.

Se han desarrollado varios algoritmos metaheurísticos para optimizar las antenas patch rectangulares, incluidos Algorithm, Firefly Algorithm, Harris Hawks Optimization, Marine Predators Algorithm, Particle Swarm Optimization y Slime Mould Algorithm. Cada uno de estos algoritmos tiene sus propias características y ventajas en términos de eficiencia y precisión en la optimización de antenas patch rectangulares.

Una vez que se han desarrollado estos algoritmos, se necesita una interfaz de

usuario amigable para permitir a los usuarios aplicarlos y realizar la optimización de las antenas patch rectangulares de manera fácil y rápida. En este caso, se optó por la interfaz de usuario proporcionada por App Designer en MATLAB.

App Designer es una herramienta de desarrollo de aplicaciones gráficas en MATLAB que permite a los usuarios crear interfaces de usuario personalizadas y funcionales sin necesidad de conocimientos previos de programación. Con App Designer, los usuarios pueden crear aplicaciones interactivas para la optimización de antenas patch rectangulares, lo que permite una mejor visualización y control de los procesos de optimización.

La interfaz de usuario de App Designer proporciona una serie de herramientas gráficas y de diseño que permiten al usuario crear una interfaz de usuario personalizada de manera fácil y rápida. El usuario puede importar datos de archivos de texto, Excel, mat, hdf5, imágenes y otros formatos de archivo comunes en MATLAB. Además, la herramienta de diseño visual de App Designer proporciona una vista previa en tiempo real de la aplicación, lo que permite al usuario ver cómo se verá la aplicación antes de que se ejecute.

En conclusión, la combinación de algoritmos metaheurísticos inspirados en la naturaleza y App Designer en MATLAB permite una optimización eficiente y precisa de las antenas patch rectangulares, con una interfaz de usuario amigable y fácil de usar. Con esta herramienta, los usuarios pueden optimizar sus antenas patch rectangulares sin la necesidad de tener conocimientos avanzados de programación, lo que aumenta la eficiencia y la productividad en el campo de la optimización de sistemas de comunicación inalámbrica.

2.2.1. App Designer en Matlab

App Designer es una herramienta de desarrollo de aplicaciones gráficas en MATLAB que permite crear interfaces de usuario personalizadas y funcionales sin necesidad de conocimientos previos de programación. Con App Designer, los usuarios pueden crear aplicaciones interactivas, desde aplicaciones de análisis de datos hasta aplicaciones de control de instrumentos y mucho más.

App Designer es una herramienta muy poderosa que permite al usuario

crear aplicaciones de manera fácil y rápida. La herramienta está diseñada para simplificar el proceso de diseño de interfaces de usuario al proporcionar una serie de herramientas gráficas y de diseño que permiten al usuario crear una interfaz de usuario personalizada de manera fácil y rápida.

App Designer está diseñado para trabajar con el entorno de programación de MATLAB. Esto significa que los usuarios pueden utilizar todas las funciones y capacidades de MATLAB en sus aplicaciones. También es posible importar datos de archivos de texto, Excel, mat, hdf5, imágenes y otros formatos de archivo comunes en MATLAB.

App Designer también proporciona una gran cantidad de componentes predefinidos que el usuario puede arrastrar y soltar en la interfaz de usuario. Estos componentes incluyen botones, cuadros de texto, casillas de verificación, deslizadores, menús desplegados y muchos otros elementos de la interfaz de usuario comunes.

El proceso de diseño de la aplicación se realiza utilizando la herramienta de diseño visual. Esta herramienta proporciona una vista previa en tiempo real de la aplicación, lo que permite al usuario ver cómo se verá la aplicación antes de que se ejecute. La herramienta de diseño visual también permite al usuario agregar, eliminar y modificar los componentes de la interfaz de usuario de manera fácil y rápida.

Una vez que se ha diseñado la aplicación, se puede guardar como un archivo .mlapp. Este archivo se puede distribuir y ejecutar en cualquier computadora que tenga MATLAB instalado. El usuario también puede generar un archivo ejecutable para que la aplicación se pueda ejecutar en computadoras que no tienen MATLAB instalado.

En resumen, App Designer es una herramienta de desarrollo de aplicaciones gráficas en MATLAB que permite a los usuarios crear interfaces de usuario personalizadas y funcionales sin conocimientos previos de programación. Con App Designer, los usuarios pueden crear aplicaciones interactivas, desde aplicaciones de análisis de datos hasta aplicaciones de control de instrumentos y mucho más. App Designer es una herramienta poderosa y fácil de usar que permite a los usuarios diseñar y distribuir aplicaciones de manera fácil y rápida.

Implementación del aplicativo en app designer en Matlab

Una vez desarrallados los seis algoritmos de optimizacion se los agrega al app designer de Matlab. Como se ve en Figura 2.8 donde se puede visualizar la primera pestaña denominada "FIRST PHASE". Aquí se ingresan las variables de entrada como son: la constante dieléctrica, la frecuencia, el valor de búsqueda, la impedancia, la altura del sustrato y el barrido de frecuencia; donde estan remarcados en recuadros azules. Una vez ingresados los parametros de entrada se da clic en "CALCULATE", donde se tendran las dimensines iniciales del patch como son W , L y Y_0 . Ademas, de forma gráfica se puede el S11 y la impedancia con las dimensiones iniciales sin optimizar.

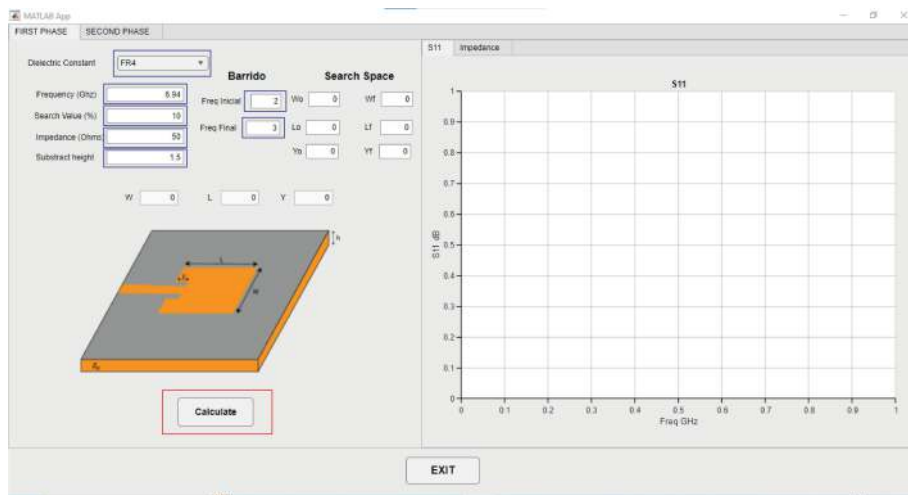


Figura 2.8: Función fitness de todos los algoritmos empleados.

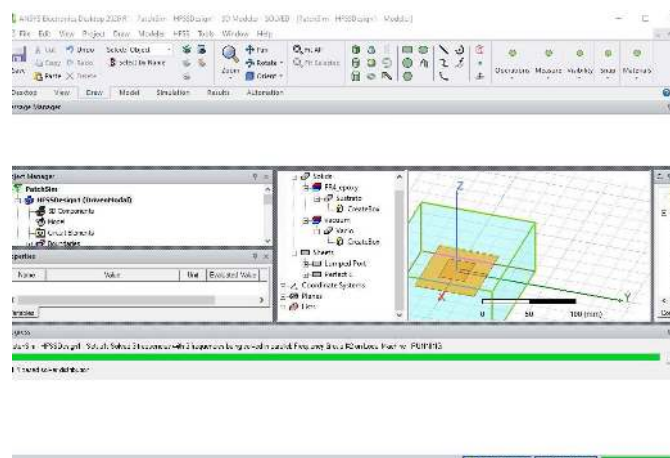
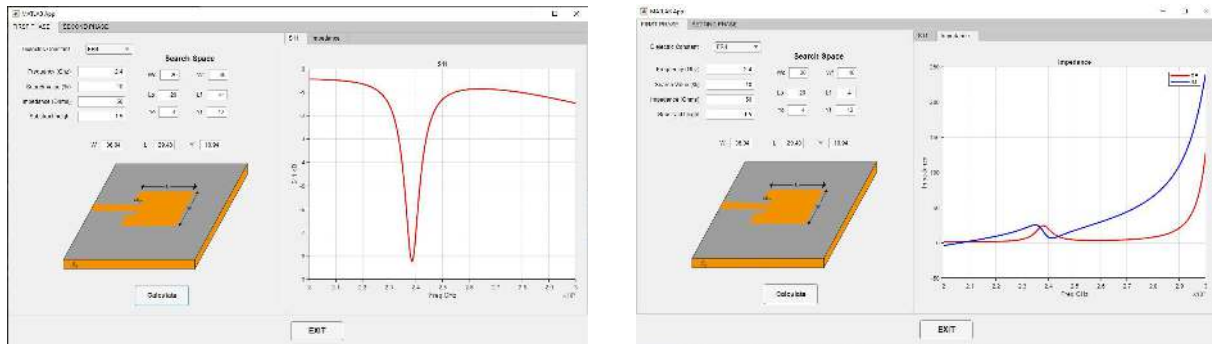


Figura 2.9: Simulación en HFSS en la First Phase.

En el proceso de calcular las dimensiones iniciales, se inicia la simulación en HFSS, Como se ve en la Figura 2.8. Una vez que se tiene los resultados de la simulación se muestran como son el S11 y la impedancia, de la atena patch rectangular Como se ve en la Figura 2.10.



(a) S11 (b) Impedancia
Figura 2.10: Resultados de la primera pestaña del aplicativo.

Una vez que se calculan los valores iniciales de la antena, se va a la siguiente pestaña denominada "SECOND PHASE", como se ve en la Figura 2.11. Donde se ingresan los parámetros como el algoritmo de optimización, puede escoger entre: BA, FA, HHO, MPA, PSO y SMA. Luego, se debe escoger el numero de iteraciones y agentes que va utilizar el algoritmo para la optimización de la antena patch rectangular. Después, se debe dar clic en CALCULATE. Finalmente, el algoritmo empezara a optimizar la antena para mostrar los resultados mas optimos de la misma.

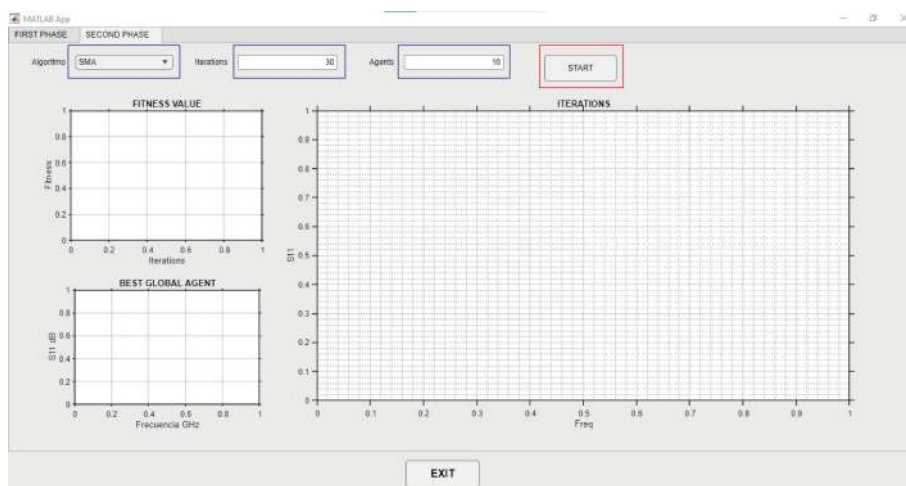


Figura 2.11: Función fitness de todos los algoritmos empleados.

Los algoritmos basados en la inteligencia de enjambre son una herramienta

poderosa en la optimización de problemas complejos. En este capítulo, se presentaron seis algoritmos basados en la inteligencia de enjambre que se implementaron para optimizar las dimensiones del patch de una antena patch rectangular. Estos algoritmos son el Bat Algorithm, el Firefly Algorithm, el Harris Hawks Optimization, el Marine Predators Algorithm, el Particle Swarm Optimization y el Slime Mould Algorithm. Cada uno de ellos tiene su propia metodología única y se han demostrado efectivos en una amplia variedad de problemas de optimización.

En la implementación de estos algoritmos, se utilizaron diferentes estrategias para ajustar los parámetros. Se estableció un tamaño de población fijo y se aplicaron diferentes técnicas de selección de padres y reemplazo de la población. Además, se utilizó una técnica de mutación para mejorar la exploración del espacio de búsqueda. En general, se encontró que estos algoritmos eran muy eficaces en la optimización de las dimensiones del patch de la antena, y ofrecían resultados comparables o mejores que otros enfoques de optimización.

El segundo punto de este capítulo se centra en el desarrollo de la aplicación. Se implementó una interfaz gráfica de usuario utilizando App Designer en Matlab. Esta aplicación permite al usuario final ejecutar cualquiera de los seis algoritmos mencionados anteriormente en la optimización de las dimensiones del patch de la antena rectangular. La interfaz gráfica de usuario es fácil de usar y permite al usuario ajustar los parámetros de cada algoritmo para obtener los resultados deseados.

En resumen, los algoritmos basados en la inteligencia de enjambre son una herramienta muy poderosa en la optimización de problemas complejos, y se han demostrado efectivos en la optimización de las dimensiones del patch de una antena patch rectangular. La aplicación desarrollada permite al usuario final ejecutar estos algoritmos de manera fácil y eficiente, lo que les permite obtener resultados óptimos en su aplicación.

Capítulo 3

Pruebas y análisis de resultados

El presente capítulo se centra en los resultados del aplicativo de las antenas tipo patch rectangulares y su optimización mediante el uso de seis algoritmos. Sin embargo, para lograr un rendimiento óptimo en términos de resonancia y ancho de banda, es necesario optimizar los parámetros de diseño de la antena. Esta optimización se puede lograr mediante el uso de algoritmos de optimización, que permiten encontrar los valores óptimos de los parámetros de diseño en un tiempo razonable.

Los algoritmos de optimización utilizados en este capítulo incluyen algoritmos: BAT, FA, HHO, MPA, PSO y SMA. Estos algoritmos se compararán y evaluarán en términos de su capacidad para encontrar soluciones óptimas y su eficiencia en términos de tiempo de cálculo.

Además, se realizarán comparaciones de los algoritmos del diseño de antenas para verificar la validez de los resultados obtenidos mediante los algoritmos de optimización.

Finalmente, se presentarán y discutirán los resultados obtenidos y se harán recomendaciones para futuros trabajos en este campo. Este capítulo proporcionará una visión general exhaustiva de la optimización de las antenas tipo patch mediante el uso de algoritmos y servirá como una guía valiosa para los investigadores y profesionales interesados en este campo.

3.1. Parámetros

En esta sección debe tomar en cuenta:

- Función fitness
- Criterio de paro
- S11
- Resonancia

Ya que estos parámetros, son datos que deben ingresar para poder utilizar la "First Phase" del aplicativo.

3.1.1. Función Fitness

La función fitness es una medida de la aptitud o desempeño de un individuo en un sistema de optimización por algoritmos genéticos o inteligencia artificial. En el contexto de las antenas patch, la función fitness puede ser utilizada para evaluar la eficacia de un diseño de antena en particular en términos de su capacidad para transmitir o recibir señales de radiofrecuencia.

Esta función es una medida de la calidad de la solución, y se utiliza para determinar qué individuos serán seleccionados para la próxima generación. Los individuos con una puntuación más alta en la función fitness tienen una mayor probabilidad de ser seleccionados para el cruzamiento y la mutación, mientras que los individuos con una puntuación más baja tienen una menor probabilidad.

La función fitness puede ser definida de diversas formas, dependiendo de los objetivos específicos de la optimización de la antena patch. Por ejemplo, puede basarse en la ganancia de la antena, el ancho de banda, el factor de forma, la directividad, entre otros parámetros. En general, se trata de una medida subjetiva que se define en función de los objetivos y requisitos específicos de cada aplicación.

3.1.2. Criterio de paro

El criterio de paro es una condición que determina cuándo un algoritmo debe detenerse. Es un aspecto crítico de la programación de un algoritmo, ya que permite controlar la eficiencia y la precisión del mismo.

Hay varios criterios de paro comunes en los algoritmos de optimización, que incluyen:

1. Número máximo de iteraciones: El algoritmo se detiene después de un número específico de iteraciones, independientemente de si ha alcanzado una solución óptima o no.
2. Tolerancia a la solución: El algoritmo se detiene cuando la solución se aproxima lo suficiente a una solución óptima, según una tolerancia predefinida.
3. Mejora mínima: El algoritmo se detiene cuando la mejora de la solución en cada iteración es menor que un valor mínimo establecido.
4. Tiempo límite: El algoritmo se detiene después de un período de tiempo determinado, independientemente de si ha alcanzado una solución óptima o no.

El criterio de paro adecuado depende del problema específico que se está abordando y de los requisitos de tiempo y precisión. Por lo general, es un equilibrio entre obtener una solución lo más precisa posible y minimizar el tiempo de ejecución del algoritmo.

3.1.3. S11

El parámetro S11 en una antena patch es una medida de la reflexión de la señal en la entrada de la antena. Se mide en términos de la relación de onda estacionaria esta se expresa en decibelios (dB).

El valor de S11 representa la fracción de la energía incidente en la entrada de la antena que se refleja y no se transmite hacia el espacio libre. Un valor bajo de S11 indica una alta eficiencia de transmisión y una baja pérdida de reflexión, mientras que un valor alto de S11 indica una baja eficiencia de transmisión y una alta pérdida de reflexión.

El valor óptimo de S_{11} para una antena patch depende de la aplicación específica y de los requisitos de rendimiento. Por lo general, se busca un valor lo más cercano a cero posible, lo que indica una baja pérdida de reflexión y una alta eficiencia de transmisión.

El parámetro S_{11} es una medida importante para el diseño y la optimización de las antenas patch, y es un indicador clave de la calidad de la antena. Se utiliza para evaluar la eficiencia de la transmisión de la señal, la selectividad de frecuencia y la directividad de la antena.

3.1.4. Resonancia

La resonancia es un concepto clave en el diseño y la optimización de las antenas patch. La resonancia se refiere a la frecuencia a la cual una antena se comporta de manera más eficiente y produce una radiación electromagnética máxima.

En una antena patch, la resonancia se produce cuando la impedancia de la entrada de la antena es igual a la impedancia característica del medio libre, lo que permite que la señal incidente se transmita de manera eficiente hacia el espacio libre. La frecuencia de resonancia se puede ajustar mediante la variación de las dimensiones físicas de la antena, como su tamaño y forma, o mediante la modificación de la impedancia eléctrica y magnética de la entrada de la antena.

El funcionamiento de la resonancia en una antena patch es fundamental para su eficiencia de transmisión y para la calidad de la señal transmitida. Al ajustar la frecuencia de resonancia a la frecuencia deseada, se puede mejorar la eficiencia de transmisión y la selectividad de frecuencia de la antena, lo que puede ser crítico en aplicaciones como la comunicación inalámbrica y la detección de radares.

3.2. Aplicativo

En la Figura 3.1 se muestra como se debe interactuar con la aplicación, las variables de entrada y así mismo las variables de salida que se obtendrán.

1. Definir los parámetros de entrada, como las dimensiones de la antena, los

requisitos de rendimiento, la frecuencia de operación, etc.

2. Inicializar los valores iniciales de las variables de diseño, como la forma y las dimensiones de la superficie de la antena.
3. Aplicar el algoritmo metaheurístico seleccionado para optimizar las variables de diseño.
4. Evaluar el rendimiento de la antena con los valores óptimos de las variables de diseño.
5. Comparar el rendimiento con los requisitos de diseño y determinar si se deben realizar ajustes adicionales.
6. Repetir los pasos 3-5 hasta alcanzar un rendimiento satisfactorio.
7. Presentar los resultados finales, incluyendo las variables de diseño óptimas y el rendimiento de la antena.

3.3. Pruebas

Las dimensiones de una antena patch (como la longitud y el ancho de la superficie de la antena) son importantes porque afectan la impedancia y la radiación electromagnética de la antena. Si las dimensiones no son adecuadas, la impedancia puede ser incorrecta y la radiación puede ser ineficiente, lo que resulta en un rendimiento deficiente de la antena.

Al optimizar las dimensiones de una antena patch, se pueden mejorar sus resultados al ajustar la impedancia y la radiación electromagnética para cumplir con los requisitos de diseño. Esto se puede lograr ajustando la forma y las dimensiones de la superficie de la antena para mejorar la distribución de la corriente en la superficie y, por lo tanto, mejorar la radiación electromagnética.

Además, la optimización de las dimensiones de la antena también puede mejorar otros parámetros de rendimiento, como la ganancia, la direccionalidad, el ancho de banda, etc. Estos parámetros son importantes para garantizar que la antena

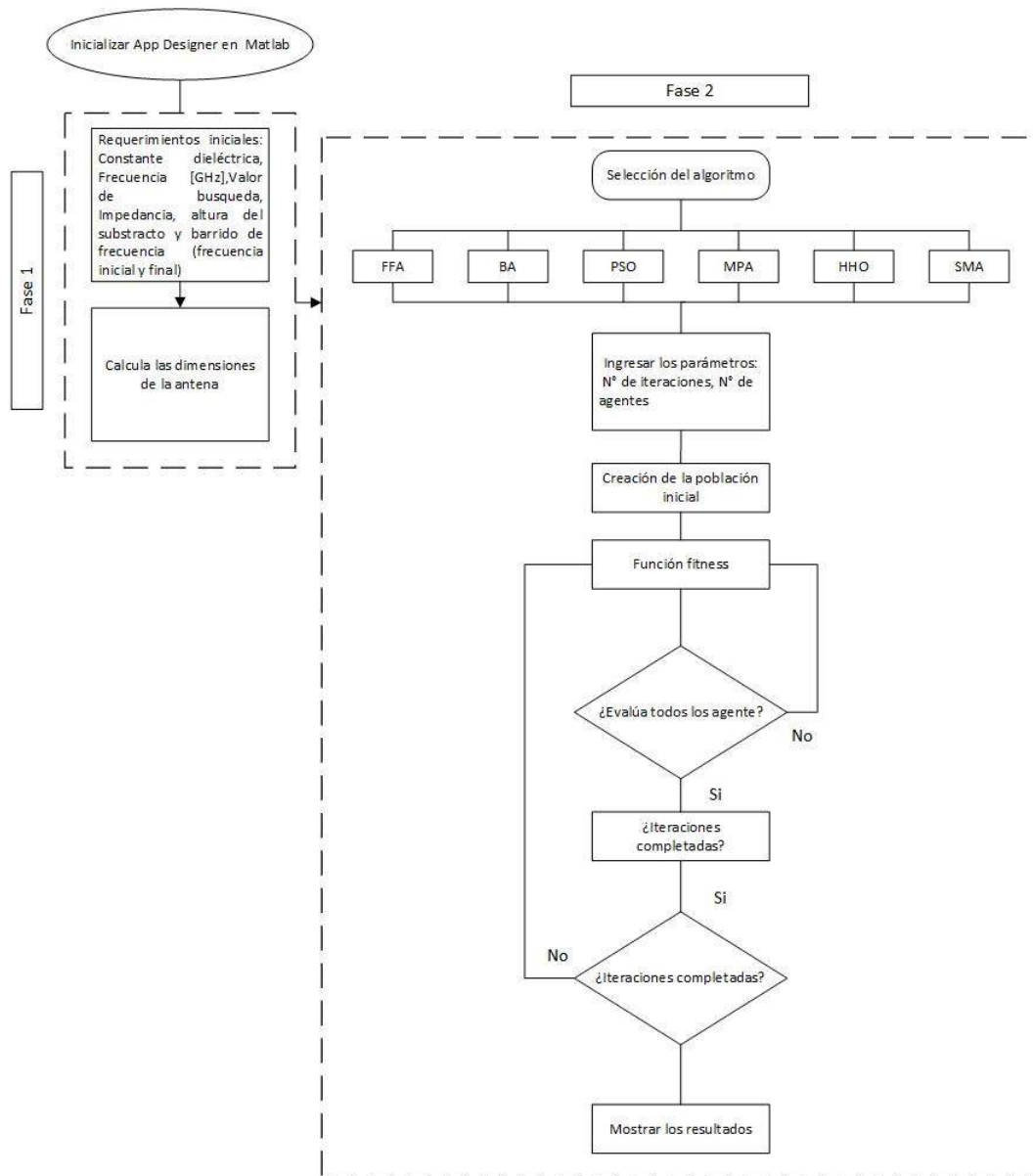


Figura 3.1: Diagrama de flujo.

cumpla con los requisitos de aplicación específicos, como la transmisión de señales de larga distancia o la detección de señales débiles.

Se utilizó una antena inicial, como se ve en la Figura 3.2 la cual tiene las dimensiones de W, L, Y . La optimización de su forma y dimensiones es importante para mejorar su rendimiento, incluyendo la ganancia, el ancho de banda, la directividad y la polarización.

Se comparó el rendimiento de la optimización de la antena patch mediante BA, FFA, HHO, MPA, PSO, SMA.

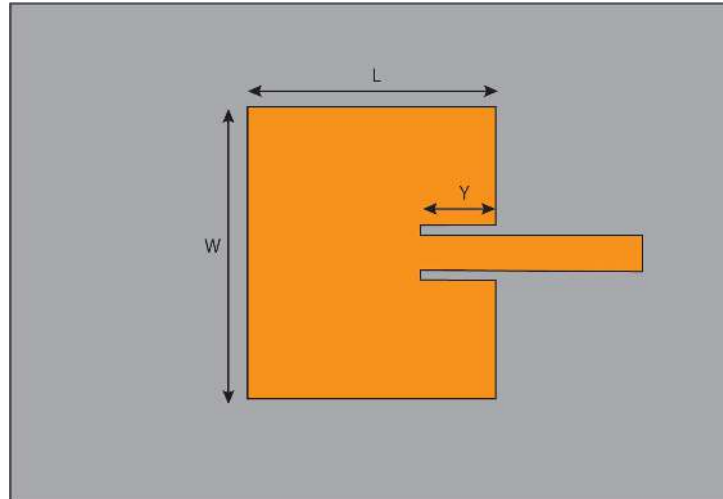


Figura 3.2: Antena patch rectangular.

En general, los resultados de estos estudios y experimentos demuestran la eficacia de los algoritmos meta-heurísticos en la optimización de las antenas patch. Estos algoritmos son útiles herramientas para mejorar el rendimiento de las antenas por lo tanto, para mejorar la eficiencia en muchas aplicaciones importantes.

Para cada algoritmo se empleo 10 agentes y 30 iteraciones teniendo como resultados los siguientes:

3.3.1. Bat Algorithm

Se han realizado varias optimizaciones para las frecuencias de 2.4GHz, consiguiendo que la antena funcionara a la frecuencia especificada y con mejores valores de S_{11} que el deseado.

A continuación se presentan los resultados obtenidos empleando el algoritmo metaheurístico inspirado en la naturaleza BAT. En la Figura 3.3 se presenta los 30 resultados que se obtuvieron, donde en cada iteración se observa los cambios de la optimización de la función fitness, se puede visualizar como el algoritmos se va adaptando hasta encontrar la mejor solución mas óptimos.

En la Figura 3.4 se presenta el mejor parámetro de reflexión de la antena, teniendo como el mejor resultado global una resonancia en el S_{11} es de $-41,3033dB$ en la frecuencia de 2.40301 Ghz, el resultados es mas óptimo que encontró el algoritmo.

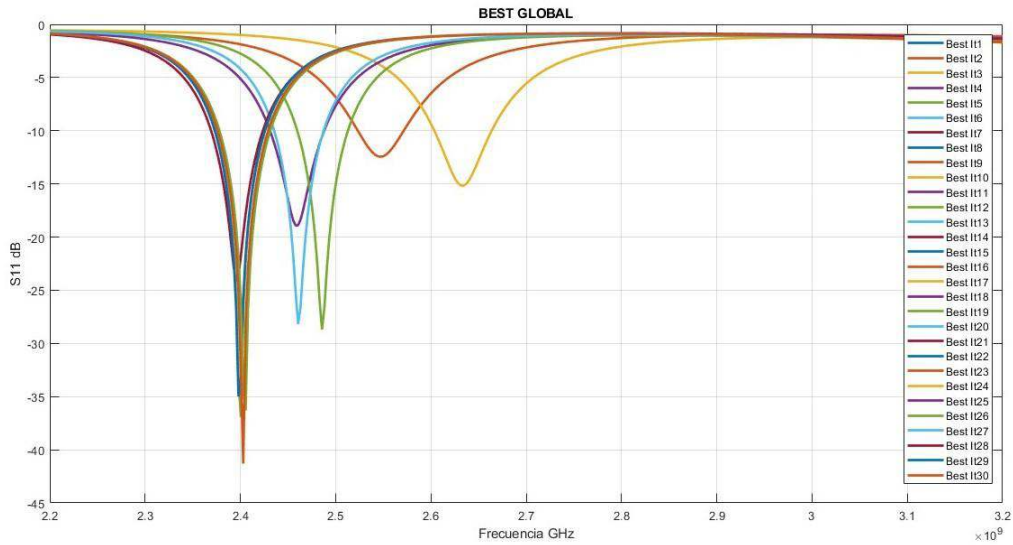


Figura 3.3: Algoritmo BAT, parámetros de reflexión de la antena.

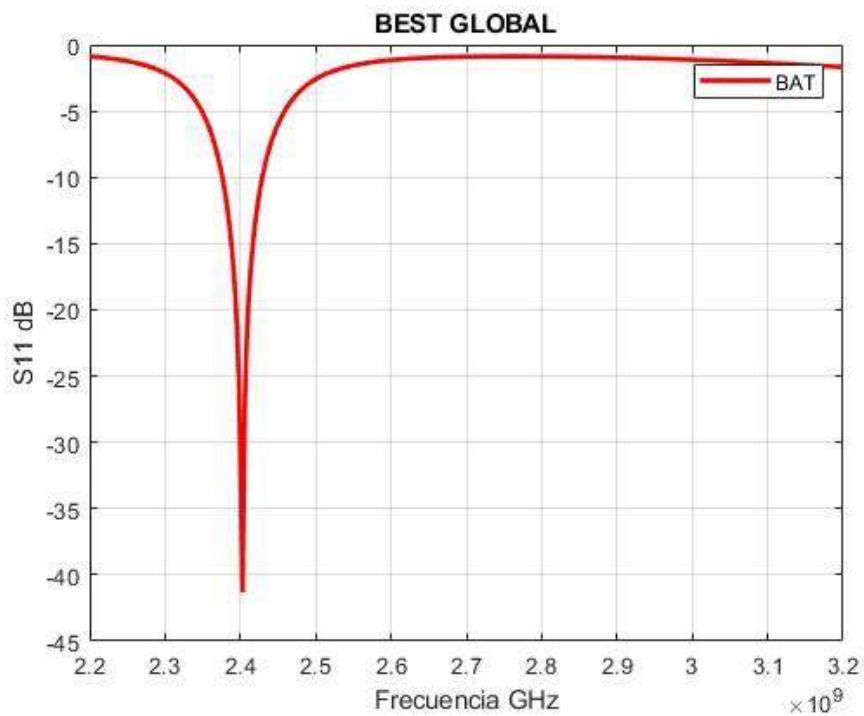


Figura 3.4: Mejor parámetro de reflexión de la antena usando el algoritmo BAT.

En la Figura 3.5 se presenta gráficamente la función fitness, donde se puede observar como converge el algoritmo y va encontrando en cada iteración un función mas óptima o en algunos casos de mantiene.

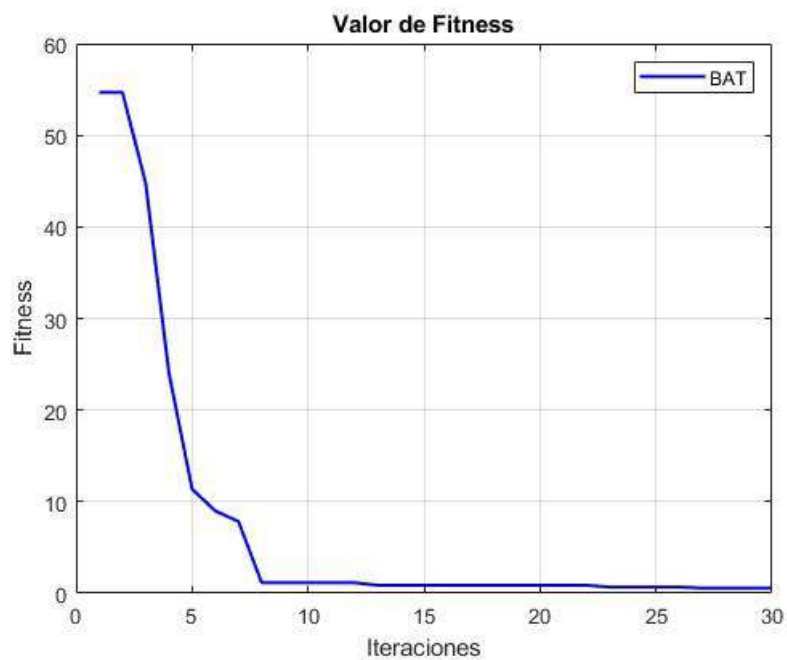


Figura 3.5: Algoritmo BAT, Función fitness obtenida en 30 iteraciones.

Además, en la Tabla 3.1 se observa, que el algoritmo empieza con la iteración 1, que tiene una función de fitness de 54.6779. Finalmente, converge en la iteración 27 que encontró como resultado un error de 0.5621 de la función fitness y se mantiene hasta la iteración 30.

Tabla 3.1: Resultados obtenidos del algoritmo BAT.

BAT	
Iter.	Fitness
1-2	54,6779
3	44,7785
4	24,1205
5	11,3835
6	8,99355
7-12	7,83708
13-23	0,873991
24-26	0,684878
27-30	0,562172

3.3.2. Firefly Algorithm

Se han realizado varias optimizaciones para las frecuencias de 2.4GHz, en este caso usando FA, se consiguió que la antena funcionara a la frecuencia especificada y con mejores valores de S11 que el deseado.

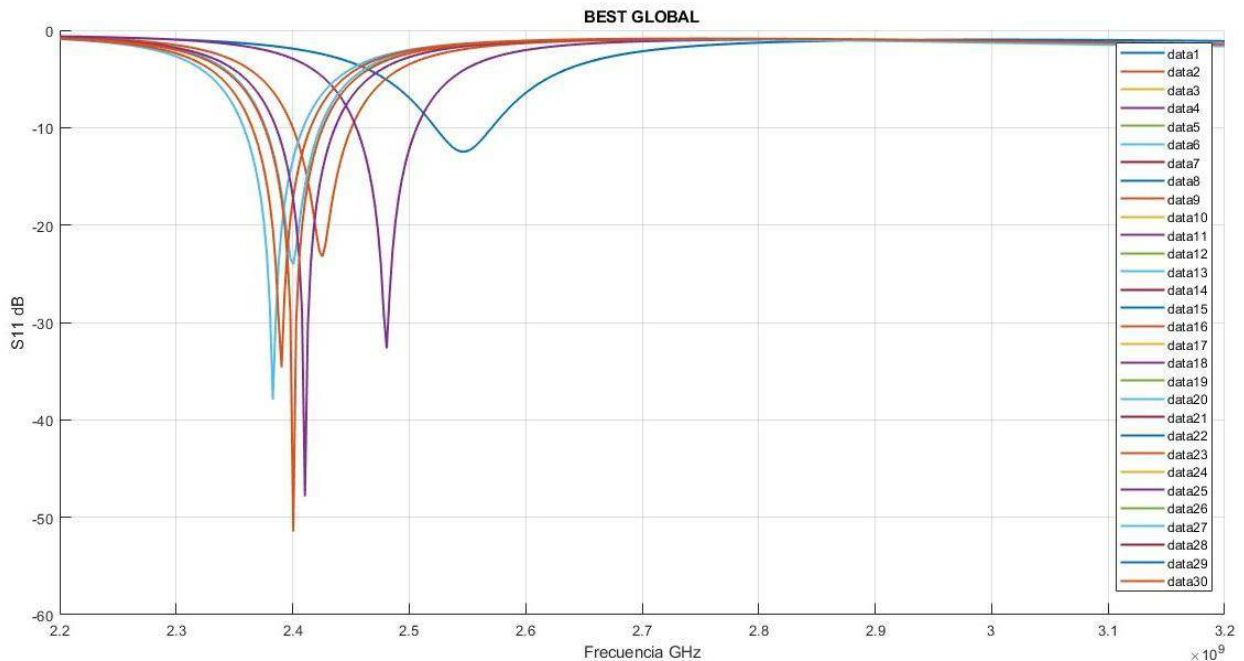


Figura 3.6: Parámetro de reflexión de la antena usando el algoritmo Firefly.

A continuación se presentan los resultados obtenidos empleando el algoritmo metaheurístico inspirado en la naturaleza FA. En la Figura 3.6 se presenta los 30 resultados que se obtuvieron, donde en cada iteración se observa los cambios de la optimización de la función fitness, se puede visualizar como el algoritmos se va adaptando hasta encontrar la mejor solución mas óptimos.

En la Figura 3.7 se presenta el mejor parámetro de reflexión de la antena, teniendo como el mejor resultado global una resonancia en el S11 es de -51.4699 [dB] en una frecuencia de 2.4[GHz] el resultados es mas óptimo que encontró el algoritmo.

En la Figura 3.8 se presenta gráficamente la función fitness, donde se puede observar como converge el algoritmo y va encontrando en cada iteración un función mas óptima.

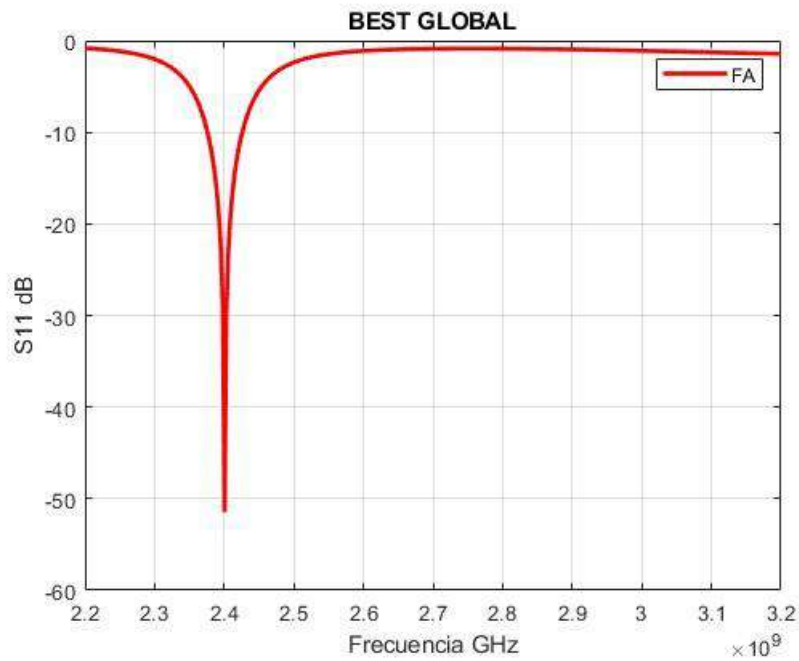


Figura 3.7: Mejor parámetro de reflexión de la antena usando el algoritmo Firefly.

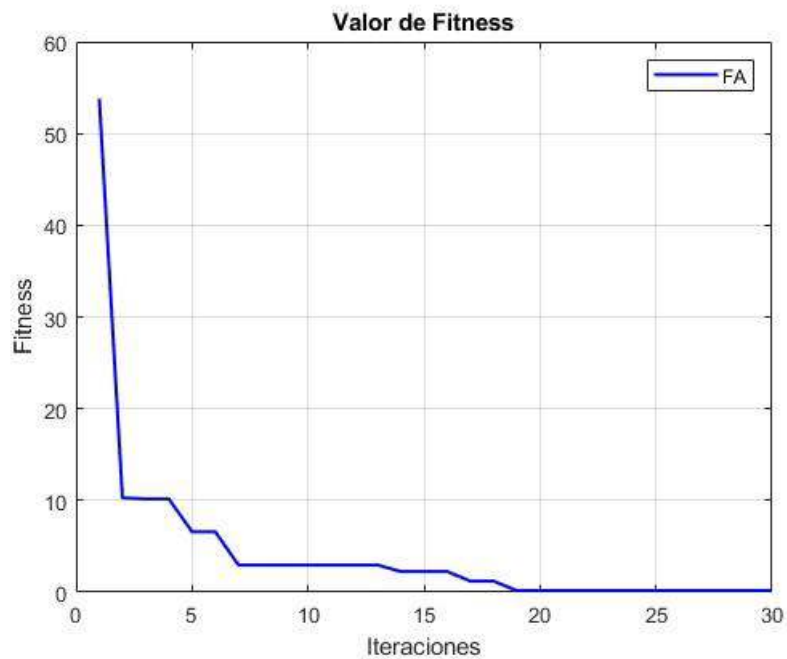


Figura 3.8: Mejor parámetro de reflexión de la antena usando el algoritmo FA.

Además, en la Tabla 3.2 se observa, que el algoritmo empieza con la iteración 1, que tiene una función de fitness de 53.8009. Finalmente, converge en la iteración 19 que encontró como resultado un error de 0.163755 de la función fitness y se mantiene hasta la iteración 30.

Tabla 3.2: Resultados obtenidos del algoritmo FA.

FA	
Iter.	Fitness
1	53,8009
2-4	10,2826
5-6	6,58574
7-16	2,96204
17-18	1,1998
19-30	0,163755

3.3.3. Harris Hawks Optimization

Se han realizado varias optimizaciones para las frecuencias de 2,4GHz, en este caso usando HHO, se consiguió que la antena funcionara a la frecuencia especificada y con mejores valores de S11 que el deseado.

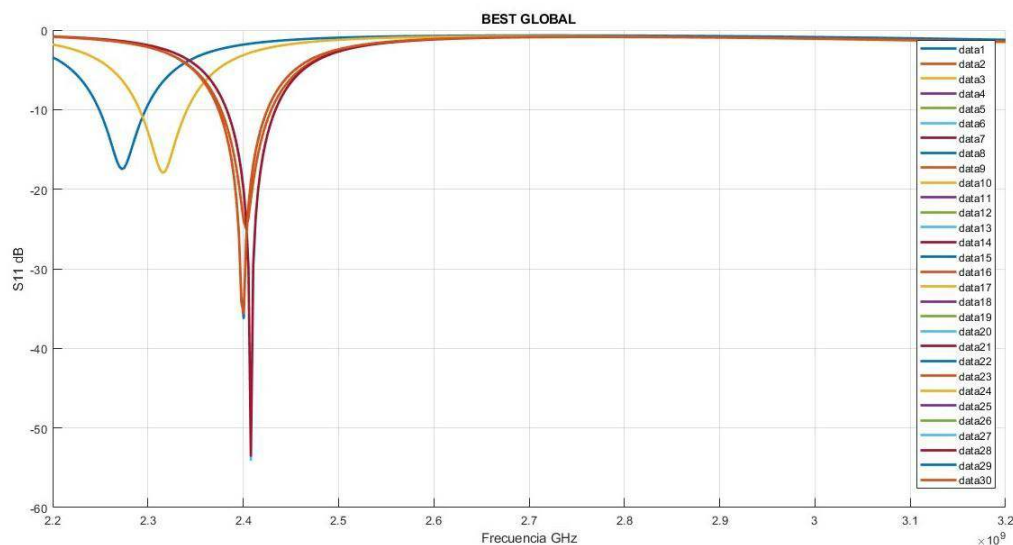


Figura 3.9: Parámetro de reflexión de la antena usando el algoritmo HHO.

A continuación se presentan los resultados obtenidos empleando el algoritmo metaheurístico inspirado en la naturaleza HHO. En la Figura 3.9 se presenta los 30 resultados que se obtuvieron, donde en cada iteración se observa los cambios de la optimización de la función fitness, se puede visualizar como el algoritmos se va adaptando hasta encontrar la mejor solución mas óptimos. En la Figura 3.10

se presenta el mejor parámetro de reflexión de la antena, teniendo como el mejor resultado global una resonancia en el S11 es de $-36,2926 \text{ dB}$ en una frecuencia de $2,4 \text{ GHz}$ el resultados es mas óptimo que encontró el algoritmo.

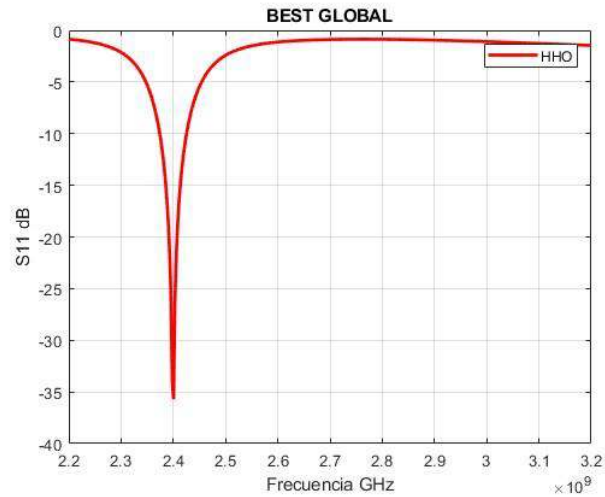


Figura 3.10: Mejor parámetro de reflexión de la antena usando el algoritmo HHO.

En la Figura 3.11 se presenta gráficamente la función fitness, donde se puede observar como converge el algoritmo y va encontrando en cada iteración un función mas óptima. Además, en la Tabla 3.3 se observa, que el algoritmo empieza con la iteración 1, que tiene una función de fitness de 35,6895. Finalmente, converge en la iteración 30 que encontró como resultado un error de 0.106555 de la función fitness.

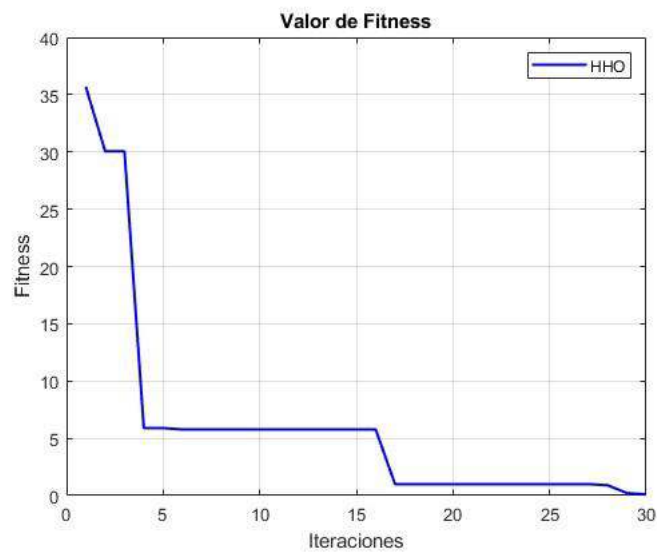


Figura 3.11: Función fitness del algoritmo HHO.

En la tabla 3.3 se puede apreciar como en la iteración 17 converge y no es hasta la iteración 28 en la que baja su valor y en la iteración 30 alcanzando 0,10655

Tabla 3.3: Resultados obtenidos del algoritmo HHO.

HHO	
Iter.	Fitness
1	35,6895
2-3	30,0539
4	5,89274
5	5,89274
6-16	5,76772
17-27	0,996425
28	0,901985
29	0,212375
30	0,106555

3.3.4. Marine Predators Algorithm

Se han realizado varias optimizaciones para las frecuencias de $2,4\text{GHz}$, en este caso usando MPA, se consiguió que la antena funcionara a la frecuencia especificada y con mejores valores de S_{11} que el deseado.

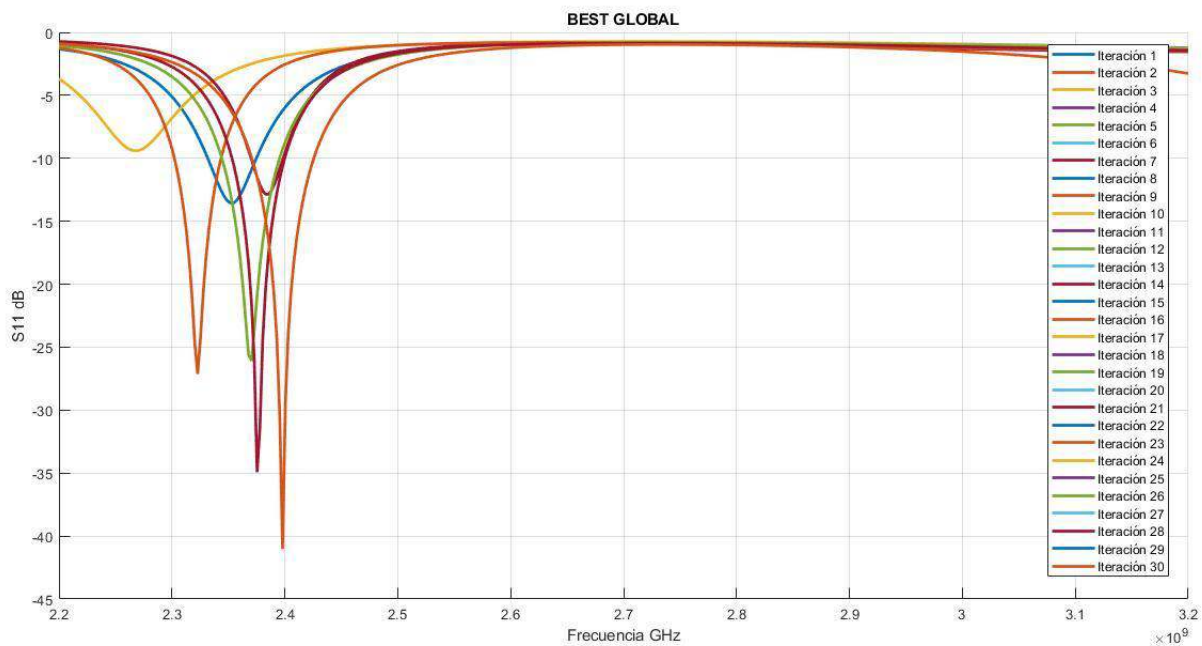


Figura 3.12: Parámetro de reflexión de la antena usando el algoritmo MPA.

A continuación se presentan los resultados obtenidos empleando el algoritmo metaheurístico inspirado en la naturaleza MPA. En la Figura 3.12 se presenta los 30 resultados que se obtuvieron, donde en cada iteración se observa los cambios de la optimización de la función fitness, se puede visualizar como el algoritmo se va adaptando hasta encontrar la mejor solución más óptima.

En la Figura 3.13 se presenta el mejor parámetro de reflexión de la antena, teniendo como el mejor resultado global una resonancia en el S_{11} es de $-41,025\text{ dB}$ en una frecuencia de $2,4\text{GHz}$ el resultado es más óptimo que encontró el algoritmo.

En la Figura 3.14 se presenta gráficamente la función fitness, donde se puede observar como converge el algoritmo y va encontrando en cada iteración un función más óptima.

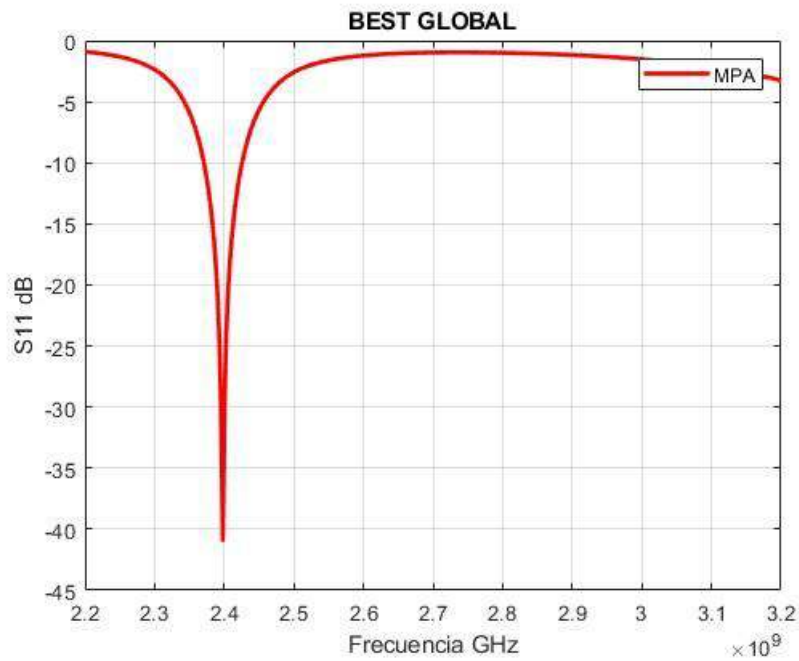


Figura 3.13: Mejor parámetro de reflexión de la antena usando el algoritmo MPA.

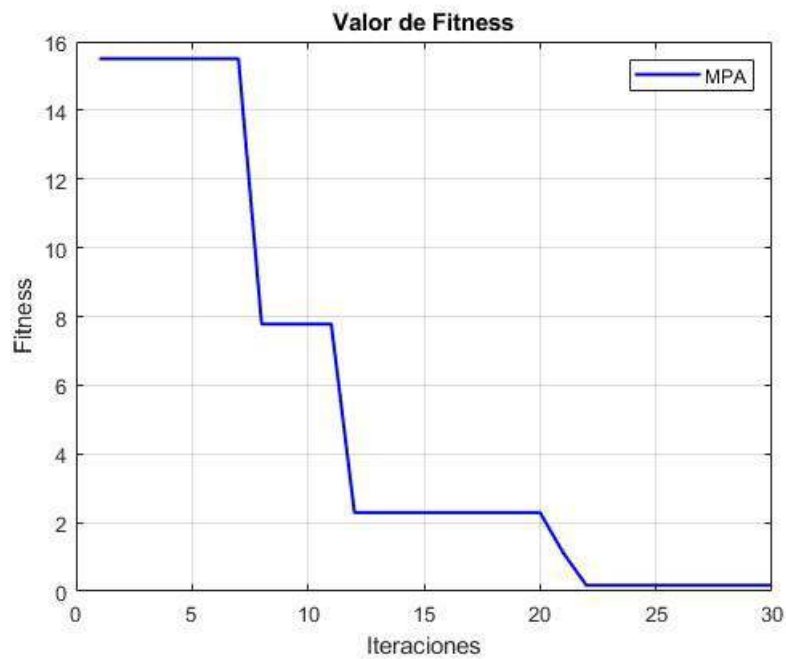


Figura 3.14: Función fitness del algoritmo MPA.

Además, en la Tabla 3.4 se observa, que el algoritmo empieza con la iteración 1, que tiene una función de fitness de 15,5051; MPA es la que comienza con un menor error en la función fitness. Finalmente, converge en la iteración 22 que encontró como resultado un error de 0,174525 de la función fitness.

Tabla 3.4: Resultados obtenidos del algoritmo MPA.

MPA	
Iter.	Fitness
1-7	15,5051
8-11	7,78102
12-20	2,29399
21	1,12012
22-30	0,174525

3.3.5. Particle Swarm Optimization

Se han realizado varias optimizaciones para las frecuencias de 2,4GHz, en este caso usando PSO, se consiguió que la antena funcionara a la frecuencia especificada y con mejores valores de S11 que el deseado.

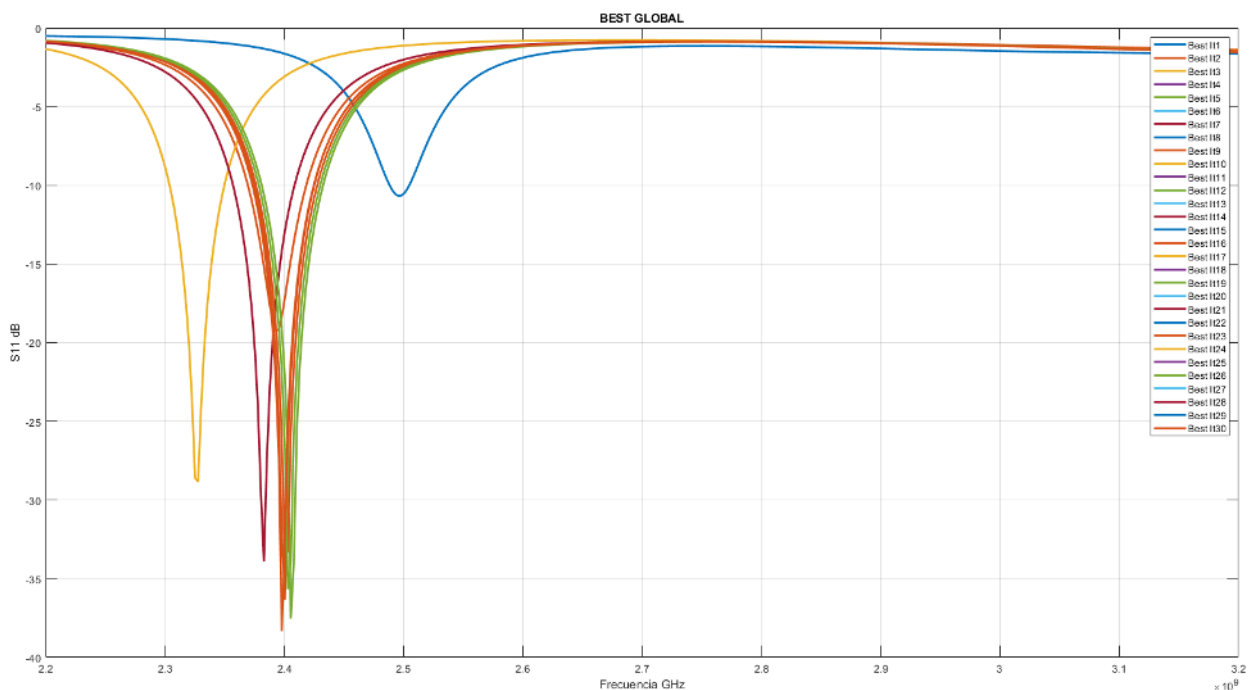


Figura 3.15: Parámetro de reflexión de la antena usando el algoritmo PSO.

A continuación, se presentan los resultados obtenidos empleando el algoritmo metaheurístico inspirado en la naturaleza PSO. En la Figura 3.15 se presenta los 30 resultados que se obtuvieron, donde en cada iteración se observa los cambios de la optimización de la función fitness, se puede visualizar como el algoritmos se va

Tabla 3.5: Resultados obtenidos del algoritmo PSO.

PSO	
Iter.	Fitness
1	40,7822
2	17,6842
3-7	10,6674
8-12	1,42692
13-23	0,719935
24-26	0,508392
27-30	0,405291
28	0,405291
29	0,405291
30	0,405291

adaptando hasta encontrar la mejor solución mas óptimos.

En la Figura 3.16 se presenta el mejor parámetro de reflexión de la antena, teniendo como el mejor resultado global una resonancia en el S11 es de $-38,3364 \text{ dB}$ en una frecuencia de $2,4 \text{ GHz}$ el resultados es mas óptimo que encontró el algoritmo.

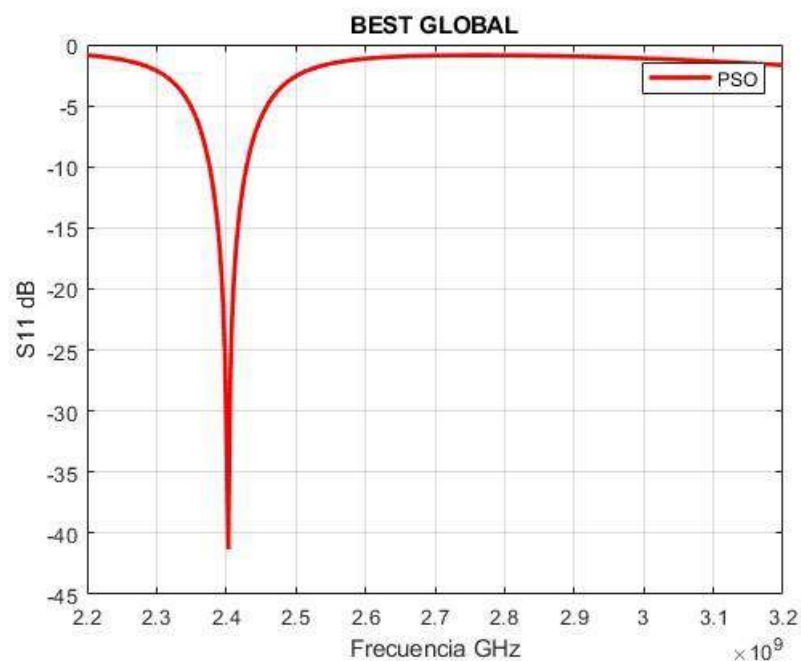


Figura 3.16: Mejor parámetro de reflexión de la antena usando el algoritmo PSO.

En la Figura 3.17 se presenta gráficamente la función fitness, donde se puede observar como converge el algoritmo y va encontrando en cada iteración un función mas óptima.

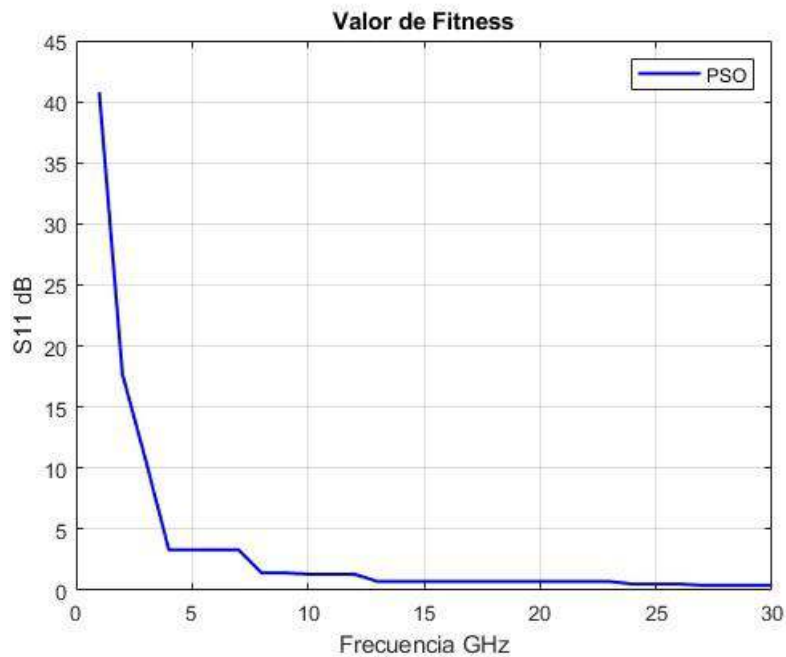


Figura 3.17: Función fitness del algoritmo PSO.

Además, en la Tabla 3.5 se observa, que el algoritmo empieza con la iteración 1, que tiene una función de fitness de 15,5051. Finalmente, converge en la iteración 22 que encontró como resultado un error de 0,174525 de la función fitness.

3.3.6. Slime Mould Algorithm

Se han realizado varias optimizaciones para las frecuencias de 2,4GHz, en este caso usando SMA, se consiguió que la antena funcionara a la frecuencia especificada y con mejores valores de S11 que el deseado.

A continuación se presentan los resultados obtenidos empleando el algoritmo metaheurístico inspirado en la naturaleza PSO. En la Figura 3.18 se presenta los 30 resultados que se obtuvieron, donde en cada iteración se observa los cambios de la optimización de la función fitness, se puede visualizar como el algoritmo se va adaptando hasta encontrar la mejor solución más óptima.

En la Figura 3.19 se presenta el mejor parámetro de reflexión de la antena, teniendo como el mejor resultado global una resonancia en el S11 es de -38,3364 dB en una frecuencia de 2,4GHz el resultado es más óptimo que encontró el algoritmo.

En la Figura 3.20 se presenta gráficamente la función fitness, donde se puede

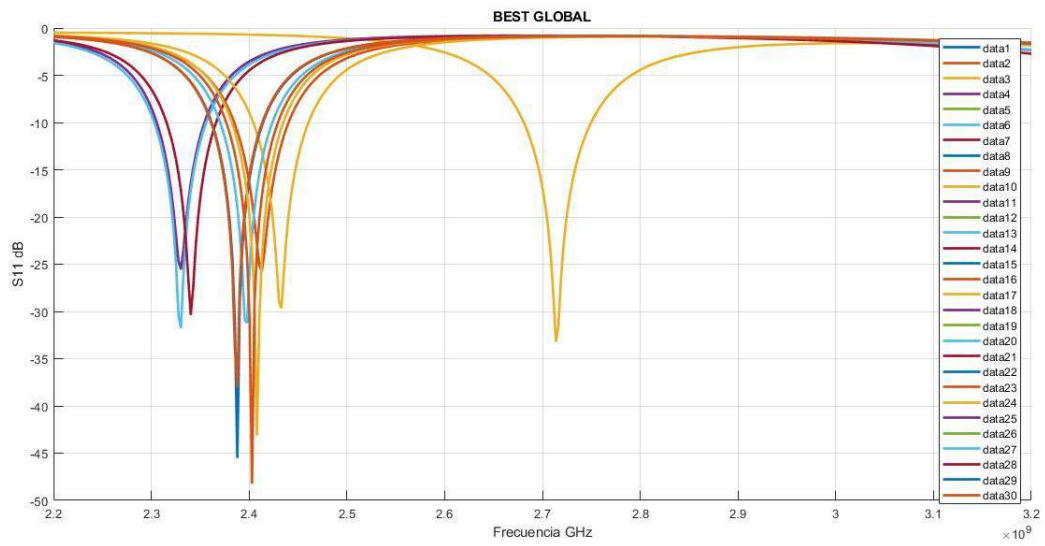


Figura 3.18: Parámetro de reflexión de la antena usando el algoritmo SMA.

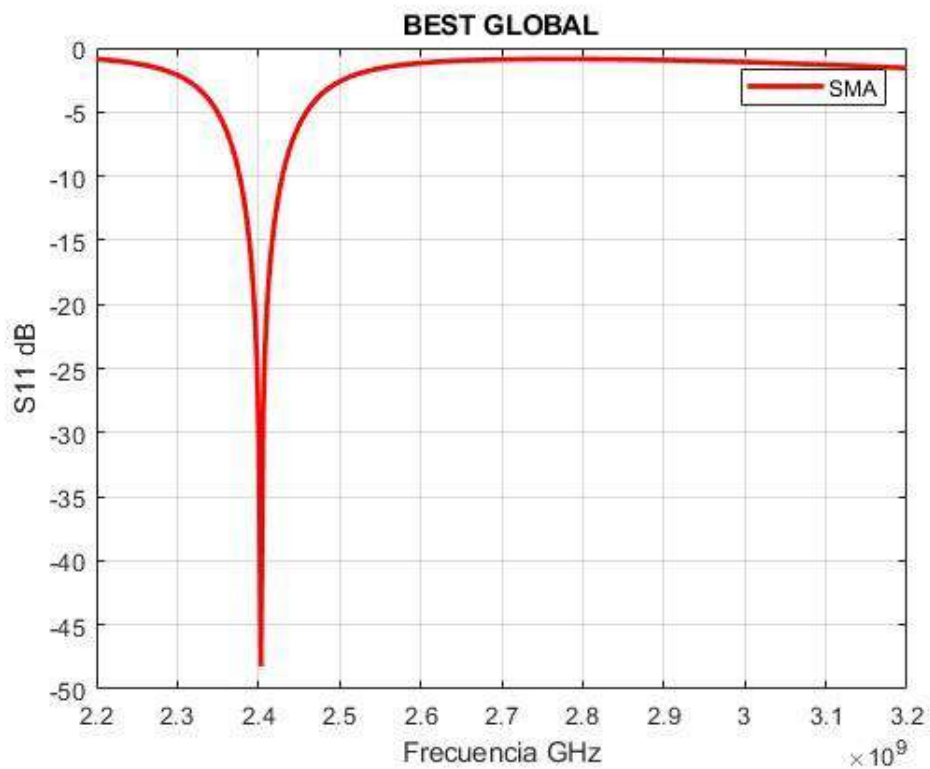


Figura 3.19: Mejor parámetro de reflexión de la antena usando el algoritmo SMA.

observar como converge el algoritmo y va encontrando en cada iteración un función mas óptima.

Además, en la Tabla 3.6 se observa, que el algoritmo empieza con la iteracion 1, que tiene una función de fitness de -32,9395. Finalmente, converge en la iteración 25

Tabla 3.6: Resultados obtenidos del algoritmo SMA.

SMA	
Iter.	Fitness
1-3	32,9395
4	11,9701
5-6	9,11556
7	8,79842
8-9	6,03827
10	5,81104
11-22	1,88188
23	1,35708
24	1,24112
25-30	0,409522

que encontró como resultado un error de 0,409522 de la función fitness.

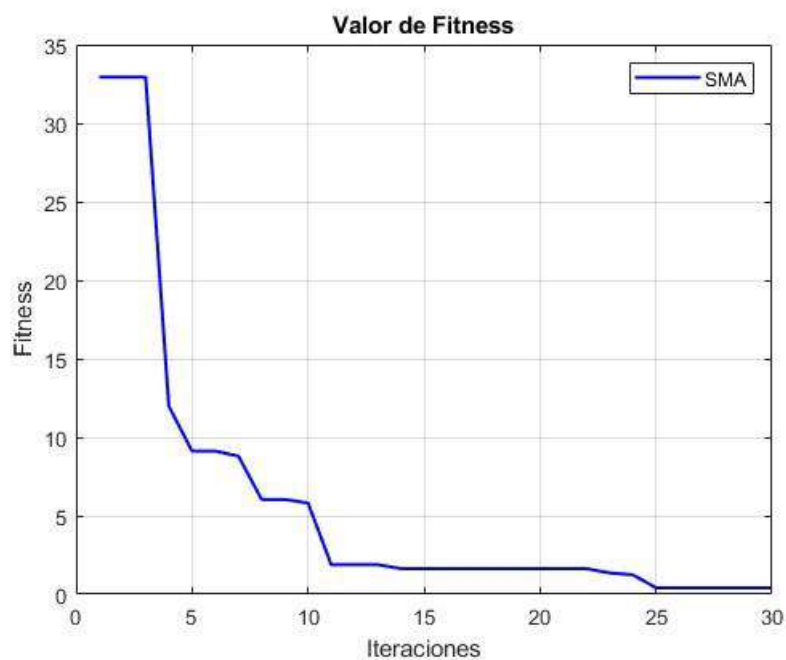


Figura 3.20: Función fitness SMA.

3.4. Análisis y comparación de los resultados obtenidos.

En esta sección, en base a los mejores resultados obtenidos de los seis algoritmos anteriores se realiza un análisis y comparación de datos para ver cual es el mejor; en base a su convergencia, resonancia, medidas del patch y tiempo de ejecución.

Tabla 3.7: Resultados obtenidos de los algoritmos.

Iter.	Fitness BAT	Fitness FA	Fitness HHO	Firness MPA	Fitness PSO	Fitness SMA
1	54,6779	53,8009	35,6895	15,5051	40,7822	32,9395
2	54,6779	10,2826	30,0539	15,5051	17,6842	32,9395
3	44,7785	10,1629	30,0539	15,5051	10,6674	32,9395
4	24,1205	10,1629	5,89274	15,5051	3,31777	11,9701
5	11,3835	6,58574	5,89274	15,5051	3,31777	9,11556
6	8,99355	6,58574	5,76772	15,5051	3,31777	9,11556
7	7,83708	2,96204	5,76772	15,5051	3,31777	8,79842
8	1,15619	2,96204	5,76772	7,78102	1,42692	6,03827
9	1,15619	2,96204	5,76772	7,78102	1,42692	6,03827
10	1,15619	2,96204	5,76772	7,78102	1,3146	5,81104
11	1,15619	2,96204	5,76772	7,78102	1,3146	1,88188
12	1,15619	2,96204	5,76772	2,29399	1,3146	1,88188
13	0,873991	2,96204	5,76772	2,29399	0,719935	1,88188
14	0,873991	2,24651	5,76772	2,29399	0,719935	1,63242
15	0,873991	2,24651	5,76772	2,29399	0,719935	1,63242
16	0,873991	2,24651	5,76772	2,29399	0,719935	1,63242
17	0,873991	1,1998	0,996425	2,29399	0,719935	1,63242
18	0,873991	1,1998	0,996425	2,29399	0,719935	1,63242
19	0,873991	0,163755	0,996425	2,29399	0,719935	1,63242
20	0,873991	0,163755	0,996425	2,29399	0,719935	1,63242
21	0,873991	0,163755	0,996425	1,12012	0,719935	1,63242
22	0,873991	0,163755	0,996425	0,174525	0,719935	1,63242
23	0,873991	0,163755	0,996425	0,174525	0,719935	1,35708
24	0,684878	0,163755	0,996425	0,174525	0,508392	1,24112
25	0,684878	0,163755	0,996425	0,174525	0,508392	0,409522
26	0,684878	0,163755	0,996425	0,174525	0,508392	0,409522
27	0,562172	0,163755	0,996425	0,174525	0,405291	0,409522
28	0,562172	0,163755	0,901985	0,174525	0,405291	0,409522
29	0,562172	0,163755	0,212375	0,174525	0,405291	0,409522
30	0,562172	0,163755	0,106555	0,174525	0,405291	0,409522

En la Tabla 3.7, se pueden apreciar el valor de la función fitness de forma global las 30 iteraciones de cada uno de algoritmos y como cada uno de ellos va convergiendo. De manera visual se puede apreciar en la Figura 3.21, donde a simple vista el que converge mas rápido en menos iteraciones es el PSO. Sin embargo, el que empieza con un menor error en la función fitness es el MPA con un error de 15.5051. Pero el que tiene una mejor convergencia y menor error en la función fitness es el algoritmo HHO con una función fitness de 0.10655, en el cual podemos apreciar que mantiene en su inicio en la iteración 4 a la iteración 16 valores similares del fitness; y en la iteracion 17 a la 28 mantiene otro valor de fitness de 0.901. Finalmente, en las dos ultimas iteraciones vemos un valor de fitness de 0.106, un valor muy cercano a comparación de los demás algoritmos de optimización. Además, anteriormente en cada algoritmo se explica de forma mas detallada los resultados que se obtuvieron de cada uno de ellos.

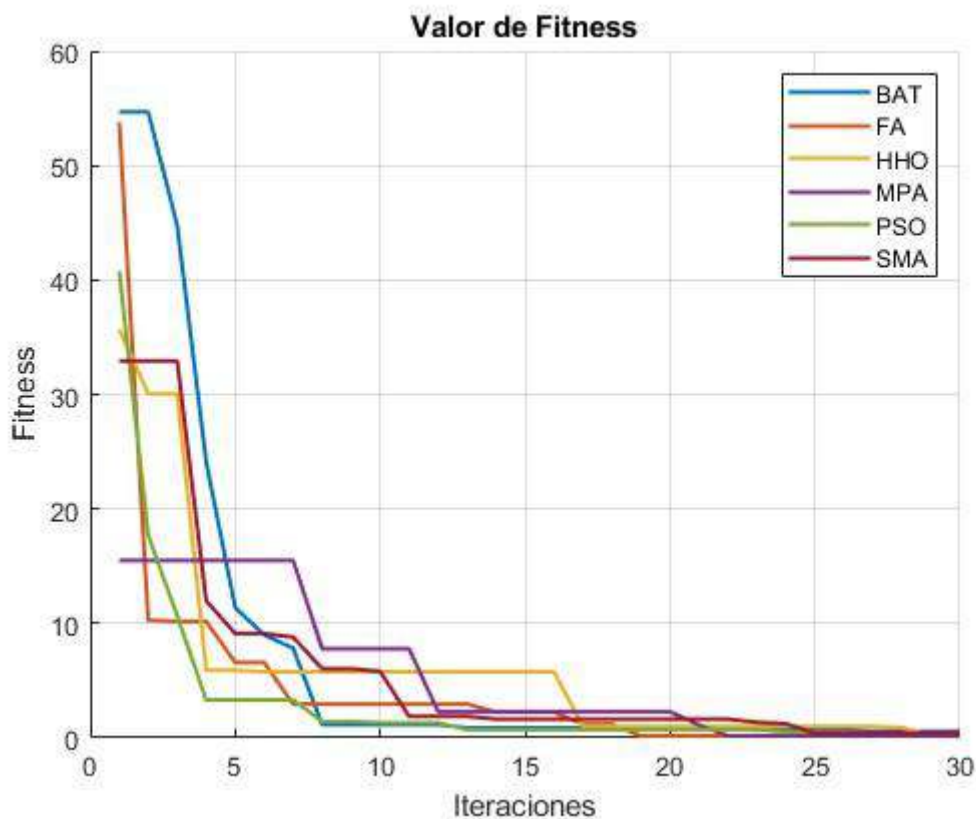
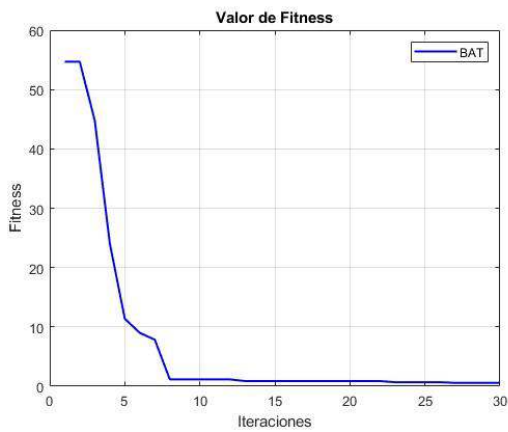


Figura 3.21: Función fitness de todos los algoritmos empleados.

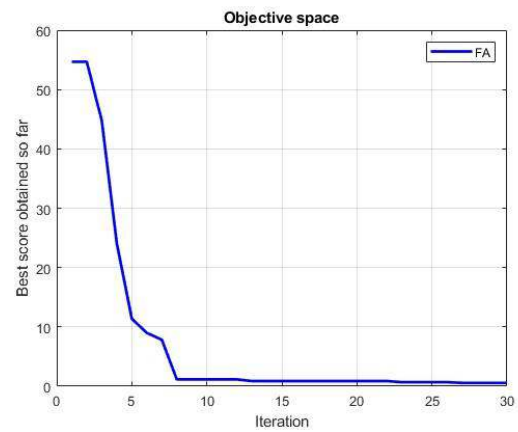
La Figura 3.22 que se presenta consta de seis imágenes que muestran los valores de fitness obtenidos por seis algoritmos diferentes: BAT, FA, HHO, MPA, PSO y SMA.

Cada imagen representa el valor de la función fitness de cada algoritmo específico y muestra el progreso de la aptitud o fitness a lo largo del tiempo de ejecución.

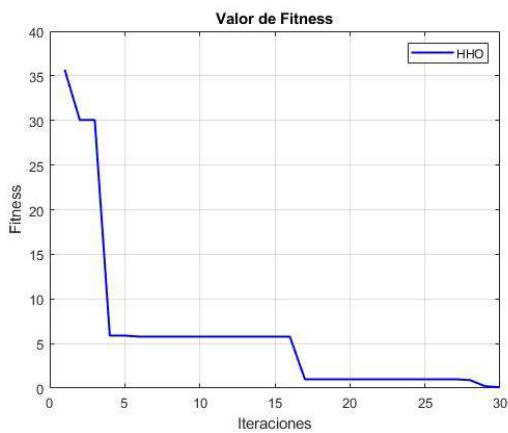
En general, se puede observar que todos los algoritmos muestran una disminución en los valores de aptitud a lo largo del tiempo de ejecución, aunque con diferentes patrones de comportamiento. La visualización de estos patrones de comportamiento es importante para entender el desempeño de los algoritmos y poder seleccionar el más adecuado para resolver un problema específico.



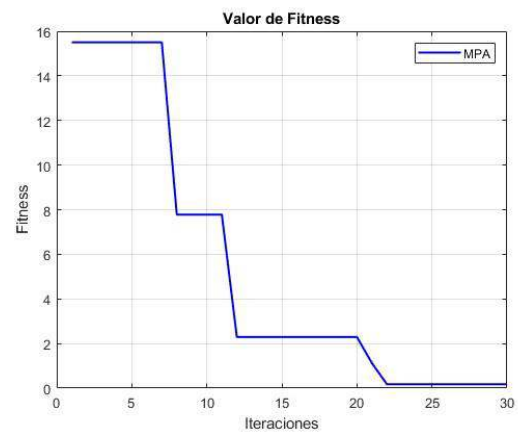
(a) Función fitness usando el algoritmo BAT.



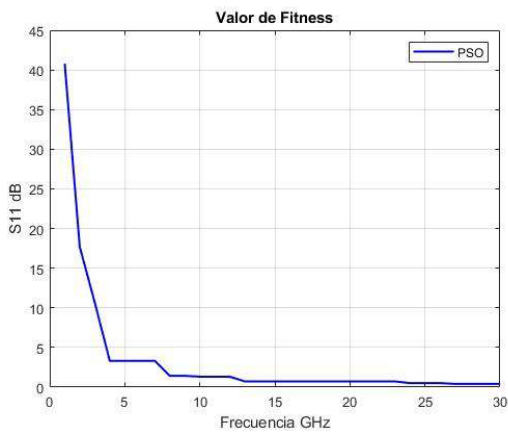
(b) Función fitness usando el algoritmo FA.



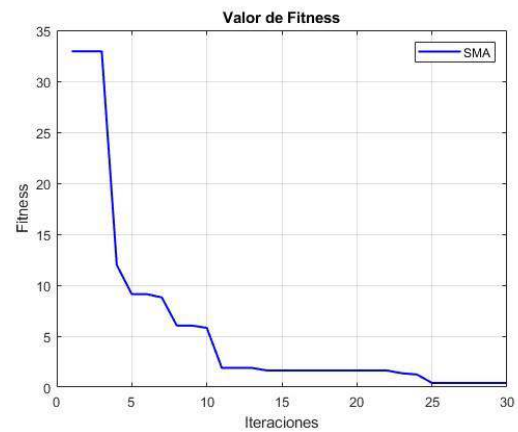
(c) Función fitness usando el algoritmo HHO.



(d) Función fitness usando el algoritmo MPA.

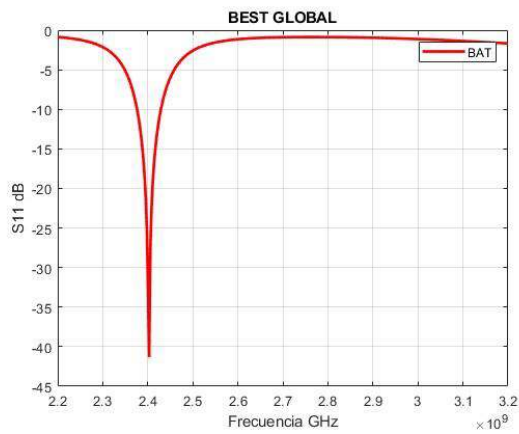


(e) Función fitness usando el algoritmo PSO.

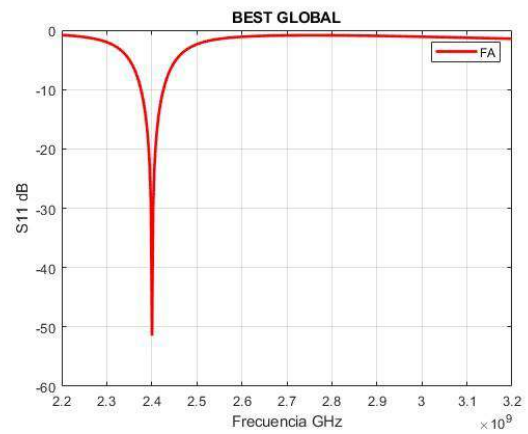


(f) Función fitness usando el algoritmo SMA.

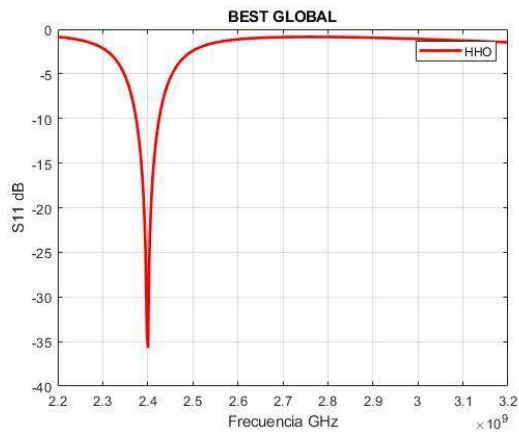
Figura 3.22: Resultados de convergencia obtenidos para la antena con los seis algoritmos.



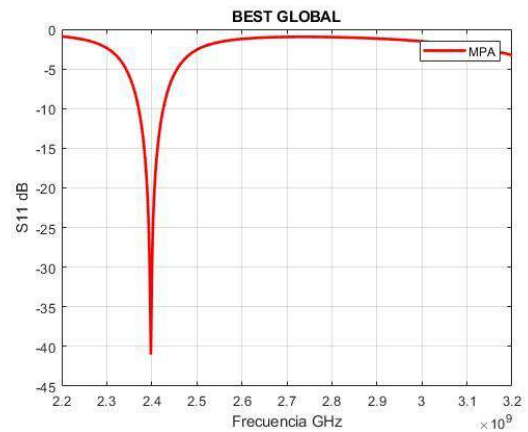
(a) Mejor parámetro de reflexión de la antena usando el algoritmo BAT.



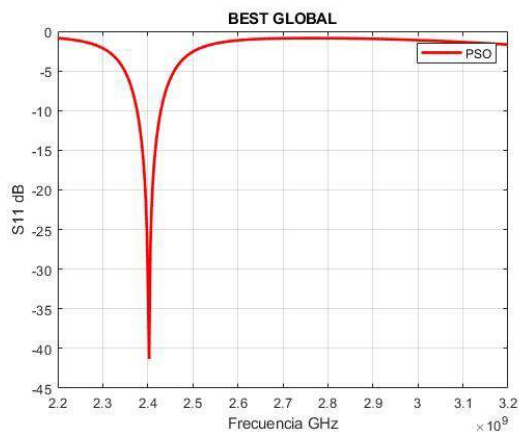
(b) Mejor parámetro de reflexión de la antena usando el algoritmo FA.



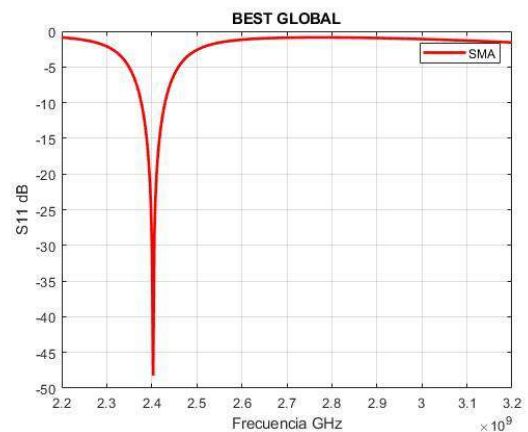
(c) Mejor parámetro de reflexión de la antena usando el algoritmo HHO.



(d) Mejor parámetro de reflexión de la antena usando el algoritmo MPA.



(e) Mejor parámetro de reflexión de la antena usando el algoritmo PSO.



(f) Mejor parámetro de reflexión de la antena usando el algoritmo SMA.

Figura 3.23: Los mejores parámetros de reflexión de la antena usando algoritmos de optimización.

La Figura 3.23 en cuestión muestra seis imágenes que representan los resultados de la optimización de parámetros de reflexión de una antena utilizando diferentes algoritmos de optimización. En particular, se han utilizado los algoritmos BAT, FA, HHO, MPA, PSO y SMA para llevar a cabo la optimización.

Cada imagen muestra los mejores valores de los parámetros de reflexión obtenidos para cada uno de los algoritmos mencionados. Los parámetros de reflexión son importantes en la antena ya que controlan la cantidad de señal que se refleja de vuelta hacia la antena y la cantidad de señal que se transmite hacia el receptor. En este caso el que tiene mayor resonancia son las FA, que trabaja en el frecuencia de 2,4GHz con una resonancia de $-51,4699$.

Mejores resultados

De forma resumida y con los mejores resultados del S11, función fitness, dimensiones de la antena (W, L, Y_0) y el tiempo de ejecución, en la Tabla 3.8. Se puede comparar en base a estas variables.

Si tomamos en cuenta el tiempo el mejor es el SMA ya que se demora 05:14:21 en optimizar la antena con un valor de fitness del 0.409522. En cambio, si se toma en cuenta la resonancia el mejor es FA ya que llega a resonar en la frecuencia de 2,4GHz a -51.4699 , pero este algoritmo se demora 11:11:28 en converger. Sin embargo, otro de los parámetros que se toma en cuenta es la función fitness que en este caso el mejor es el HHO teniendo un valor de fitness de 0.10655 con una resonancia de -36.2926 en la frecuencia que se trabaja y se toma un tiempo de 06:31:54. Estos tres algoritmos se consideran los mejores para la optimización de antenas patch rectangulares, ya dependerá del usuario que algoritmo utilizara para la optimización.

Tabla 3.8: Resultados obtenidos.

Algoritmo	S11 [dB]	Fitness	W[mm]	L[mm]	Yo[mm]	Tiempo
Antena Inicial	-8.2		38.04	29.48	10.94	00:02:06
Bat	-41,3033	0,562172	28,4814	36,9324	8,1651	11:11:28
FA	-51,4699	0,163755	28,8897	31,5716	9,7370	11:36:48
HHO	-36,2926	0,106555	28,895358	33,24148798	9,161602395	06:31:54
MPA	-41,025	0,174525	40	28.4086	7.3640	07:58:36
PSO	-38,3364	0,405291	28,5029	34,4796	9,0227	06:38:10
SMA	-48,2338	0,409522	28,755479	36,6446828	8,483955143	05:14:21

En este capítulo se han presentado las pruebas y análisis de resultados obtenidos en el desarrollo del aplicativo de optimización de antenas patch rectangulares.

En el primer subtema se ha explicado acerca de los parámetros que tiene el aplicativo, como la función fitness, el criterio de paro, S11 y resonancia. Cada uno de estos parámetros juega un papel importante en la optimización del patch de la antena.

En el segundo subtema se ha descrito el funcionamiento del aplicativo en dos fases. En la primera fase, se generan los valores iniciales del patch de la antena y en la segunda fase, se escoge uno de los seis algoritmos de optimización para optimizar las dimensiones del patch.

En el tercer subtema se han presentado las pruebas de los seis algoritmos implementados, que son el Bat Algorithm, Firefly Algorithm, Harris Hawks Optimization, Marine Predators Algorithm, Particle Swarm Optimization y Slime Mould Algorithm. Cada algoritmo se ha ejecutado con 30 iteraciones y 10 agentes de búsqueda para la optimización, y se ha registrado el valor de los 30 parámetros de reflexión de la antena (S11) y el mejor resultado obtenido. Además, se ha presentado de forma gráfica la función fitness y una tabla con las funciones fitness de cada algoritmo.

En el último subtema, se ha llevado a cabo un análisis y comparación de los resultados obtenidos de los seis algoritmos implementados. Se han comparado los mejores resultados de cada algoritmo y se ha evaluado cuál es el mejor algoritmo en cada caso, dependiendo del criterio utilizado.

En conclusión, el aplicativo de optimización de antenas patch rectangulares

ha sido exitoso en la optimización de los parámetros de la antena, utilizando los seis algoritmos implementados. Los resultados obtenidos han sido analizados y comparados, y se ha determinado cuál es el mejor algoritmo en cada caso. Esto proporciona información valiosa para futuras investigaciones en el campo de las antenas patch rectangulares y su optimización.

Capítulo 4

Conclusiones y Trabajos Futuros

En el presente trabajo se ha demostrado que la optimización de antenas patch rectangulares mediante algoritmos metaheurísticos es una herramienta efectiva para mejorar las prestaciones de las antenas. Se ha comprobado que los algoritmos metaheurísticos permiten resolver problemas de optimización complejos de manera eficiente y han demostrado ser útiles en la optimización de las antenas patch. Además, se ha observado que la combinación de diferentes algoritmos metaheurísticos y técnicas puede dar lugar a resultados aún más optimizados.

En cuanto a los objetivos propuestos al inicio del trabajo, se puede afirmar que han sido alcanzados satisfactoriamente. Se ha logrado demostrar que la optimización de antenas patch mediante algoritmos metaheurísticos es una técnica valiosa en el diseño de sistemas de comunicación es un avance significativo en el campo de las telecomunicaciones. La investigación y el desarrollo continuo en este campo podrían llevar a mejoras significativas en la eficiencia, el rendimiento y la confiabilidad de los sistemas de comunicación. Se han evaluado diferentes algoritmos de optimización metaheurísticos que como se plantearon son seis: Bat Algorithm, Firefly Algorithm, Harris Hawks Optimization, Marine Predators Algorithm, Particle Swarm Optimization y Slime Mould Algorithm, donde se evaluaron parámetros como: la frecuencia de resonancia, función fitness y las dimensiones del patch (W, L, Y_0). Luego, con los algoritmos ya mencionados se utilizan las técnicas de optimización de cada uno de ellos para tener una antena patch rectangular optimizada.

Donde los mejores algoritmos de optimización, tomando como parámetros de

comparación: la reflexión de la antena, tiempo de ejecución y función fitness. Los mejores son FA, HHO y SMA de los seis algoritmos. Por un lado, si tomamos en cuenta el tiempo, el mejor es el SMA con un S_{11} de -48.2338 dB y un error de 0.40 en la función fitness. En cambio, si tomamos en cuenta la función fitness, el mejor es el HHO teniendo un error de 0.174; aunque este algoritmo se demora 06:31:54. Por otro lado, si tomamos en cuenta la resonancia en la frecuencia de 2.4 GHz, el mejor es la FA teniendo como parámetro de reflexión de -51.4699 dB. En el Capítulo 3 se mostraron los resultados de cada uno de los algoritmos y como se fue optimizando hasta obtener los mejores resultados.

En las recomendaciones se sugiere continuar investigando y desarrollando nuevas técnicas de optimización para antenas patch, que permitan obtener resultados aún más optimizados. Además, se recomienda realizar estudios más detallados de la influencia de los diferentes parámetros en la optimización de las antenas patch, para mejorar la comprensión de los procesos de optimización y poder obtener mejores resultados de la antena. Basado en las características mencionadas, se recomienda al usuario adquirir una computadora con un mínimo de 16 Gigas de RAM y un procesador de al menos 3.2 GHz para poder utilizar eficientemente la interfaz del aplicativo. También se debe considerar la capacidad de almacenamiento del disco duro para garantizar un buen desempeño en el manejo de archivos y datos.

Trabajos a futuro:

Como posibles trabajos a futuro, se sugiere continuar investigando en la optimización de antenas patch mediante algoritmos metaheurísticos y explorar nuevas técnicas y combinaciones de algoritmos que permitan obtener resultados aún más precisos y eficientes. Asimismo, se sugiere aplicar los resultados obtenidos en este trabajo a la implementación de sistemas de comunicación inalámbrica, y evaluar su desempeño en un contexto real. Se puede implementar mas diseños para antenas como pueden ser circulares, hexagonales entre otras. Así mismo, se puede acoplar a mas softwares dependiendo de las necesidades requeridas, Cabe mencionar que se podrían construir las antenas acoplando una cortadora láser al programa para que este mismo pueda diseñarlos.

Además, se pueden explorar nuevas aplicaciones de la optimización de antenas

patch, como en sistemas de radar y teledetección.

Anexos

4.1. Anexo 1

En esta sección se encuentra un manual de usuarios utilizando el aplicativo en dos escenarios de optimización.



FORMATO DE GUÍA DE PRÁCTICAS DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTE

ESTUDIANTE:	
CARRERA: TELECOMUNICACIONES	ASIGNATURA: Antenas
NRO. PRÁCTICA:	1 TÍTULO PRÁCTICA: Tipos de Antenas
OBJETIVOS: OBJETIVO GENERAL: Diseñar una antena patch rectangular que resuene a una frecuencia de 6.94 GHz usando el aplicativo de Matlab. OBJETIVOS ESPECÍFICOS: <ul style="list-style-type: none"> • Calcular las dimensiones iniciales del patch. • Optimizar la antena patch usando uno de los algoritmos de optimización del aplicativo 	
INSTRUCCIONES (Detallar las instrucciones para esta práctica.)	<ol style="list-style-type: none"> 1. Diseñe una antena patch rectangular, que resuene a una frecuencia de 6.94 GHz. Luego, compare los valores calculados con los que le da el aplicativo en “First Phase”. 2. En la “Second Phase” del aplicativo escoja uno de los algoritmos de optimización. 3. Diseñe una antena patch rectangular, que resuene a una frecuencia de 2.4 GHz. Luego, compare los valores calculados con los que le da el aplicativo en “First Phase”. 4. En la “Second Phase” del aplicativo escoja uno de los algoritmos de optimización.
ACTIVIDADES DESARROLLADAS	
(Anotar todas las actividades que se siguió para el desarrollo de la práctica. Agregue filas si necesita más espacio)	
1. LISTADO DE HERRAMIENTAS Y MATERIALES <ul style="list-style-type: none"> • Matlab • HFSS 	
2. EXPERIMENTOS Diseño de una antena patch como la de la Figura 1, a una frecuencia de operación de 6.94 GHz sobre FR4 con $\epsilon_r = 4.4$ y con una altura del substrato $h = 1.5mm = 0.0015m$	

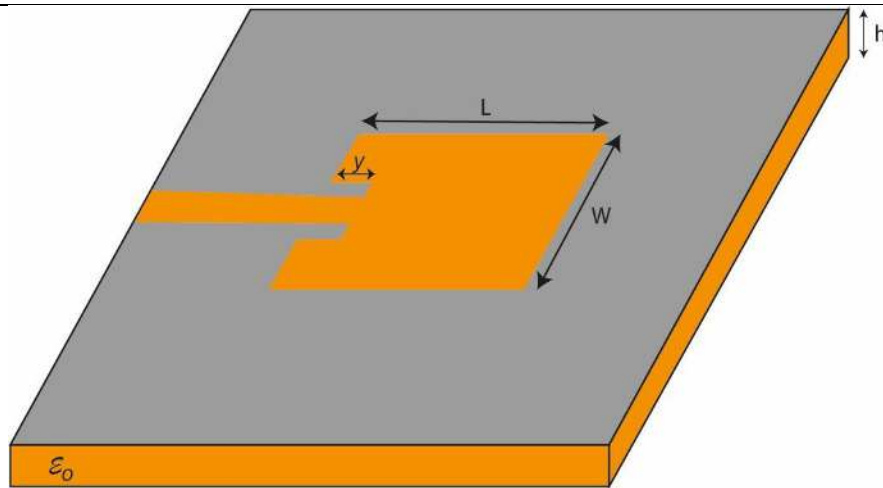


Figura 1 Antena patch rectangular

Cálculos:

- Ancho de la antena patch (W):

$$W = \frac{c}{2 \cdot f_0 \cdot \sqrt{\frac{\epsilon_r + 1}{2}}} = \frac{3 \times 10^8 \text{ [m/s]}}{2 \cdot 6.94 \times 10^9 \text{ [Hz]} \cdot \sqrt{\frac{4.4 + 1}{2}}} = 0.01315 \text{ [m]} = 13.16 \text{ mm}$$

- Longitud real del patch (L):

$$L = L_{eff} - 2\Delta L$$

Donde: m

La longitud efectiva del patch L_{eff} (media onda) es:

$$L_{eff} = \frac{c}{2 \cdot f_0 \cdot \sqrt{\epsilon_{reff}}}$$

$$\epsilon_{reff} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \cdot \left[1 + 12 \frac{h}{W} \right]^{-\frac{1}{2}} = \frac{4.4 + 1}{2} + \frac{4.4 - 1}{2} \cdot \left[1 + 12 \frac{0.0015 \text{ [mm]}}{0.01315 \text{ [mm]}} \right]^{-\frac{1}{2}}$$

$$= 3.80454$$

$$L_{eff} = \frac{3 \times 10^8 \text{ [m/s]}}{2 \cdot 6.94 \times 10^9 \text{ [Hz]} \cdot \sqrt{3.80454}} = 0.01108$$

El incremento de L (ΔL) está definido por:

$$\Delta L = 0.412h \cdot \frac{(\epsilon_{reff} + 0.3) \left(\frac{W}{h} + 0.264 \right)}{(\epsilon_{reff} - 0.258) \left(\frac{W}{h} + 0.8 \right)}$$

$$= 0.412(0.0015 \text{ [mm]}) \cdot \frac{(3.80454 + 0.3) \left(\frac{0.01315 \text{ [mm]}}{0.0015 \text{ [mm]}} + 0.264 \right)}{(3.80454 - 0.258) \left(\frac{0.01315 \text{ [mm]}}{0.0015 \text{ [mm]}} + 0.8 \right)} = 0.00029$$

$$L = L_{eff} - 2\Delta L = 0.009731 \text{ m} = 9.731 \text{ mm}$$

- La inserción Y_0 para una resistencia de entrada adecuada (por ejemplo, de 50 Ω). La fórmula para 50 Ω es:

$$Y_o = \frac{L \times \arccos\left(\sqrt{\frac{50}{R_{in}(Y=0)}}\right)}{\pi} = 3.597 \text{ mm}$$

Para esto se debe calcular la resistencia de entrada de la inserción

$$R_{in}(Y=0) = \frac{1}{2(G_1 \pm G_{21})} \cos^2\left(\frac{\pi}{L} y_o\right)$$

Donde la conductividad de una sola ranura:

$$G_1 = \frac{I_1}{120\pi^2}$$

$$I_1 = -2 + \cos(X) + X S_i(X) + \frac{\sin(X)}{X}$$

$$X = k_o W$$

$$k_o = \frac{2\pi}{\lambda_o}$$

En este caso también se debe tomar en cuenta el efecto de acoplamiento que existe entre las ranuras de la antena:

$$G_{12} = \frac{1}{120\pi^2} \int_0^\pi \left[\frac{\sin\left(\frac{k_o W}{2} \cos \theta\right)}{\cos \theta} \right]^2 J_o(k_o \cdot L \cdot \sin \theta) \sin^3 \theta d\theta$$

- Plano de tierra L_g

$$L_g = 6h + L = 6(1.5) + 9.731 = 18.731 \text{ [mm]}$$

- Plano de tierra W_g

$$W_g = 6h + W = 6(1.5) + 13.16 = 22.16 \text{ [mm]}$$

Finalmente, para comparar los valores se ve en la "First Phase" del aplicativo. Donde se debe ingresar los valores iniciales como: la constante dieléctrica, la frecuencia, valor de búsqueda, impedancia, altura del substrato y el barrido; en la Figura 2 se observa con recuadros azules, donde se debe ingresar los datos. Finalmente, en el recuadro rojo se debe dar clic para calcular.

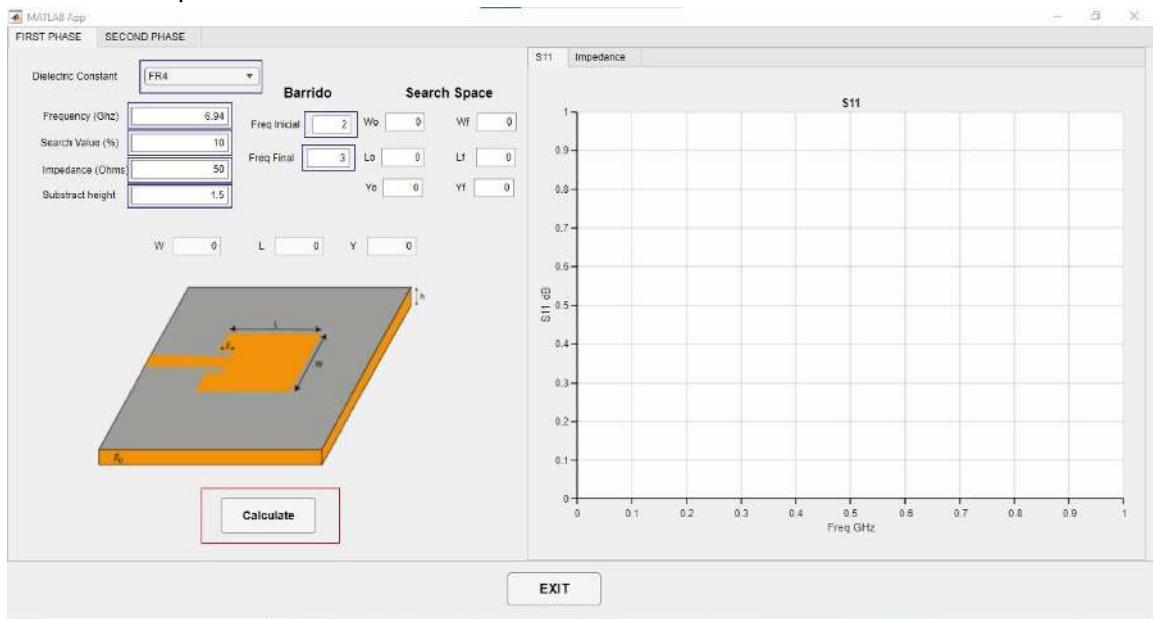


Figura 2 Ingreso de datos en "First Phase" del aplicativo

Como se ve en la Figura 3, se encuentran los cálculos iniciales de la antena, donde el S11 se observa que no tiene buenos resultados realizando los cálculos.

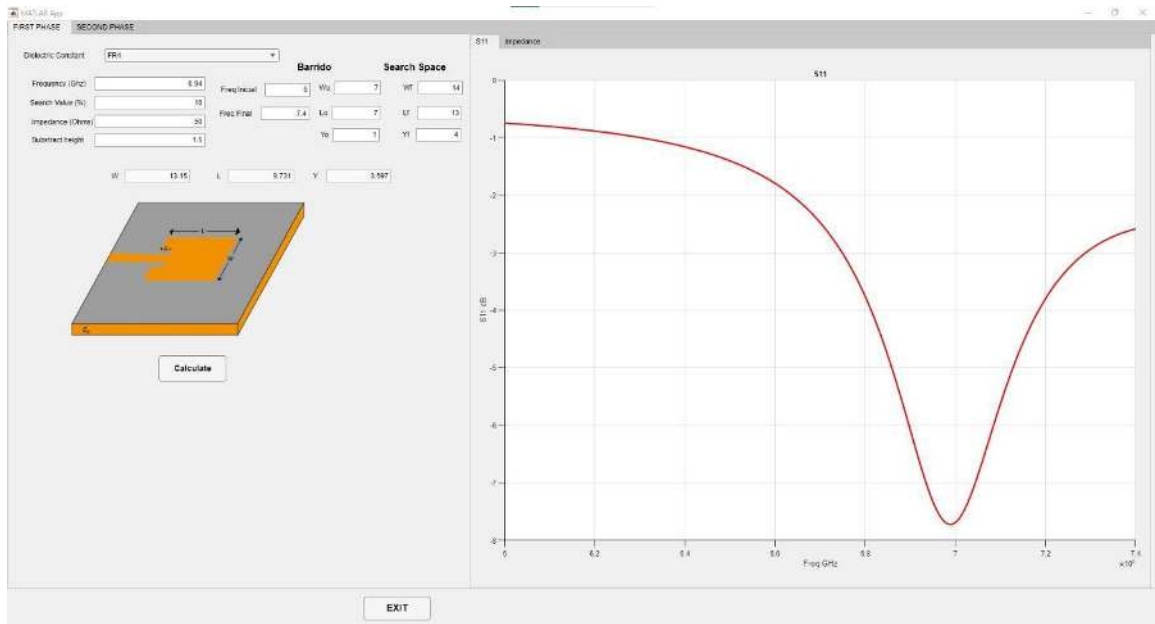


Figura 3 First Phase del aplicativo

En "Second Phase" que se ve gráficamente en la Figura 4, se debe seleccionar uno de los seis algoritmos de optimización para antenas, el número de iteraciones y el número de agentes, para darle clic en el botón START para que empiece a calcular. En este caso se escogió el algoritmo SMA (Slime Mould Algorithm) con 30 iteraciones con 10 agentes.

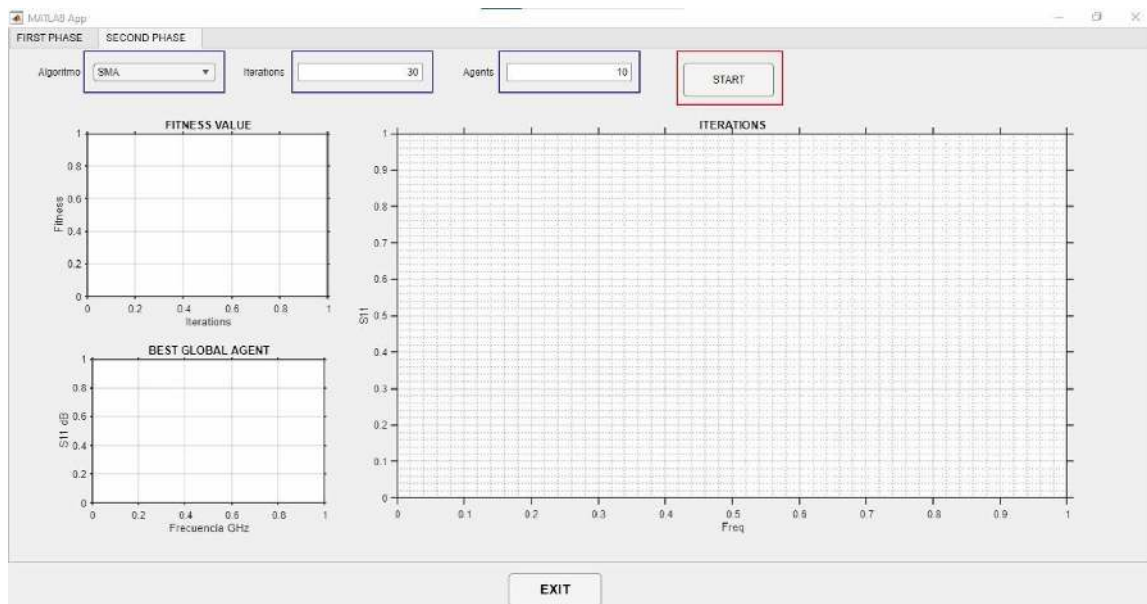


Figura 4 Ingreso de datos en "Second Phase"

En cada una de las iteraciones se va a ir presentando una ventana, como en la Figura 5 donde se puede visualizar como va optimizando en cada uno de los 10 agentes.

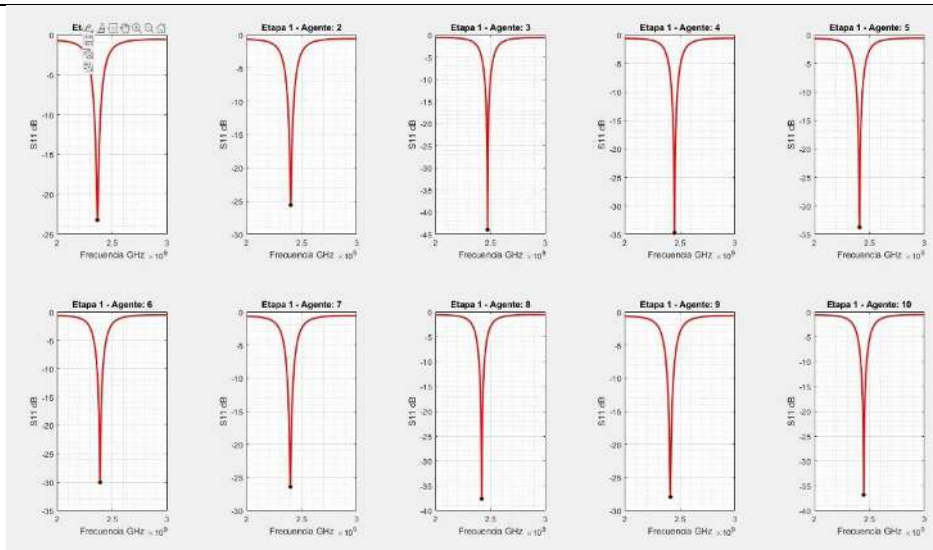


Figura 5 Ventana de una iteración

Cuando termine las 30 iteraciones se verá los resultados que obtuvo el algoritmo como en la Figura 6, donde se puede ver gráficamente el valor fitness, el mejor global del S11 y los mejores globales de las 30 iteraciones; y como ventanas emergentes: las dimensiones de la antenna y el mejor de la función fitness.

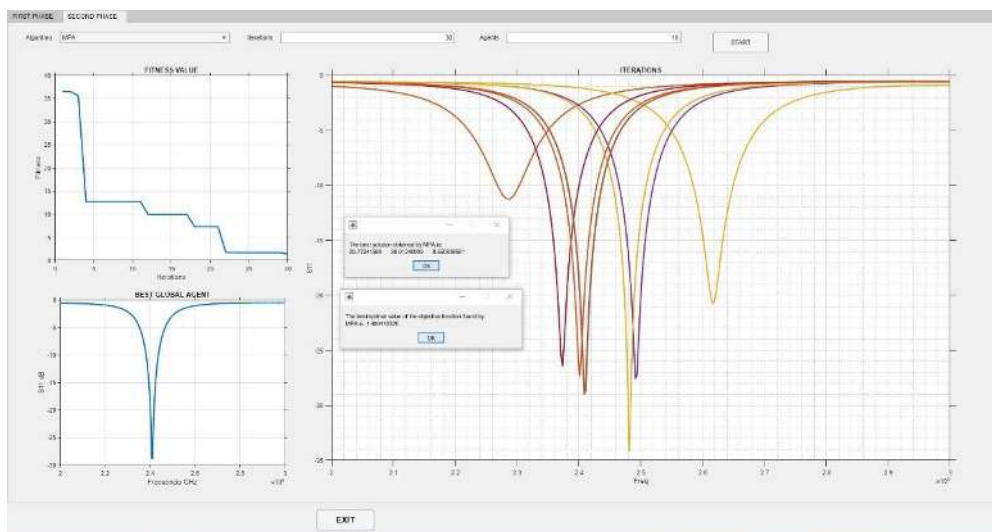


Figura 6 Resultados de optimización de la antenna

Resultados:

En la Tabla 1 se puede observar cómo va optimizando el algoritmo SMA en cada una de las iteraciones, en algunos de los casos se ve que se mantiene o varía.

Tabla 1 Resultados de la función fitness de las 30 iteraciones

Iteración	Valor fitness
1	46.9918
2	25.2387
3	25.2387
4	19.3735
5	19.3735
6	19.3735
7	15.534

8	4.85524
9	3.04698
10	3.04698
11	3.04698
12	1.11263
13	1.11263
14	1.11263
15	1.11263
16	1.00764
17	1.00764
18	1.00764
19	0.410065
20	0.410065
21	0.410065
22	0.410065
23	0.402515
24	0.402515
25	0.402515
26	0.0487977
27	0.0487977
28	0.0487977
29	0.0487977
30	0.0487977

En la Tabla 2 se puede comparar los valores de las dimensiones la antenna calculada y los valores optimizados por el algoritmo.

Tabla 2 Dimensiones de la antenna

Dimensiones de la antenna	Valor calculado	Valor optimizado
W [mm]	38.04	29.77241589
L [mm]	29.48	36.01240009
Yo [mm]	10.94	8.550839581

3. Diseño de una antenna patch como la de la Figura 1, a una frecuencia de operación de 2.4 GHz sobre FR4 con $\epsilon_r = 4.4$ y con una altura del substrato $h = 1.5\text{mm} = 0.0015\text{m}$.

Cálculos:

- Ancho de la antenna patch (W):

$$W = \frac{c}{2 \cdot f_o \cdot \sqrt{\frac{\epsilon_r + 1}{2}}} = \frac{3 \times 10^8 \text{ [m/s]}}{2 \cdot 2.4 \times 10^9 \text{ [Hz]} \cdot \sqrt{\frac{4.4 + 1}{2}}} = 0.03803 \text{ [m]} = 38.03 \text{ mm}$$

- Longitud real del patch (L):

$$L = L_{eff} - 2\Delta L$$

Donde:m

La longitud efectiva del patch L_{eff} (media onda) es:

$$L_{eff} = \frac{c}{2 \cdot f_o \cdot \sqrt{\epsilon_{reff}}}$$

$$\epsilon_{reff} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \cdot \left[1 + 12 \frac{h}{W}\right]^{-\frac{1}{2}} = \frac{4.4 + 1}{2} + \frac{4.4 - 1}{2} \cdot \left[1 + 12 \frac{0.0015 [mm]}{0.03803 [mm]}\right]^{-\frac{1}{2}}$$

$$= 4.10056$$

$$L_{eff} = \frac{3 \times 10^8 [m/s]}{2 \cdot 6.94 \times 10^9 [Hz] \cdot \sqrt{4.10056}} = 0.01115$$

El incremento de L (ΔL) está definido por:

$$\Delta L = 0.412h \cdot \frac{(\epsilon_{reff} + 0.3) \left(\frac{W}{h} + 0.264\right)}{(\epsilon_{reff} - 0.258) \left(\frac{W}{h} + 0.8\right)}$$

$$= 0.412(0.0015 [mm]) \cdot \frac{(4.10056 + 0.3) \left(\frac{0.03803 [mm]}{0.0015 [mm]} + 0.264\right)}{(4.10056 - 0.258) \left(\frac{0.03803 [mm]}{0.0015 [mm]} + 0.8\right)} = 0.00069$$

$$L = L_{eff} - 2\Delta L = 0.00069 m = 9.77 mm$$

- La inserción Y_o para una resistencia de entrada adecuada (por ejemplo, de 50Ω). La fórmula para 50Ω es:

$$Y_o = \frac{L \times \arccos \left(\sqrt{\frac{50}{R_{in}(Y=0)}} \right)}{\pi} = 10.94 mm$$

Para esto se debe calcular la resistencia de entrada de la inserción

$$R_{in}(Y=0) = \frac{1}{2(G_1 \pm G_{21})} \cos^2 \left(\frac{\pi}{L} y_o \right)$$

Donde la conductividad de una sola ranura:

$$G_1 = \frac{I_1}{120\pi^2}$$

$$I_1 = -2 + \cos(X) + X S_i(X) + \frac{\sin(X)}{X}$$

$$X = k_o W$$

$$k_o = \frac{2\pi}{\lambda_o}$$

En este caso también se debe tomar en cuenta el efecto de acoplamiento que existe entre las ranuras de la antena:

$$G_{12} = \frac{1}{120\pi^2} \int_0^\pi \left[\frac{\sin \left(\frac{k_o W}{2} \cos \theta \right)}{\cos \theta} \right]^2 J_o(k_o \cdot L \cdot \sin \theta) \sin^3 \theta d\theta$$

- Plano de tierra L_g

$$L_g = 6h + L = 6(1.5) + 9.77 = 18.77 [mm]$$

- Plano de tierra W_g

$$W_g = 6h + W = 6(1.5) + 38.03 = 47.03 [mm]$$

Finalmente, para comparar los valores se ve en la "First Phase" del aplicativo. Donde se debe ingresar los valores iniciales como: la constante dieléctrica, la frecuencia, valor de búsqueda, impedancia, altura del sustrato y el barrido; en la Figura 7 se observa con recuadros azules, donde se debe ingresar los datos. Finalmente, en el recuadro rojo se debe dar clic para calcular.

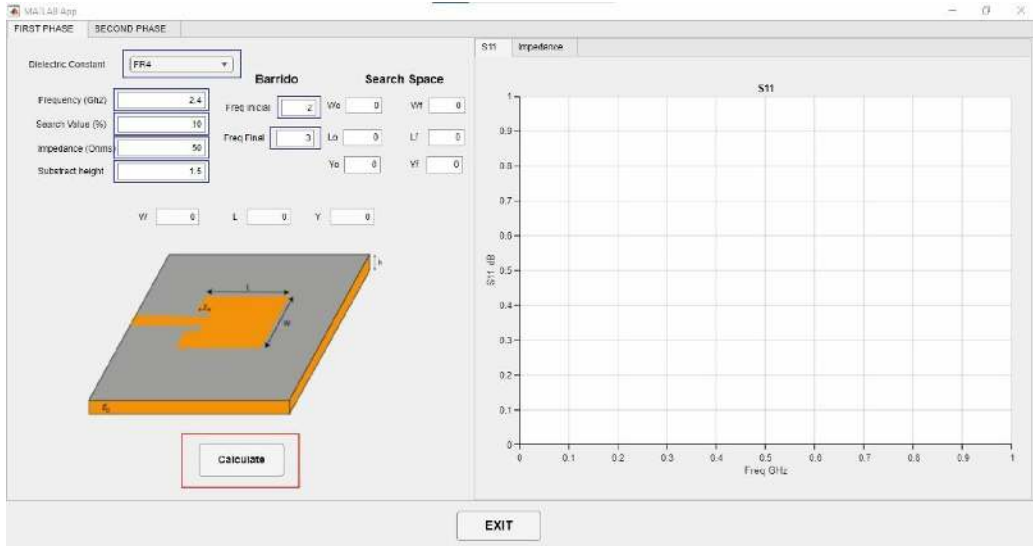


Figura 7 Ingreso de datos en "First Phase" del aplicativo en $F=2.4\text{GHz}$

Como se ve en la Figura 8, se encuentran los cálculos iniciales de la antena, donde el S11 se observa que no tiene buenos resultados realizando los cálculos.

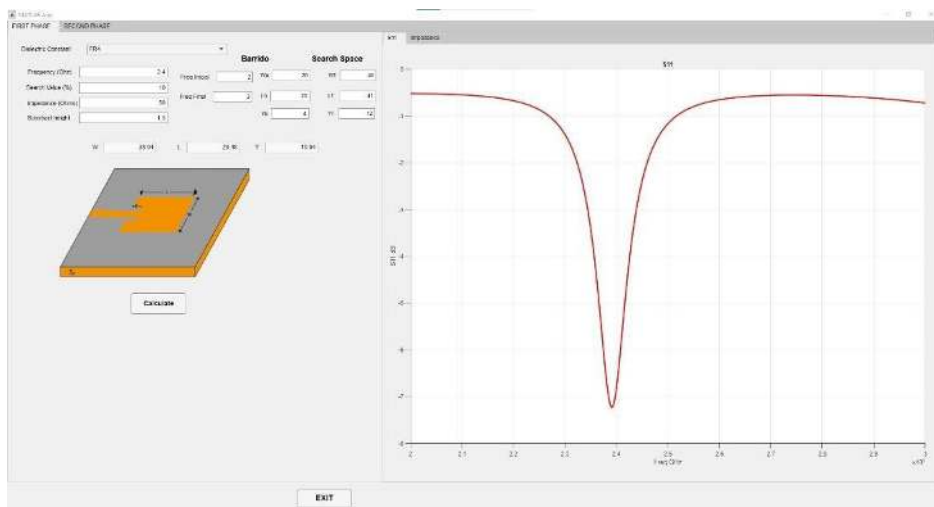


Figura 8 First Phase del aplicativo

En "Second Phase" que se ve gráficamente en la Figura 4, se debe seleccionar uno de los seis algoritmos de optimización para antenas, el número de iteraciones y el número de agentes, para darle clic en el botón START para que empiece a calcular. En este caso se escogió el algoritmo HHO (Harris Hawks Optimization) con 30 iteraciones con 10 agentes.

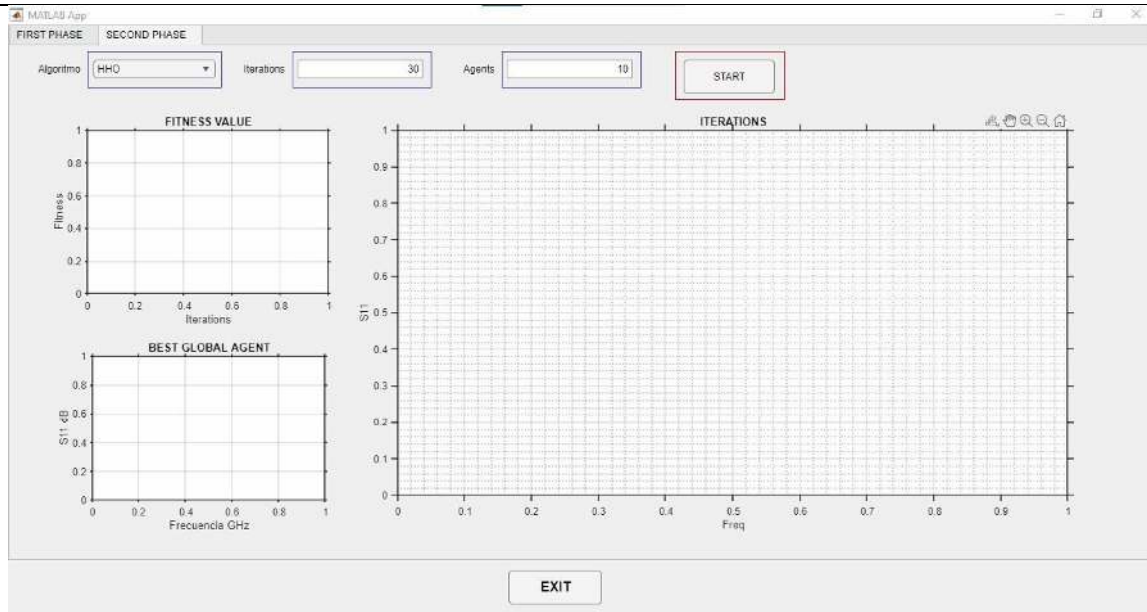


Figura 9 Ingreso de datos en "Second Phase"

En cada una de las iteraciones se va a ir presentando una ventana, como en la Figura 10 donde se puede visualizar como va optimizando en cada uno de los 10 agentes.

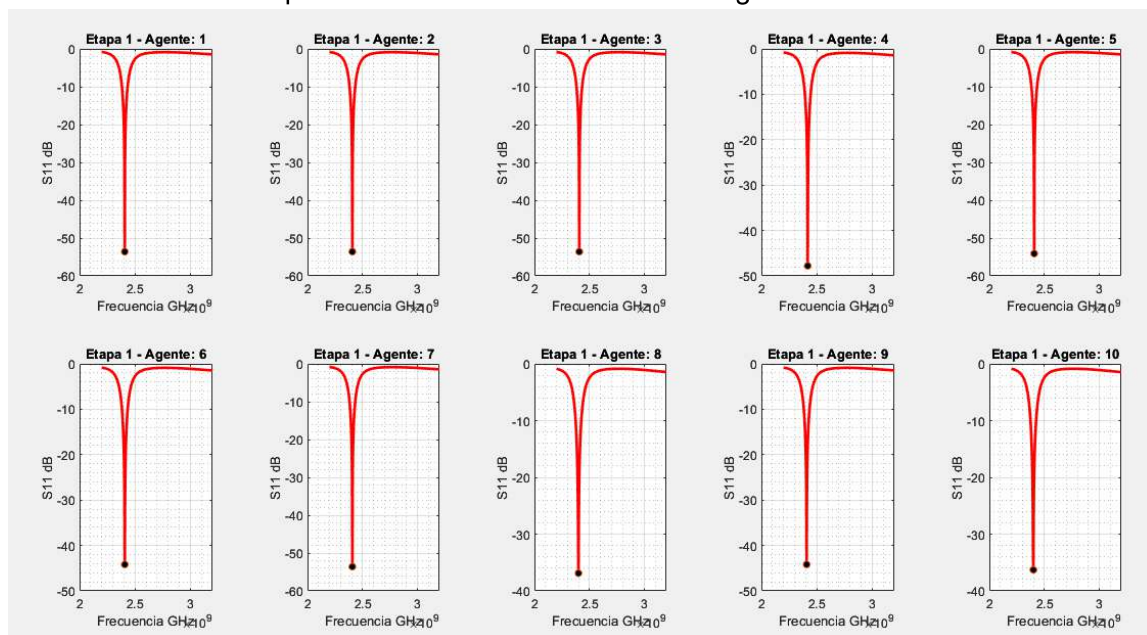


Figura 10 Ventana de una iteración

Cuando termine las 30 iteraciones se verá los resultados que obtuvo el algoritmo como en la Figura 11, donde se puede ver gráficamente el valor fitness, el mejor global del S11 y los mejores globales de las 30 iteraciones; y como ventanas emergentes: las dimensiones de la antena y el mejor de la función fitness.

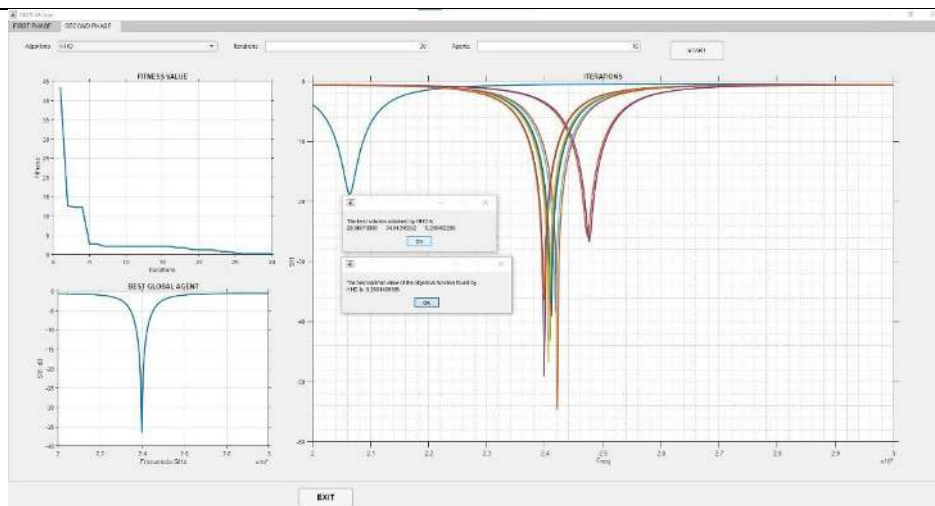


Figura 11 Resultados de optimización de la antena

Resultados:

En la Tabla 3 se puede observar cómo va optimizando el algoritmo HHO en cada una de las iteraciones, en algunos de los casos se ve que se mantiene o varía.

Tabla 3 Resultados de la función fitness de las 30 iteraciones

Iteración	Valor fitness
1	43.4093
2	12.6944
3	12.3338
4	12.3338
5	2.81421
6	2.81421
7	2.17854
8	2.17854
9	2.17854
10	2.17854
11	2.17854
12	2.17854
13	2.17854
14	2.17854
15	2.17854
16	2.17854
17	1.83489
18	1.83489
19	1.34927
20	1.27788
21	1.27788
22	1.21755
23	0.798464
24	0.798464
25	0.502996
26	0.266146
27	0.266146

28	0.266146
29	0.266146
30	0.266146

En la Tabla 4 se puede comparar los valores de las dimensiones la antena calculada y los valores optimizados por el algoritmo.

Tabla 4 Dimensiones de la antena

Dimensiones de la antena	Valor calculado	Valor optimizado
W [mm]	38.04	29.98070389
L [mm]	29.48	34.01245352
Yo [mm]	10.94	9.299462268

CONCLUSIONES:

El algoritmo SMA es un método de optimización basado en el comportamiento de los mohos mucilaginosos, los cuales son capaces de encontrar el camino más corto para conectar diferentes puntos en su entorno. En cambio, el algoritmo HHO es un algoritmo metaheurístico que utiliza el comportamiento de caza de los halcones de Harris para resolver problemas de optimización. El algoritmo simula la interacción entre los halcones y utiliza operadores de búsqueda y de movimiento inspirados en su comportamiento para mejorar la calidad de la solución. HHO ha demostrado ser eficaz en la resolución de problemas de optimización complejos en diversos campos. En ambos casos, se ha aplicado un algoritmo para optimizar las dimensiones de una antena patch rectangular.

Para ello, se han definido tres variables: W, L y Yo, que representan el ancho, largo y la altura del sustrato de la antena, respectivamente. El objetivo es encontrar los valores óptimos de estas variables para maximizar la eficiencia de la antena en una frecuencia específica de resonancia.

El algoritmo SMA ha sido ejecutado durante un tiempo de 05:14:21. Sin embargo, el algoritmo HHO tiene un tiempo de ejecución de 07:34:57. En cada caso se han generado una serie de soluciones candidatas que han sido evaluadas mediante una función fitness. En cada iteración, se ha seleccionado la solución con el valor de fitness más bajo como punto de partida para la siguiente iteración.

El resultado final ha sido una antena patch rectangular optimizada, cuyas dimensiones son diferentes a las calculadas inicialmente. Por un lado, el algoritmo SMA ha encontrado una combinación de valores de W, L y Yo que ha permitido obtener una resonancia de -48.1996 en la frecuencia de 6.94 GHz. Por otro lado, para la frecuencia de resonancia de 2.4 GHz utilizando el algoritmo HHO resuena a -36.3024, así mismo variando las dimensiones del patch inicial a comparación del patch optimizado.

Estos resultados demuestran la capacidad de los algoritmos SMA y HHO, para encontrar soluciones óptimas en problemas de optimización complejos, y sugieren que este método podría ser útil para diseñar antenas u otros sistemas de comunicaciones. Además, cabe destacar que el algoritmo ha sido capaz de encontrar una solución más eficiente que la inicialmente propuesta, lo que indica que la exploración sistemática de soluciones mediante SMA o HHO puede llevar a resultados sorprendentes e innovadores.

En conclusión, la aplicación de los algoritmos para optimizar la antena patch rectangular ha demostrado su capacidad para encontrar soluciones óptimas en un problema de optimización complejo. Además,

el algoritmo ha permitido descubrir una combinación de valores de dimensiones que mejora la eficiencia de la antena en la frecuencia de resonancia. Estos resultados sugieren que el algoritmo SMA y HHO podría ser una herramienta útil para el diseño de antenas y otros sistemas de comunicaciones.

RECOMENDACIONES

Se recomienda enfocarse en la comparación de los algoritmos de optimización, como son: Bat Algorithm, Firefly Algorithm, Harris Hawks Optimization, Marine Predators Algorithm, Particle Swarm Optimization y Slime Mould Algorithm del aplicativo, de los resultados obtenidos por los diferentes algoritmos de optimización mencionados en el estudio, así como en la evaluación del desempeño de las antenas patch rectangulares en diferentes frecuencias para aplicaciones de comunicaciones inalámbricas.

Es importante destacar los principales hallazgos y conclusiones del estudio, así como destacar las ventajas y limitaciones de cada uno de los algoritmos de optimización y la efectividad de las antenas patch rectangulares en diferentes frecuencias.

Se sugiere también presentar posibles mejoras y futuras líneas de investigación para profundizar en el tema, como la implementación de combinaciones de diferentes algoritmos de optimización o la exploración de otras formas de antenas para aplicaciones específicas.

REFERENCIAS:

Glosario

ABHA Artificial Bee Hive Algorithm.

ACO Ant Colony Optimization.

BA Bat Algorithm.

BFO Bacteril Foraging Optimization.

BSFO Bacteria Swarm Foraging Optimization.

CS Cuckoo Search .

DE Differential Evolution.

FA Firefly Algorithm .

GA Genetic Algorithm.

HHO Harris Hawks Optimization.

MPA Marine Predators Algorithm.

PSO Particle Swarm Optimization.

SDS Stochastic Diffusion Search.

SHF Super High Frecuency.

SI Swarm Intelligence .

SMA Slime Mould Algorithm .

TS Tabu Search .

UHF Ultra High Frecuency.

VNS Variable Neighborhood Search .

Bibliografía

- [1] C. A. Balanis, *Antenna theory: analysis and design*. John Wiley & Sons, 2015.
- [2] J. B. Benavides, R. A. Lituma, P. A. Chasi, and L. F. Guerrero, "A novel modified hexagonal shaped fractal antenna with multi band notch characteristics for uwb applications," in *2018 IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications (APWC)*. IEEE, 2018, pp. 830–833.
- [3] P. M. Guaman, L. F. Guerrero-Vasquez, J. P. Bermeo, and P. A. Chasi, "Metaheuristic optimization algorithms of swarm intelligence in patch antenna design," in *2018 IEEE 10th Latin-American Conference on Communications (LATINCOM)*. IEEE, 2018, pp. 1–6.
- [4] T. A. Jara-Quito, L. F. Guerrero-Vásquez, J. O. Ordonez-Ordonez, P. A. Chasi-Pesantez, L. A. Morocho-Maita, and L. B. Peralta, "Design of a flame fractal patch antenna for uwb applications," in *2020 IEEE ANDESCON*. IEEE, 2020, pp. 1–5.
- [5] T. A. Jara-Quito, H. Duy-Romero, L. F. Guerrero-Vásquez, P. A. Chasi-Pesantez, and J. O. Ordoñez-Ordoñez, "Development of an augmented reality application as support in antenna characterization," in *Proceedings of Sixth International Congress on Information and Communication Technology*. Springer, 2022, pp. 131–140.
- [6] L. M. Martínez Gómez *et al.*, "Optimización de un controlador mpc mediante algoritmos metaheurísticos," 2021.
- [7] F. Herrera, "Introducción a los algoritmos metaheurísticos," 2017.
- [8] H. Y. Chong, H. J. Yap, S. C. Tan, K. S. Yap, and S. Y. Wong, "Advances of metaheuristic algorithms in training neural networks for industrial applications," *Soft Computing*, vol. 25, no. 16, pp. 11 209–11 233, 2021.

- [9] J. C. Bansal, P. K. Singh, and N. R. Pal, *Evolutionary and swarm intelligence algorithms*. Springer, 2019, vol. 779.
- [10] A. Chakraborty and A. K. Kar, "Swarm intelligence: A review of algorithms," *Nature-inspired computing and optimization*, pp. 475–494, 2017.
- [11] L. Brezočnik, I. Fister Jr, and V. Podgorelec, "Swarm intelligence algorithms for feature selection: a review," *Applied Sciences*, vol. 8, no. 9, p. 1521, 2018.
- [12] M. M. Millonas, "Swarms, phase transitions, and collective intelligence," *arXiv preprint adap-org/9306002*, 1993.
- [13] M. A. Muñoz, J. A. López, and E. F. Caicedo, "Inteligencia de enjambres: sociedades para la solución de problemas (una revisión)," *Ingeniería e Investigación*, vol. 28, no. 2, pp. 119–130, 2008.
- [14] C. Darwin, "On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life," *London: John Murray*, 1859.
- [15] A. N. Sloss and S. Gustafson, "2019 evolutionary algorithms review," *Genetic programming Theory and practice XVII*, pp. 307–344, 2020.
- [16] M. Janga Reddy and D. Nagesh Kumar, "Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review," *H2Open Journal*, vol. 3, no. 1, pp. 135–188, 2020.
- [17] A. Slowik and H. Kwasnicka, "Evolutionary algorithms and their applications to engineering problems," *Neural Computing and Applications*, vol. 32, pp. 12 363–12 379, 2020.
- [18] T. Bartz-Beielstein, J. Branke, J. Mehnen, and O. Mersmann, "Evolutionary algorithms," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 3, pp. 178–195, 2014.
- [19] D. Corne and M. A. Lones, "Evolutionary algorithms," in *Handbook of Heuristics*. Springer, 2018, pp. 409–430.
- [20] N. Siddique and H. Adeli, *Nature-Inspired Computing: Physics-and Chemistry-Based Algorithms*. Chapman and Hall/CRC, 2017.

- [21] A. Naik and S. C. Satapathy, "Past present future: a new human-based algorithm for stochastic optimization," *Soft Computing*, vol. 25, pp. 12 915–12 976, 2021.
- [22] R. C. Pérez and N. G. González, "Resolución y análisis de problemas de búsqueda de caminos mínimos mediante algoritmos metaheurísticos," *TRILOGÍA*, p. 128.
- [23] J. R. Artunduaga, *Antenas: conceptos básicos simplificados de funcionamiento y diseño*. Bonaventuriana, 2007.
- [24] S. V. Ati Sarango, "Implementación de una antena patch reconfigurable en frecuencia para la operación en dos bandas lte," B.S. thesis, Quito, 2020., 2020.