

A tool for the design of broadband MSMN networks on transmission lines based on the comparison of optimization heuristic algorithms

Abraham Hidalgo ^a , Fabián Robledo ^{*,b} 

^aEscuela de Ingeniería de Telecomunicaciones, Facultad de Ingeniería, Universidad de Carabobo. Valencia, Venezuela.

^bDepartamento de Electrónica y Comunicaciones, Escuela de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Carabobo. Valencia, Venezuela.



<https://doi.org/10.54139/revinguc.v29i2.171>

Abstract.- We present the development and results of a Python-programmed tool for the design of broadband multiple stub impedance matching networks (MSMN) in transmission line systems, employing a suite of heuristic optimization algorithms including: Nelder-Mead, Differential Evolution, Dual Annealing, Brute Force, Whale Optimization Algorithm (WOA), Equilibrium Optimizer (EO), Harris Hawks Optimization (HHO), Virus Colony Search (VCS) and Artificial Ecosystem-Based Optimization (AEO); offering comparable solutions that seek to satisfy a given matching specification, taking as the work function the magnitude of the voltage reflection coefficient, in a given bandwidth, just at the input plane to the network, connected to a line loaded with a modelable arbitrary impedance. The tool uses the indicated algorithms to determine the positions and lengths of the stubs of the MSMN network. The algorithms were obtained from the Mealpy and SciPy libraries. Figures of merit were designed that allowed comparing the quality of the solutions. The results were compared with those of the Keysight ADS simulator, and 3 test examples were executed using the algorithms, yielding viable adaptation results, which are discussed.

Keywords: Impedance matching; Heuristic algorithms; Transmission lines; MSMN; Optimization; Stubs.

Una herramienta para el diseño de redes MSMN de banda ancha en líneas de transmisión basada en algoritmos heurísticos de optimización comparados

Resumen.- Se presenta el desarrollo y resultados de una herramienta programada en Python para el diseño de redes de adaptación de impedancia de banda ancha y múltiples stubs (MSMN) en sistemas de líneas de transmisión, que emplea una suite de algoritmos heurísticos de optimización, incluyendo: Nelder-Mead, evolución diferencial, recocido dual, fuerza bruta, optimización de ballenas (WOA), optimizador de equilibrio (EO), optimización de halcones Harris (HHO), búsqueda de colonias de virus (VCS) y optimización basada en ecosistemas artificiales (AEO); ofreciendo soluciones comparables que procuran satisfacer una especificación dada de adaptación, tomando como función de trabajo la magnitud del coeficiente de reflexión de voltaje, en un ancho de banda determinado, justo en el plano de entrada a la red, conectada a una línea cargada con una impedancia arbitraria modelable. La herramienta emplea los algoritmos señalados para determinar las posiciones y longitudes de los stubs de la red MSMN. Los algoritmos se obtuvieron de las librerías Mealpy y SciPy. Se diseñaron figuras de mérito que permitieron comparar la calidad de las soluciones. Se contrastaron los resultados con los del simulador Keysight ADS y se ejecutaron 3 ejemplos de prueba empleando los algoritmos, proporcionando resultados viables de adaptación, los cuales se discuten.

Palabras clave: Adaptación de impedancia; Algoritmos heurísticos; Líneas de transmisión; MSMN; Optimización; Stubs.

Received: 11 de julio, 2022.

Accepted: 15 de agosto, 2022.

1. Introducción

En los sistemas de líneas de transmisión, formados por generador, línea y carga (por ejemplo, una antena), la adaptación de impedancias es una operación difícil cuando se abarca una banda ancha de frecuencias. Generalmente, la impedancia

* Autor para correspondencia:

Correo-e: frobledo@uc.edu.ve (F. Robledo)

característica de las líneas de transmisión y la impedancia de entrada de los sistemas de antena pueden incluir componentes reactivas, lo que hace que varíen en la banda de frecuencias de operación, produciendo desadaptación y dando lugar a ondas reflejadas indeseadas hacia el generador [1]. Se requiere que el sistema de transmisión esté adaptado en impedancia para maximizar la potencia hacia la carga, de modo que adaptar una línea de transmisión consiste en hacer desaparecer de ella las ondas reflejadas, al conseguir que la impedancia vista desde el plano de adaptación sea igual a la impedancia característica de la línea de transmisión [2]. Para lograr esto una alternativa es colocar, entre la línea de transmisión de impedancia característica Z_c y la carga Z_L , una red de adaptación que emplee N brazos de reactancia o *stubs*, conocida como MSMN (Multiple Stub Matching Network), de la cual un ejemplo se muestra en el sistema de la Figura 1 para stubs en paralelo cargados en cortocircuito. Idealmente, en el diseño se persigue el objetivo de que la impedancia de entrada Z_{in} vista en el plano $d = d_A$ hacia la derecha sea igual a la Z_c de la línea.

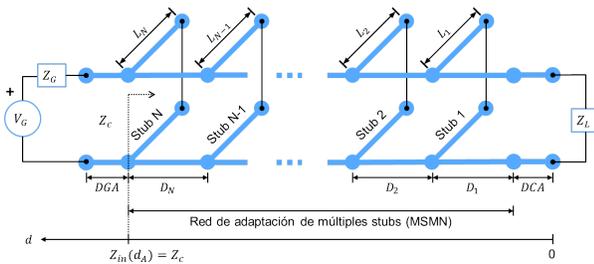


Figura 1: Red de adaptación MSMN de N stubs en paralelo (en corto circuito), ubicada entre el generador y la carga

Un stub es un segmento de línea de transmisión sin pérdidas (o con muy bajas pérdidas), cargado en corto circuito o en circuito abierto, conectado en paralelo o serie en una línea principal, que presenta cierta reactancia dominante en su puerto de entrada, dependiendo de su longitud y de la frecuencia. Diseñar la red MSMN consiste en determinar todas las longitudes $L_i (i = 1, \dots, N)$ y posiciones D_i de los N stubs que posee, para lograr

cierto grado de adaptación a partir de la entrada a la red. El problema de desadaptación ha sido tratado analíticamente en sistemas de banda estrecha y hasta 2 stubs por diversos autores [1, 2, 3, 4, 5]. Sin embargo, estos se enfocan en soluciones analíticas exactas para adaptar el sistema a una frecuencia de operación, y condiciona el funcionamiento óptimo, pues el ancho de banda puede llegar a ser insuficiente según se incrementa la tasa de transmisión en los sistemas digitales, llegando hasta 20 Gbps en redes 5G [6], así como también en sistemas de espectro disperso, o cuando se requiere la multiplexión FDM de señales de TV en plantas que reutilizan la antena para varios canales UHF, en sitios de transmisión y torres comunitarias.

Considerar la adaptación de sistemas de líneas de transmisión en banda ancha con redes MSMN ofrece oportunidades para el desarrollo de nuevas metodologías [7, 8, 9]. No obstante, determinar las posiciones y longitudes exactas de los N stubs de la red MSMN en diseños de banda ancha representa una tarea complicada para ser considerada en la práctica, haciendo que el recurso computacional sea indispensable a los fines de obtener soluciones numéricas, y donde el empleo de algoritmos adecuados en la búsqueda de soluciones al problema no lineal es pertinente, siendo ahora posible aplicar Algoritmos Heurísticos de Optimización (AHO), cuya viabilidad se plantea en este estudio.

Para ello se desarrolló una herramienta de software que, dado un sistema de línea de transmisión desadaptado, diseñe la red MSMN con stubs en paralelo requerida para lograr un grado determinado de adaptación en banda ancha, en torno a una frecuencia central. Con este fin se emplearon varios AHO en base a sus ventajas [10, 11], candidatos para tratar de determinar soluciones que indiquen las posiciones y longitudes de los N stubs de la red MSMN. Se emplearon varios AHO de naturaleza lógica distinta, debido a que la no linealidad inherente al problema que se resuelve implica que no necesariamente todos estos podrán arrojar una solución viable en tiempos de cómputo razonables, siendo deseable disponer de alternativas ante un problema que puede tener múltiples soluciones, algunas de ellas subóptimas

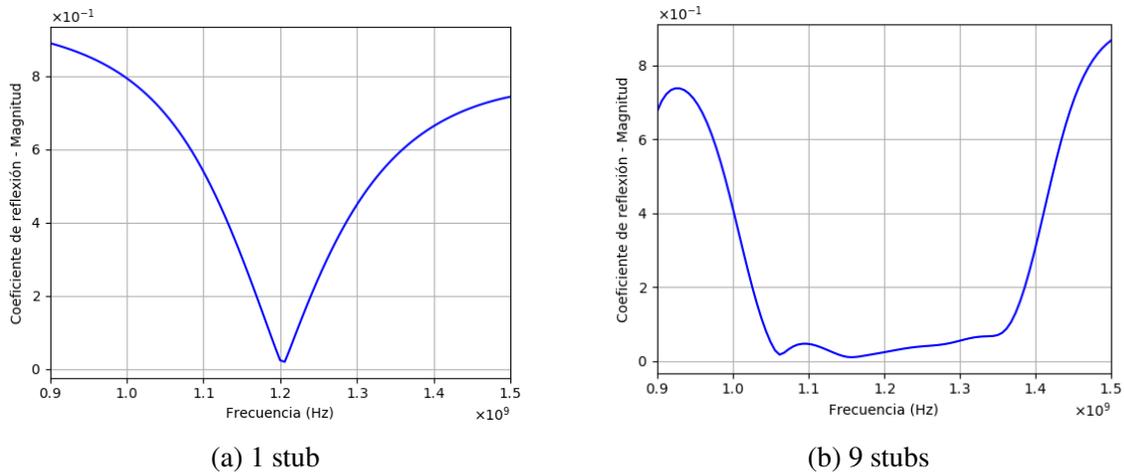


Figura 3: Adaptación en un sistema ejemplo de banda L con una red MSMN, empleando la herramienta de software *Eureka MSMN*

2. Metodología y desarrollo de la herramienta de software *Eureka MSMN*

Los AHO son algoritmos para resolver problemas que se modelan como búsquedas u optimización de funciones, difíciles de abordar en un tiempo razonable por otros algoritmos exhaustivos. La ventaja de los AHO es que son eficientes para obtener mínimos o máximos de alguna función que tenga una gran cantidad de variables independientes [10], como es el caso en estudio. La función generalmente se llama “función objetivo”, y representa una medida cuantitativa de la “bondad” de la configuración de un sistema determinado como solución de un problema. Dicha función depende de la configuración del sistema, que incluye parámetros y topología. Los AHO recorren el espacio de soluciones e intentan encontrar la combinación de variables independientes que corresponda a la mejor solución posible (configuración) para extremalizar la función objetivo [11]. Estos métodos toman como punto de partida una solución inicial dada, que intentarán mejorar aplicándole ciertos operadores, para calcular otras soluciones similares (vecinas) por las cuales se procede a continuar la búsqueda. Así se supone que, modificando una solución al problema, se podrían calcular soluciones mejores, iterando.

Estos AHO se aplican cuando el tamaño del espacio de soluciones es muy grande. Los AHO

se encargan de “podar” el espacio de soluciones, es decir, ayudan a eliminar caminos que llevan a soluciones de peor calidad que la solución actual. El inconveniente es que se pueden descartar caminos que conduzcan a soluciones “buenas”, o que se encuentren máximos o mínimos locales. Los 9 AHO empleados en *Eureka MSMN* se obtuvieron de la librería especializada de nombre *Meta-heuristics Algorithms in Python*, Mealpy [12] y de la librería estándar SciPy de Python, ambas de licencia libre; y poseen los siguientes nombres: Nelder-Mead [13], evolución diferencial [14], recocido dual [15], fuerza bruta, optimización de ballenas [16], optimizador de equilibrio [17], optimización de halcones Harris [18], búsqueda de colonias de virus [19], y optimización basada en ecosistemas artificiales [20]. En la Tabla 1 se muestra un resumen de las características de los AHO utilizados.

En este estudio, la función objetivo que se pretende minimizar tiene $2N$ grados de libertad y se denota como $F_{obj}(D_i, L_i) \equiv F_{obj}(D_1, D_2, \dots, D_N, L_1, L_2, \dots, L_N)$, definiéndose como una acumulación espectral no negativa de errores cuadráticos $E(f_n, D_i, L_i)$ en el ABA de interés, según la ecuación (1):

$$F_{obj}(D_i, L_i) = \sum_{n=n_l}^{n_f} E(f_n, D_i, L_i) \quad (1)$$

Donde D_i y L_i son las distancias de ubicación

Tabla 1: AHO empleados en la herramienta *Eureka MSMN*

Nombre	Abrev.	Diseñador	Enfoque	Aplicación	Ventajas	Desventajas
Nelder-Mead	-	J. Nelder y R. Mead.	Búsqueda de patrones.	Optimización de funciones unimodales.	Aplicable a funciones discontinuas.	Propenso a estancarse en óptimos locales.
Evolución diferencial	DE	R. Storn y K. Price.	Basado en población.	Diseño de filtros digitales.	Requiere pocas variables de control.	Dificultad con conjuntos no linealmente separables.
Recocido dual	-	Xiang Y, Sun DY, Fan W y Gong XG.	Estocástico.	Problemas similares al “agente viajero”.	Mayor chance de obtener el óptimo global.	Menor precisión.
Fuerza bruta	-	-	Exhaustivo.	Funciones de pocas variables.	Logra el valor óptimo global.	Alto consumo de tiempo.
Optimización de ballenas	WOA	S. Mirjalili y A. Lewis.	Inteligencia de enjambre.	Reconocimiento de patrones.	Rapidez.	Variables de control requeridas.
Optimizador de equilibrio	EO	A. Faramarzi, M. Heidarinejad, B. Stephens y S. Mirjalili.	Flujo de masa.	Problema de despacho económico de carga.	Eficiente al hallar el óptimo global.	La versión original no es multiobjetivo.
Optimización de halcones Harris	HHO	A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja y H. Chen.	Basado en cooperar.	Problemas de optimización con restricciones matemáticas.	Evita valores óptimos locales.	La versión original no es multiobjetivo
Búsqueda de colonias de virus	VCS	Mu Dong Li, Hui Zhao, Xing Wei Weng y Tong Han.	Difusión e infección.	Asignación de recursos de computación en la nube.	Mayor eficiencia al explorar.	Alto consumo de tiempo.
Optimización basada en ecosistemas artificiales	AEO	W. Zhao, L. Wang y Z. Zhang.	Flujo de energía.	Identificación de parámetros hidrogeológicos.	Mayor eficiencia de cómputo.	La versión original no es multiobjetivo.

de cada uno los N stubs con respecto a la carga, y las longitudes de estos, respectivamente; f_n es la frecuencia discreta dentro del ABA en consideración, que se barre desde una frecuencia inicial $f(n_I)$ hasta una frecuencia final $f(n_F)$, siendo n_I y n_F números enteros que indexan los límites de la discretización del rango de frecuencias del ABA = $f(n_F) - f(n_I)$, con suficiente resolución.

Los términos de error $E(f_n, D_i, L_i)$ que se suman en (1) para cada frecuencia f_n están dados por el error cuadrático entre el valor de $|\Gamma|$ a la frecuencia f_n y $|\Gamma|_{Máx}$, en el plano de adaptación $d = d_A$ donde se ubica la red MSNM (Figura 1), siendo $|\Gamma|_{Máx}$ el máximo valor permitido para $|\Gamma|$ a la entrada de la red. Esto se define condicionalmente en la ecuación (2):

$$E(f_n, D_i, L_i) = \begin{cases} 0 & , |\Gamma(f_n, D_i, L_i)| \leq |\Gamma|_{Máx} \\ [|\Gamma(f_n, D_i, L_i)| - |\Gamma|_{Máx}]^2 & , |\Gamma(f_n, D_i, L_i)| > |\Gamma|_{Máx} \end{cases} \quad (2)$$

Para una configuración dada de stubs, la situación ideal $F_{obj}(D_i, L_i) = 0$ se obtendría en el caso de que en todo el ABA la solución satisface $|\Gamma(f_n, D_i, L_i)| \leq |\Gamma|_{Máx}$. Si esto no se logra en todo el ABA, entonces el error cuadrático de (2) se toma en cuenta para esa configuración, acumulándose según (1), para dar la función objetivo $F_{obj}(D_i, L_i)$

asociada a esa configuración particular, la cual se compara con la obtenida por otras configuraciones por parte del AHO, para tratar de encontrar una configuración que se aproxime a una solución óptima, al menos localmente.

El lenguaje de programación fue Python v3.7, debido a su carácter OO, a ser de código abierto

y a disponerse de librerías para su reutilización al implementar los AHO. La librería para desarrollo de interfaz gráfica de usuario (IGU) fue *PyQt*, con su herramienta *Qt Designer*. En relación a las funciones matemáticas se utilizaron las librerías estándar *math* y *cmath* de *Python*. También se usaron las librerías *Numpy* y *SciPy* para el manejo de arreglos y la interpolación. Para graficar se empleó *Matplotlib* de *Python*, así como también el módulo *plotting.smith* de la librería *Scikit-rl* con el objetivo de generar cartas de *Smith*.

Los AHO se incorporaron utilizando la librería *Mealpy* y el módulo *optimize* de la librería *SciPy*. El módulo *optimize* aportó funciones para extremalizar funciones matemáticas sujetas a restricciones, incluyendo solucionadores para problemas no lineales, soportando algoritmos de optimización local y global. De este módulo *optimize* se emplearon 4 AHO, siendo estos: *Nelder-Mead*, *evolución diferencial*, *recocido dual* y *fuerza bruta*. Por otra parte, de la librería *Mealpy*, que contiene 176 AHO adaptados por *Van Thieu* [12] se emplearon 5 AHO, siendo estos: *Algoritmo de optimización de ballenas*, *optimizador de equilibrio*, *optimización de halcones Harris*, *búsqueda de colonia de virus* y *optimización basada en ecosistemas artificiales*. Los 9 algoritmos mencionados fueron el resultado de las pruebas preliminares, que justificaron su selección como candidatos para el problema de adaptación con la red *MSMN*. También se usó el módulo *threading* multihilo de *Python*.

Se diseñó la herramienta de modo que se pudiera elegir entre dos modos de operación, llamados “Análisis de *MSMN*” y “Diseño de *MSMN*”. En el Modo Análisis el usuario selecciona los datos de cada elemento del sistema: *Generador*, *línea de transmisión*, *configuración particular de la red MSMN* (*Número*, *posiciones y longitudes de los stubs*) e *impedancia de carga* (*resistencia*, *reactancia y modelo circuital*). El software calcula

y entrega las gráficas de diversas magnitudes y las figuras de mérito sobre el desempeño del sistema (Figura 4a). Por ser un modo de análisis no se emplea ningún AHO.

En el Modo Diseño el usuario desea obtener los datos de posiciones y longitudes de los stubs que definen a la *MSMN* (los D_i y L_i), obtenidos por los AHO seleccionados de los 9 ofrecidos (Figura 4b). El usuario debe proporcionar los requerimientos de diseño de la red: f_0 , ABA y $|\Gamma|_{Máx}$, pudiendo agregar un límite para N , y los mínimos admisibles para la longitud y la distancia de los stubs, viables mecánicamente. Esta información se procesa generando arreglos de datos con los cuales se obtienen las magnitudes relevantes, como son la impedancia “vista” en un plano arbitrario a la distancia d , llamada $Z_v(d)$, y $|\Gamma(f)|$. A partir de una lista F de frecuencias discretas se calculan, en todo el rango de frecuencia, la impedancia de carga, la impedancia característica y la constante compleja de propagación de la línea. Esto se obtuvo evaluando las ecuaciones algebraicas de sistemas distribuidos (caso armónico en estado estacionario), en cada punto frecuencial de la lista F . También se creó otra lista denominada D , con los valores discretos de distancia del eje espacial d desde la carga (Figura 1). La separación entre valores de la lista D fue de 1 mm. Los parámetros por unidad de longitud son los usuales r, l, c y g , y el parámetro de propagación γ está dado por la ecuación (3):

$$\gamma = \alpha + j\beta = \sqrt{(r + j\omega l)(g + j\omega c)} \quad (3)$$

Donde α es la constante de atenuación y β es la constante de fase, ambas dependientes de la frecuencia. La impedancia “vista” a la distancia d_i (hacia la derecha), denotada como $Z_v(d_i)$, estará dada por la ecuación (4), que emplea el algebra de los sistemas de parámetros distribuidos en líneas con pérdidas:

$$Z_v(d_i) = Z_c \left\{ \frac{Z_v(d_{i-1}) + Z_c \tanh[(\alpha + j\beta)(d_i - d_{i-1})]}{(Z_c + Z_v(d_{i-1}) \tanh[(\alpha + j\beta)(d_i - d_{i-1})])} \right\} \quad (4)$$

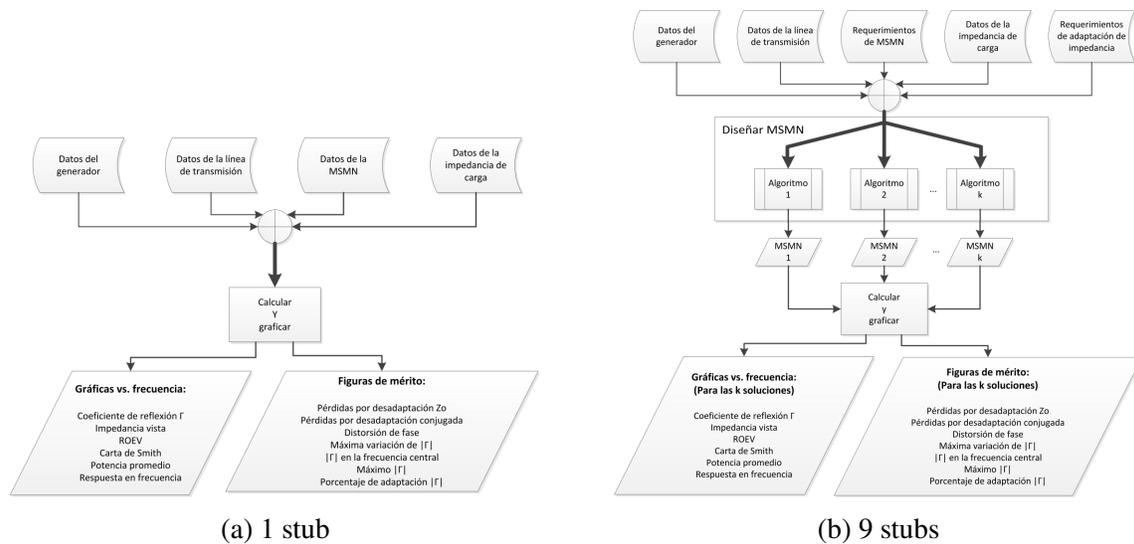


Figura 4: Modos de la herramienta de software *Eureka MSMN*

Y el coeficiente de reflexión $\Gamma(d_i)$ está dado por la ecuación (5):

$$\Gamma(d_i) = \frac{Z_v(d_i) - Z_c}{Z_v(d_i) + Z_c} \quad (5)$$

Aclarando que en la ecuación (4) se sigue la inclusión progresiva de las inserciones paralelo de cada uno de los stubs al desplazarse de derecha a izquierda, desde la carga al generador. Este proceso se realiza para cada punto de frecuencia de la lista F , y a su vez en cada entrada de la lista de puntos espaciales D . De esta manera, la impedancia de entrada vista $Z_v(d_i)$ en un plano interno arbitrario de la red MSMN resulta ser un arreglo bidimensional (2D), en el cual para acceder a un valor de impedancia se deben especificar los índices de distancia y de frecuencia. El procedimiento indicado discreto permite obtener los valores de $Z_v(d_i)$ en los puntos a lo largo de la línea principal, y dentro del ABA. La impedancia de entrada a la red MSMN, en el plano $d = d_A$ estará dada por $Z_{in} = Z_v(d_A)$.

Para ese arreglo 2D se calculan las magnitudes de interés en cada punto espacial y de frecuencia, como Γ y la relación de onda estacionaria de voltaje ROEV. Se implementó la opción de graficar las magnitudes usando la carta de Smith en cada punto a lo largo de la línea, y para el rango del ABA. Se obtuvo también la potencia activa promedio empleando $Z_v(d_i)$ y el voltaje

correspondiente obtenido bajo las condiciones de borde preestablecidas. También se determinó la respuesta en frecuencia $H(\omega)$ desde el plano del generador al plano de la carga, al considerar al filtro como una secuencia de k celdas en cascada, cada una con respuesta en frecuencia $H_k(\omega)$.

En relación a la IGU (Interfaz gráfica de usuario), se diseñó de forma intuitiva y gráfica, con control de tipo, de errores y ayuda; para la introducción por el usuario de los parámetros, ofreciendo gráficos y figuras de mérito sobre su desempeño. En la pantalla principal de la IGU se permite al usuario seleccionar el modelo de la carga, del generador y de la línea, y los parámetros de esta. Para recibir la información del usuario se usaron elementos de la librería *PyQt* incluyendo *QLineEdit* y el uso de botones, que permiten guardar y cargar las configuraciones y el proyecto completo en sus modos Diseño y Análisis. La IGU muestra al usuario el proceso de búsqueda de los AHO, informando cada vez que un AHO inicia o finaliza, indicando además el tiempo transcurrido, suministrando el desempeño del sistema mediante gráficas bidimensionales, a partir de los resultados de los módulos de cómputo versus la frecuencia, y con la posibilidad de variar el punto espacial de vista a lo largo de la línea principal (eje d). En cuanto al módulo graficador, una vez ingresados los datos, se valida su formato y se procede a iniciar los módulos de cómputo, que arrojan los arreglos

de datos de los resultados, para ser graficados. Las librerías Matplotlib y mplcursors de Python facilitan la visualización de tales gráficas.

Una validación de los resultados de los módulos de cómputo se llevó a cabo contra un simulador analizador de líneas de transmisión, el Advanced Design System (ADS) del producto PathWave Design de Keysight. Se compararon los resultados arrojados por los cálculos del simulador ADS con los obtenidos por *Eureka MSMN* en el Modo Análisis (sección 3.4).

En referencia a la implementación del Módulo Diseño, el problema se basó en emplear los AHO para buscar la adaptación de impedancias. Se le proporcionaron los datos de entrada a estos módulos de cómputo, en el formato adecuado para cada uno de los AHO señalados en la Tabla 1, y estos se emplearon en la tarea de minimizar la función objetivo F_{obj} , dada en la ecuación (1). Cada uno de los AHO entrega como respuesta una lista con los valores de las posiciones D_i y longitudes L_i de los stubs de la red MSMN que corresponden a un valor mínimo de F_{obj} y se llevan estos resultados a la entrada de datos del módulo graficador de la herramienta (Figura 4b).

En este punto se tiene el sistema de línea de transmisión definido mediante sus parámetros, junto con la información de las cantidades relevantes dadas por los módulos de cómputo (para todo punto sobre el eje d y para el rango del *ABA*). Se realizan entonces los cálculos del desempeño de la redes MSMN obtenidas (una por cada AHO). Para ello se emplean las usuales figuras de mérito, como son las pérdidas por desadaptación conjugada, y pérdidas por desadaptación Z_0 [2], y adicionalmente se diseñaron dos figuras de mérito específicas para el estudio, denominadas PA y DF .

En cuanto a la figura de mérito PA (Porcentaje de Adaptación), esta se define por medio de la ecuación (6), que indica en un solo número el grado de adaptación de impedancia obtenido con la red MSMN diseñada, considerando todo el rango de frecuencia del *ABA*, y no solamente la adaptación a una frecuencia particular, tomando en cuenta los M valores absolutos de la lista discretizada $|\Gamma(f)|$ sumados en el *ABA*, en forma discreta, de 0 a $M-1$,

que en su versión discreta se refieren como $|\Gamma_n|$:

$$PA = 100 \left(1 - \frac{1}{M} \sum_{n=0}^{M-1} |\Gamma_n| \right) (\%) \quad (6)$$

Así, un valor $PA = 100\%$ sería el caso ideal de adaptación donde $\Gamma(f) = 0$ para todas las frecuencias en la versión discreta del *ABA*. El valor de PA indica el grado de adaptación global logrado por una MSMN. Esto permite comparar la calidad de los resultados para el diseño de la red MSMN debidos a los distintos AHO, pues cada uno podría tener un PA diferente, siendo de interés el mayor.

En cuanto a la figura de mérito DF (Distorsión de fase), esta requiere el cálculo previo del retardo de grupo $\tau(\omega)$, obtenido a partir de la versión discretizada del espectro de fase $\theta(\omega)$ de la respuesta en frecuencia $H(\omega)$ ya mencionada, donde $\omega = 2\pi f$, es definido según la ecuación (7):

$$\tau(\omega) = -\frac{d\theta}{d\omega} \quad (7)$$

Si no existe distorsión de fase, $\tau(\omega)$ sería una lista de M valores iguales. En general, $\tau(\omega)$ es una lista con M valores posiblemente distintos. Se realiza entonces un ajuste de curva de grado cero a $\tau(\omega)$ mediante la función polyfit de Numpy, hallando un valor constante T que representa al conjunto de valores $\tau(\omega)$ como un número que minimiza el error cuadrático, y que coincidiría con el valor del retardo de grupo para cualquier frecuencia si $\theta(\omega)$ fuese lineal. La DF cuantifica la desviación con respecto a la fase lineal, y se define como un error relativo porcentual promedio calculado a partir del retardo de grupo $\tau(\omega)$ de la ecuación (7) en su versión discreta τ_n , según la ecuación (8).

$$DF = \frac{100}{M} \sum_{n=0}^{M-1} \left| \frac{\tau_n - T}{T} \right| (\%) \quad (8)$$

Una vez obtenidos los resultados se realiza a efectos de validación externa de la herramienta el cálculo de una figura de mérito adicional, llamada $PCRA$ (Porcentaje de Cumplimiento del Requerimiento de Adaptación), que cuantifica la proporción lineal del *ABA* para la cual se cumple

el requerimiento de adaptación, y que se define mediante la ecuación (9):

$$PCRA = \frac{100}{M} \sum_{n=0}^{M-1} 1 - u[|\Gamma_n| - |\Gamma|_{M\acute{a}x}] \text{ (\%)} \quad (9)$$

Donde u representa la función escalón unitario discreta y M es el tamaño de la lista de frecuencia F . Es decir, $PCRA$ corresponde a la suma de los segmentos horizontales dentro ABA donde se verifica $|\Gamma| < |\Gamma|_{M\acute{a}x}$. Un valor $PCRA = 100\%$ indica que para una red MSMN dada, el valor de $|\Gamma|$ resultante es siempre menor a $|\Gamma|_{M\acute{a}x}$ en todo el ABA (Figura 2), y un valor de $PCRA = 0\%$ indica que $|\Gamma| \geq |\Gamma|_{M\acute{a}x}$ en todo el ABA . El $PCRA$ permite entonces evaluar si un AHO es capaz de hallar soluciones que cumplan con el requerimiento de adaptación y cuantificar en qué proporción lo hace en el ABA . Esto facilita comparar el desempeño y eficacia de los AHO.

Para la síntesis en un ejecutable, se compiló un archivo de extensión .exe, mediante la librería auto-py-to-exe. *Eureka MSMN* es un programa para su ejecución bajo el SO Microsoft Windows 7 o superior, de 32/64 bits. Requiere una RAM mínima de 2 GB (recomendado 4 GB) y un procesador igual o superior al Intel Celeron de 1,10 GHz.

3. Análisis y discusión de resultados

3.1. Inicialización

Al ejecutar el archivo *Eureka MSMN.exe* aparece el logo de la herramienta, como se observa en la Figura 5, el cual inicia la carga de las librerías mencionadas en la sección 2.



Figura 5: Pantalla de inicialización de la herramienta *Eureka MSMN*

3.2. Interfaz gráfica de usuario, IGU

3.2.1. Ventana principal

La ventana principal tiene dos partes (Figura 6), en la parte superior se encuentra un diagrama del sistema de línea de transmisión a configurar, para esto debe hacer clic en los botones Configurar generador, Configurar línea de transmisión, Configurar MSMN y Configurar impedancia de carga. Clicando cada uno de estos botones se abre una ventana de configuración. En la parte inferior se encuentra un recuadro llamado Requerimientos de adaptación de impedancia, donde se especifica el ABA , la f_0 , el $|\Gamma|_{M\acute{a}x}$ y se eligen los AHO. En la esquina superior izquierda hay un botón Proyecto, para guardar el proyecto actual o cargar uno existente. Finalmente, en la esquina inferior izquierda se encuentran dos botones, estos son: Iniciar búsqueda y Graficar, los cuales corresponden a los modos de operación “Análisis de MSMN” y “Diseño de MSMN” y respectivamente.

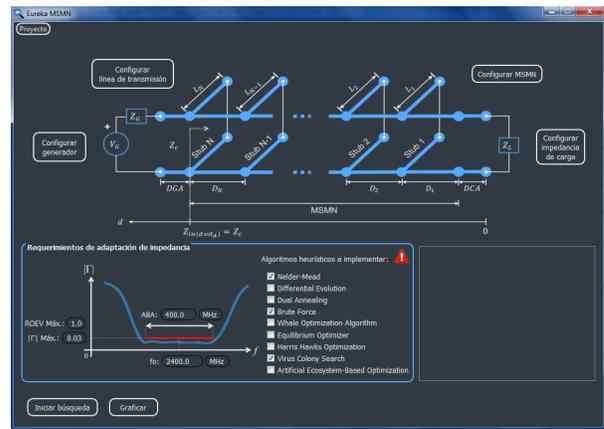


Figura 6: Ventana principal de *Eureka MSMN*

3.2.2. Ventana de configuración del generador

Clicando el botón Configurar generador de la ventana principal, se abre la ventana que se muestra en la Figura 7. Configurar el generador implica configurar la fuente y la impedancia del generador. La fuente se configura ingresando el voltaje o la potencia versus la frecuencia.

Clicando el spin box del recuadro de la fuente se puede elegir el número de valores de voltaje o potencia que se desea introducir. La impedancia

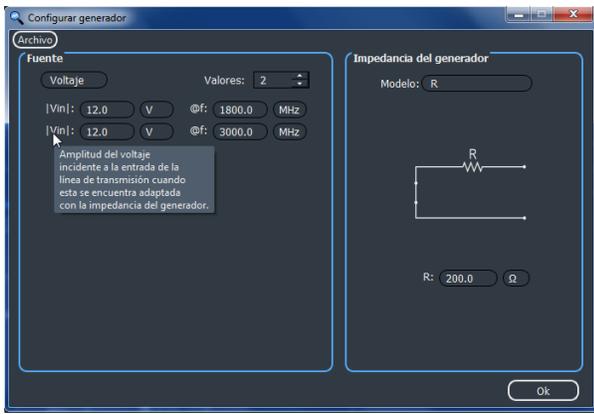


Figura 7: Ventana de configuración del generador

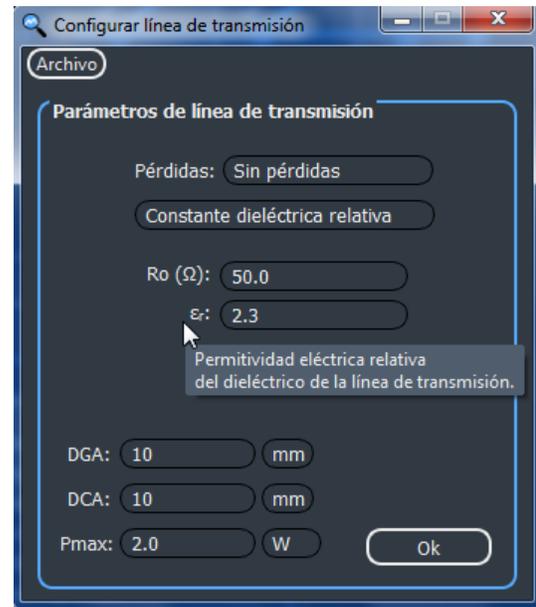
del generador se configura eligiendo el modelo circuital en un combo box que se encuentra en la parte superior del recuadro de impedancia del generador. Otra forma de ingresar la impedancia del generador es importando un archivo de extensión .imp. En la parte superior izquierda de la ventana se encuentra un botón llamado Archivo, con el cual se pueden cargar y guardar archivos de configuración de la fuente.

3.2.3. Ventana de configuración de la línea de transmisión

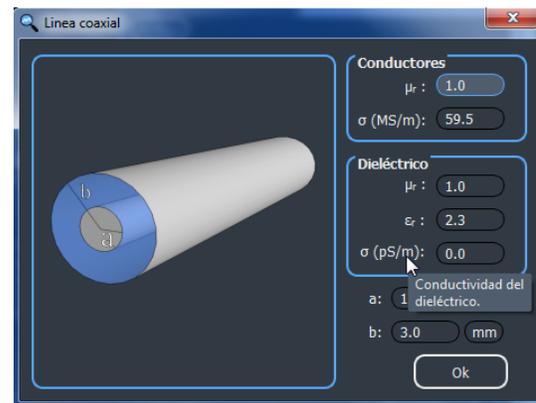
Clicando el botón Configurar línea de transmisión de la ventana principal, se abre la ventana mostrada en la Figura 8a. En ella se puede elegir el tipo de pérdidas de la línea (sin pérdidas o pérdidas generales). Si se elige el tipo pérdidas generales y luego el tipo de línea de transmisión, se ofrecen tres opciones de líneas: Coaxial, bifilar y plana. En la parte inferior de la ventana de configuración de la línea de transmisión (Figura 8a) se ingresa la distancia desde el generador hasta la red MSMN (*DGA* en la Figura 1), la distancia desde la carga hasta la red MSMN (*DCA* en la Figura 1) y la potencia máxima de la línea. En la parte superior izquierda de esta hay un botón Archivo, con el que se cargan y guardan archivos de configuración de la línea. Como ejemplo, la Figura 8b muestra la línea coaxial.

3.2.4. Ventana de configuración de la red MSMN

Al seleccionar Configurar MSMN en la Ventana principal, se abre la ventana de la Figura 9. En



(a) Ventana asociada a la línea de transmisión



(b) Ejemplo para el caso de una línea coaxial

Figura 8: Ventanas de configuración de la línea de transmisión

ella se puede especificar ya sea un número fijo N de stubs para los diseños de MSMN mediante los AHO, o bien un número máximo de stubs $N_{Máx}$. Además, se indican la longitud mínima y distancia mínima de un stub, y su carga (corto circuito o circuito abierto). En la parte superior izquierda de esta ventana se encuentra un botón Archivo, con el que se cargan y guardan archivos de configuración de la red.

3.2.5. Ventana de configuración de la impedancia de carga

Clicando el botón Configurar impedancia de carga de la Ventana principal, se abre la ventana

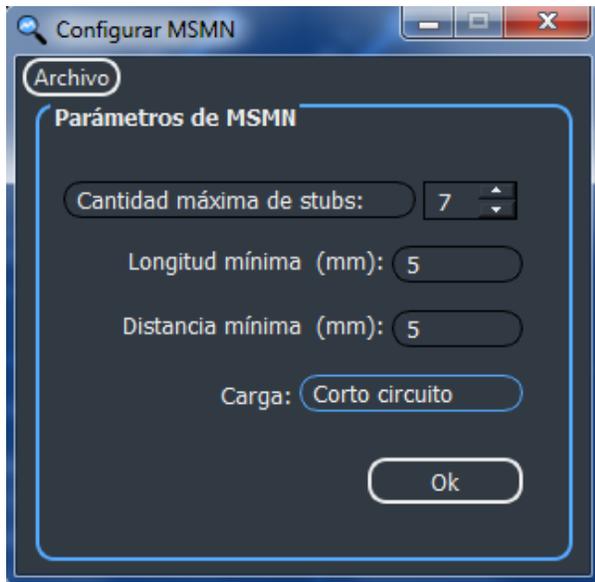


Figura 9: Ventana de configuración de la red MSMN

de la Figura 10, en ella se especifica el modelo circuital de la carga Z_L , así como sus valores de resistencia, inductancia y capacitancia, de acuerdo al modelo seleccionado. Otra forma de ingresar la impedancia de carga es importando un archivo de extensión .imp. En la parte superior izquierda se encuentra un botón Archivo, para cargar y guardar archivos de configuración.

3.3. Operación en el Modo Análisis de Eureka MSMN y Ejemplo 1 de aplicación

En esta sección se presenta mediante un ejemplo la operación en el Modo Análisis para una MSMN dada, denominando Ejemplo 1. En la Ventana principal se configura el sistema, y luego se hace clic en el botón Graficar. En la Figura 11 se ilustra la configuración del Ejemplo 1. En la Ventana principal, una vez que se ha configurado el generador, la línea de transmisión y la carga, clicando en Graficar se abre una ventana llamada Gráfica; en ella se configura la MSMN ingresando las longitudes y distancias de los stubs. En el Ejemplo 1, se utilizó un solo stub cargado en corto circuito (Figura 12).

En la ventana gráfica se especifica el rango de frecuencia para visualizar el espectro, que fue 800–1600 MHz en el Ejemplo 1. En la Figura 12 se aprecia la impedancia a la entrada

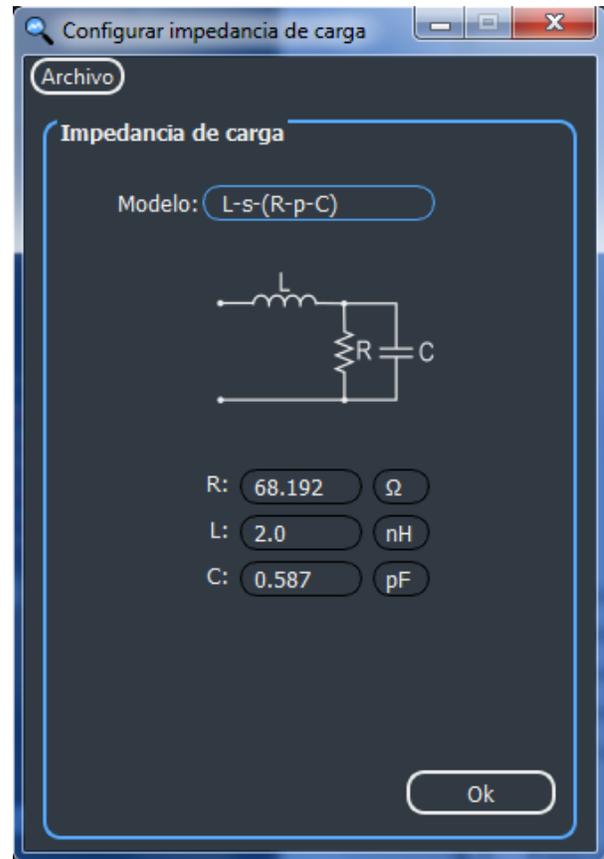


Figura 10: Ventana de configuración de la impedancia de carga: Modelo y parámetros

de la MSMN. En la parte superior izquierda de esta ventana, el usuario puede elegir el rango de frecuencia, el tipo de gráfica (coeficiente de reflexión, impedancia vista, ROEV, carta de Smith, potencia activa promedio y respuesta en frecuencia) y la presentación de esta (magnitud y fase, o parte real y parte imaginaria). En la parte inferior izquierda de la Figura 12 se aprecia la barra horizontal que permite variar el punto de vista sobre el eje d , y en la parte superior derecha se pueden cambiar los puntos de frecuencia de la gráfica y el número de celdas por cada longitud de onda. En el lado central derecho se encuentra un recuadro donde se puede configurar “manualmente” la red MSMN, también se pueden cargar/guardar configuraciones de redes. En la parte inferior derecha, el botón Ver figuras de mérito abre una ventana con las figuras de mérito resultantes para la configuración actual del sistema.

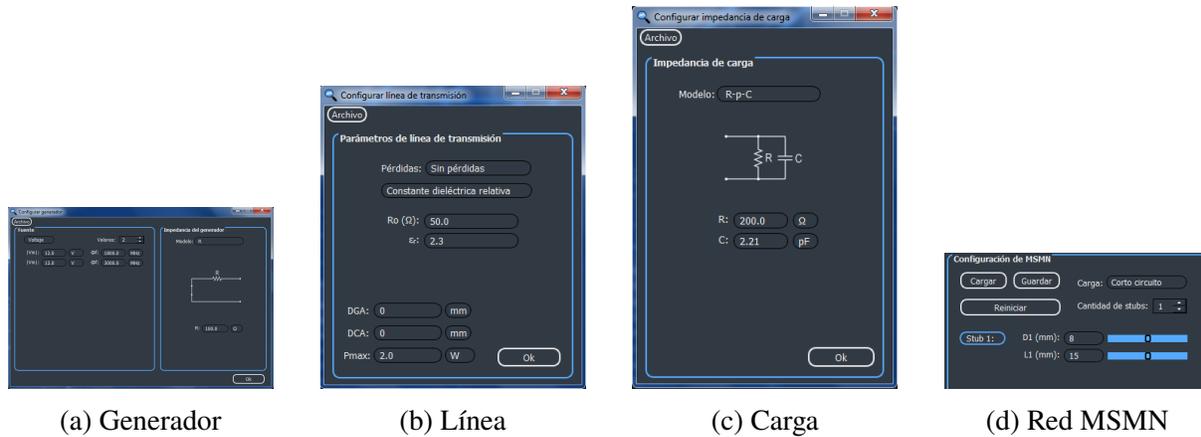


Figura 11: Configuración del Ejemplo 1 del Modo Análisis

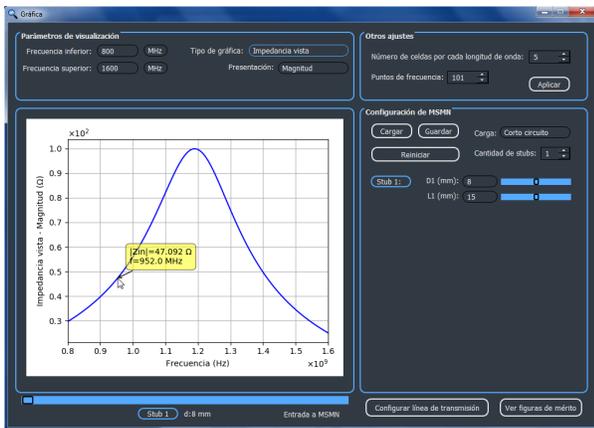


Figura 12: Ventana Gráfica. Impedancia (magnitud) vista a la entrada de la red MSMN del Ejemplo 1 en Modo Análisis

3.4. Comparación del Ejemplo 1 con el simulador Keysight ADS, en el Modo Análisis

Para validar los resultados de *Eureka MSMN* del Ejemplo 1 en el Modo Análisis dado en la sección 3.3, se realizaron pruebas comparando estos con los obtenidos mediante el simulador Advanced Design System (ADS) del software PathWave Design de Keysight. En la Figura 13 se muestra el diagrama esquemático con la misma configuración del Ejemplo 1, pero esta vez programado en el mencionado simulador comercial ADS.

La Figura 14 da los resultados de *Eureka MSMN* y ADS para el Ejemplo 1, resultando un error relativo del 0,006 % (amplitud) y de un 0,016 % (fase). Los mismos fueron calculados tomando muestras en 5 frecuencias, que incluyeron a f_0 y

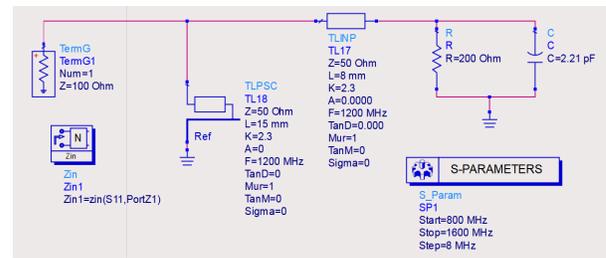


Figura 13: Programación del Ejemplo 1 de la sección 3.3 en ADS, para la comparación de resultados con *Eureka MSMN*

a los límites del *ABA*; errores que destacan por su reducido valor.

3.5. Operación en el Modo Diseño de Eureka MSMN y Ejemplo 2 de aplicación

En esta sección se presenta el resultado de operar en el Modo Diseño de la herramienta, por medio de un ejemplo multistub, que se denominará Ejemplo 2, configurándose el generador, la línea, la carga, y la red MSNM, con un número de stubs máximo $N_{m\acute{a}x}$ de 7 (Figura 15), y en la pantalla Requerimientos de adaptación de impedancia se definen para este Ejemplo 2 los parámetros $f_0 = 2.400$ MHz, $ABA = 400$ MHz y $|\Gamma|_{M\acute{a}x} = 0,03$, seleccionándose 3 de los AHO, siendo estos: Nelder Mead, Fuerza Bruta y Búsqueda por Colonia de Virus (Figura 16a).

Seguidamente, clicando el botón Iniciar búsqueda, (Figura 16a), un mensaje en la ventana principal le indica al usuario cuando cada AHO

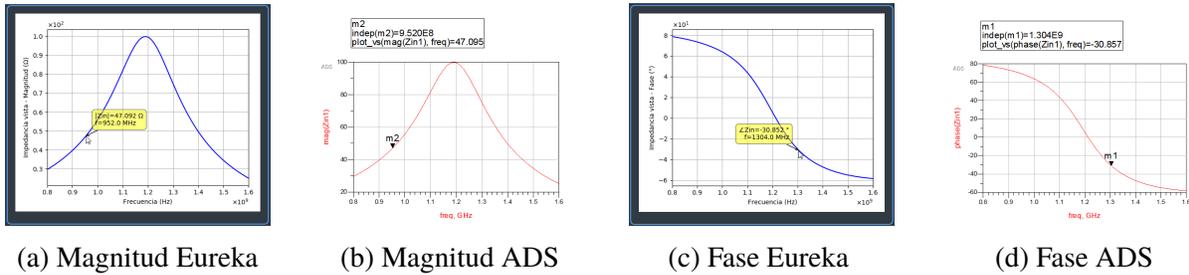


Figura 14: Validación de magnitud y fase de la impedancia vista a la entrada de la red MSMN del Ejemplo 1

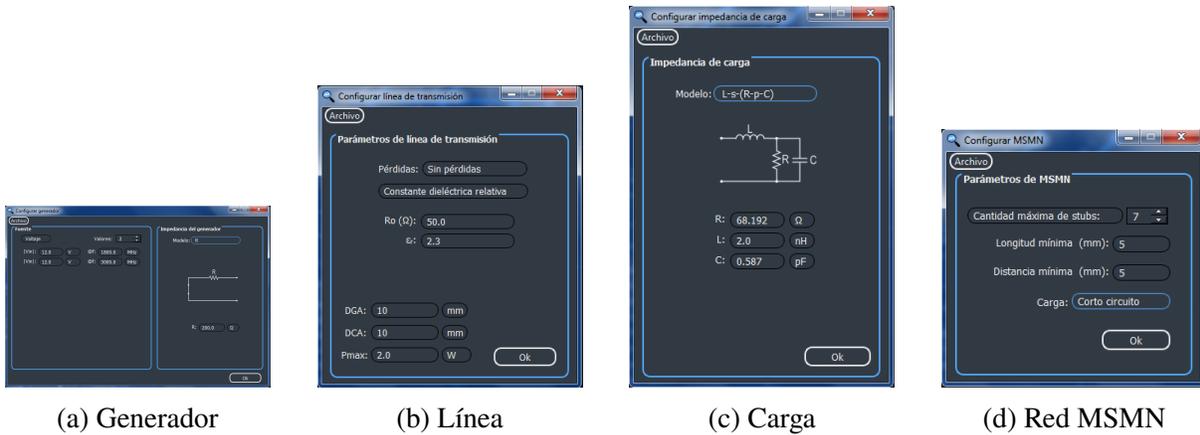


Figura 15: Configuración del Ejemplo 2 multistub del Modo Diseño

termina su ejecución, mostrando avances parciales (Figura 16b).

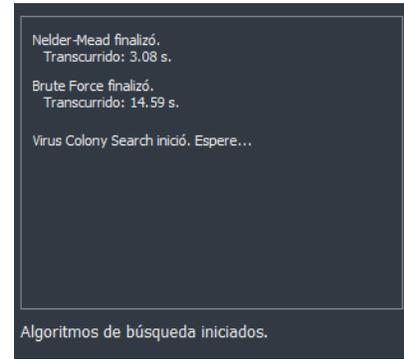
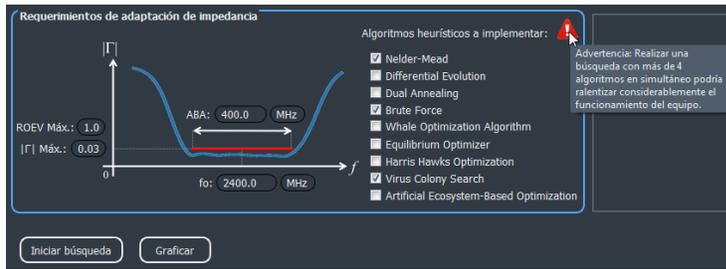
Una vez que todos los AHO hayan finalizado su ejecución, un mensaje en la ventana principal anuncia que el proceso terminó, y se muestra el tiempo consumido por cada AHO (Figura 17a). También aparecen dos nuevos botones: Ver gráficas y Ver rendimiento computacional. Clicando este último se abre una ventana con la información del consumo de recursos por parte de cada AHO, con la opción de ordenarlos de acuerdo a porcentaje de uso del procesador, memoria RAM ocupada o el tiempo transcurrido en la búsqueda (Figura 17b). De este modo, se logra comparar objetivamente el rendimiento computacional de los AHO utilizados.

Clicando el botón Ver gráficas (Figura 17a), se abre una nueva ventana titulada Gráficas. Allí se muestra para el Ejemplo 2 el desempeño de cada red MSMN diseñada mediante los AHO (Figura 18), siendo de interés conocer las figuras de mérito referidas en la sección 2, especialmente

las diseñadas para este estudio: PA y DF. En la ventana “Gráficas”, clicando en Ver figuras de mérito, se abre una nueva ventana (Figura 19), que ilustra la calidad del desempeño de las soluciones encontradas, y establece un orden de mérito. Allí el PA está indicado como “Adaptación $|\Gamma|$ (%)”.

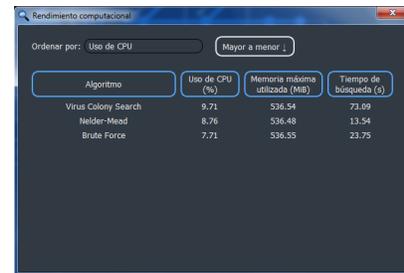
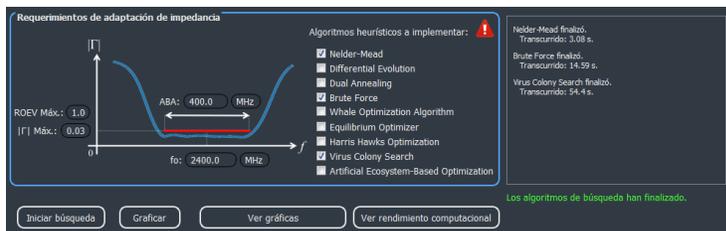
Se observó que usando el AHO Búsqueda por Colonia de Virus se cumple la condición requerida $|\Gamma| < |\Gamma|_{\text{Máx}}$ dentro del ABA, con un PCRA = 90,6 % de ese rango espectral, según la traza de la Figura 20a, mostrándose además un zoom en la Figura 20b.

En la Figura 21 se muestra la relación de onda estacionaria de voltaje ROEV de la red del Ejemplo 2 usando el AHO Búsqueda por Colonia de Virus, pero visualizando un rango de frecuencia mayor, de 1500 a 3300 MHz para apreciar el comportamiento en un ancho de banda mayor al de diseño. Considerando que una adaptación ideal se representa como una recta horizontal $ROEV = 1$, la solución encontrada se aproxima



(a) Requerimientos de adaptación para el Ejemplo 2 multistub del Modo Diseño y selección de los tres AHO (b) Mensaje de iniciación de los AHO

Figura 16: Configuración y ejecución del Modo Diseño de Eureka MSMN en el Ejemplo 2 de aplicación



(a) Mensaje de finalización de los AHO en el Modo Diseño (Ejemplo 2) (b) Rendimiento computacional

Figura 17: Finalización y aspectos de rendimiento computacional del Modo Diseño de Eureka MSMN en el Ejemplo 2 de aplicación

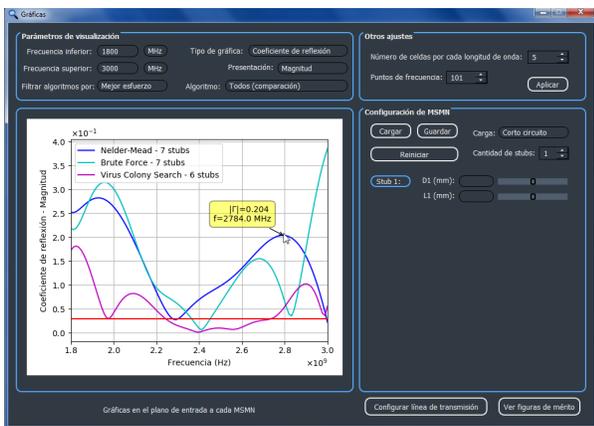


Figura 18: Ventana Gráficas, mostrando el desempeño de la red MSMN diseñada, por cada AHO, en el Ejemplo 2

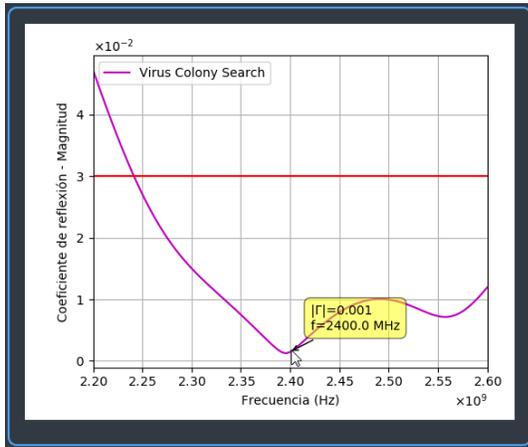
la planitud de la respuesta en el ABA, lo cual es conveniente, y la posibilidad de ajustar con deslizadores las posiciones y longitudes de los stubs en la herramienta, para una sintonía fina manual de la red diseñada, y evaluar sus efectos.

Figura 18: Ventana Gráficas, mostrando el desempeño de la red MSMN diseñada, por cada AHO, en el Ejemplo 2

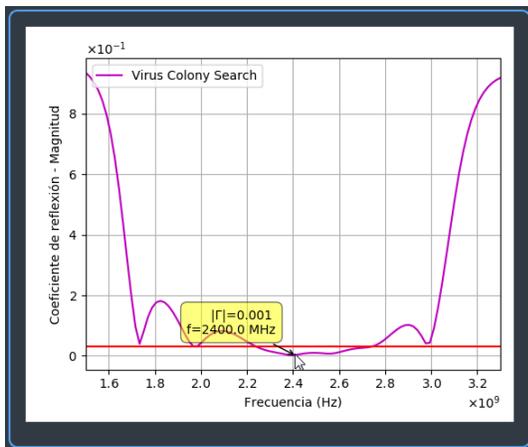
al objetivo de adaptación en banda ancha con un grado de calidad dado por un $PA = 98,69\%$ como figura de mérito global. Nótese en la Figura 21

3.6. Ejemplo 3 multistub del Modo Diseño y comparación de rendimientos de los AHO

Se ejecutó un tercer ejemplo empleando esta vez los 9 AHO, llamado Ejemplo 3, configurando



(a) Datos mostrados en la banda 2200–2600 MHz (ABA)



(b) Detalle en 1500–3300 MHz (Zoom del ABA)

Figura 20: Desempeño de la red MSMN diseñada por el AHO Búsqueda por Colonia de Virus en el Ejemplo 2

el sistema (generador, línea, y carga) con valores idénticos a los del Ejemplo 2 de la sección 3.5, así como también los mismos requerimientos de adaptación a satisfacer con la red MSMN que incluyen un $|\Gamma|_{(Máx.)} = 0,03$ en el ABA. Se realizó el proceso de búsqueda en cada AHO. Los resultados del PA, obtenidos mediante la ecuación (6) para cada algoritmo se suministran en la Tabla 2, en donde se observa que los algoritmos Recocido dual, Búsqueda de colonias de virus y Optimización basada en ecosistemas artificiales, fueron capaces de ofrecer diseños de MSMN con un elevado grado de adaptación de impedancias en un sentido absoluto, dado que ofrecen porcentajes de adaptación tan altos como $PA > 98\%$ en

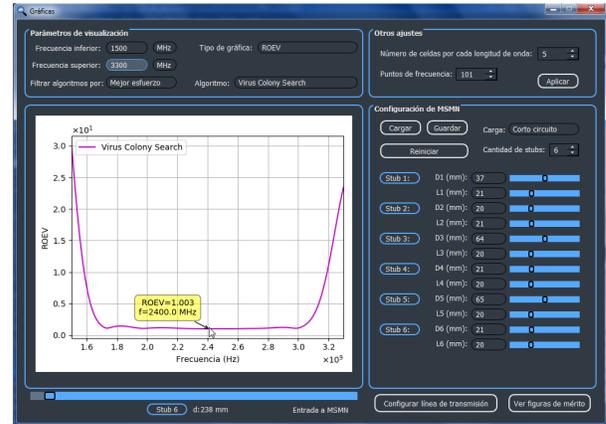


Figura 21: Ejemplo de desempeño de una red MSMN diseñada mediante el AHO Búsqueda por Colonia de Virus y uso de deslizadores paramétricos de longitud y posición de stubs para sintonía fina de la solución

el amplio ABA de 400 MHz, aproximados al cumplimiento ideal de un $PA = 100\%$. El resto de los AHO ofreció $PA > 92\%$, que si bien es inferior, pudiera ser todavía considerable para su utilización en caso de problemas de implementación de las soluciones obtenidas en primera instancia.

Como validación externa se calculó el $PCRA(\%)$ mediante la ecuación (9), y la Figura 22 muestra el $PCRA$ y la gráfica de $|\Gamma(f)|$ del Ejemplo 3 en el ABA (2200–2600 MHz), para los 9 AHO, notando que el algoritmo Recocido dual obtuvo un conveniente $PCRA = 100\%$. Por el contrario, los algoritmos Optimización de ballenas y Optimización de halcones Harris ofrecen un inadecuado $PCRA = 0\%$. Los 6 algoritmos restantes se aproximan al requerimiento de adaptación en diversos grados, con valores de $PCRA$ comprendidos entre $7,3\%$ y $88,7\%$. Esto aumenta la confianza en los resultados de *Eureka MSMN* cuando se emplean múltiples AHO, para disponer de un conjunto de opciones.

En el Ejemplo 3, no todos los algoritmos arrojaron como resultado la misma cantidad de stubs en la búsqueda óptima del diseño de la red MSMN, observándose que el algoritmo Recocido dual ofrece, con apenas 2 stubs, la mayor calidad de adaptación en forma de un $PA = 98,9\%$ y un $PCRA = 100\%$. En contraste, el algoritmo Búsqueda de colonias de virus requirió una

cantidad mucho mayor, de 7 stubs, para dar una calidad menor pero aún destacable, en forma de $PA = 98,5 \%$ y un $PCRA = 88,7 \%$. En ese sentido, la MSMN del algoritmo Recocido dual resultaría la más conveniente en el Ejemplo 3, ya que el emplear solo dos stubs implica una menor complejidad de la red para una implementación.

En cuanto al tiempo de ejecución de cada AHO en el Ejemplo 3, se observa de la Tabla 2 que los algoritmos con el mayor PA también implicaron los mayores tiempos de búsqueda.

Tabla 2: Desempeño de adaptación de los AHO del Ejemplo 3 para sus respectivas redes MSMN

Algoritmo	Cantidad de stubs, N	PCRA, %	PA, %	Tiempo de búsqueda, s
Nelder- Mead	7	7,3	92,5	2,75
Evolución diferencial	7	39,8	96,6	16,58
Recocido dual	2	100	98,9	26,61
Fuerza bruta	7	22,3	94,0	7,68
Optimización de ballenas	7	0	92,6	14,85
Optimizador de equilibrio	1	61,2	97,4	1,81
Optimización de halcones	7	0	90,7	16,75
Harris				
Búsqueda de colonias de virus	7	88,7	98,5	44,3
Optimización basada en ecosistemas artificiales	2	78,7	98,0	10,48

4. Conclusiones

En base a los resultados de la sección 3, se observa que el empleo de redes de adaptación MSMN basadas en múltiples stubs permite aproximarse a una adaptación de impedancia en un rango de frecuencia mucho mayor al que se obtiene mediante las técnicas tradicionales analíticas de uno o dos stubs, observándose que los AHO empleados son eficaces con diversos grados de éxito en la exigente tarea de hallar máximos o mínimos relativos de una función compleja de múltiples variables en el problema de adaptación de impedancias, en sistemas de líneas de transmisión para banda ancha, notándose que ciertos algoritmos

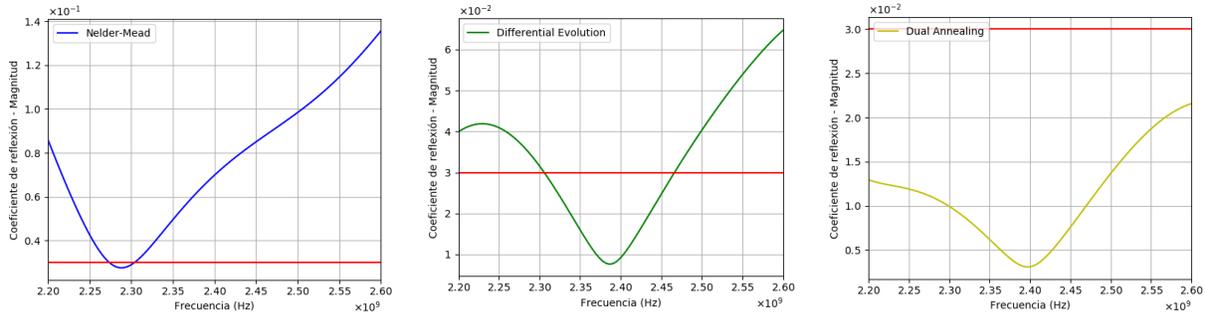
son más asertivos y eficientes que otros, en cuanto a su capacidad para adaptar con una cantidad reducida de stubs.

No todos los AHO convergen a soluciones que satisfagan exactamente los requerimientos de adaptación establecidos por el usuario en la forma de un porcentaje de adaptación $PA = 100 \%$ sin embargo, el PA en todos los AHO fue superior a un 92% , en el Ejemplo 3 de diseño, donde los algoritmos Recocido dual y Búsqueda por colonias de virus destacaron por arrojar las soluciones más aproximadas a la función ideal preestablecida. Los resultados obtenidos estimulan el estudio adicional de este tipo de técnicas para aumentar el rendimiento.

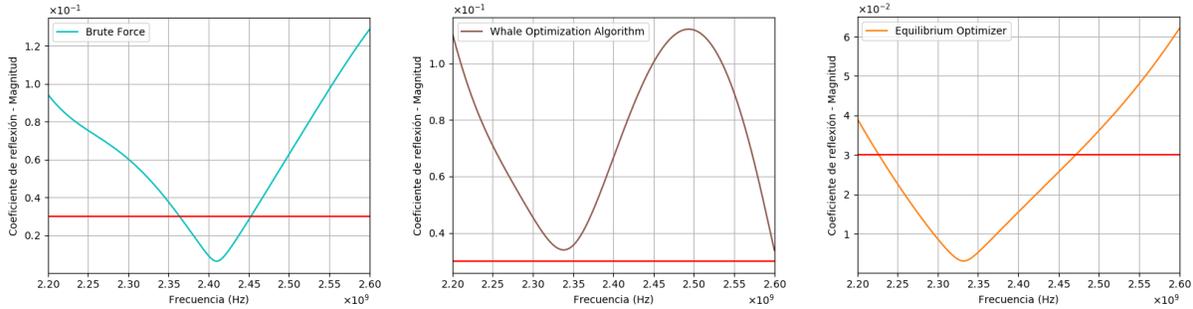
La herramienta de software *Eureka MSMN* resulta un prototipo útil como plataforma de comparación de AHOs, permitiendo clasificar la calidad de las soluciones arrojadas por los mismos, así como la eficiencia de sus procesos de búsqueda. En este sentido, *Eureka MSMN* es conveniente para la enseñanza de la adaptación de impedancias mediante redes MSMN en banda ancha, específicamente en la asignatura Sistemas de Ondas Guiadas de la Escuela de Ingeniería de Telecomunicaciones de la Facultad de Ingeniería de la Universidad de Carabobo, sirviendo como flexible complemento al sistema instruccional SEDISOG [21], diseñado, implementado y empleado regularmente en esa entidad académica, en el proceso enseñanza-aprendizaje.

Eureka MSMN puede aportar soluciones iniciales para su uso en simuladores complejos de múltiples dominios físicos, orientados específicamente a la implementación efectiva, por ejemplo en sistemas de líneas de microcinta o en los tradicionales sistemas coaxiales o de guía de onda. Es importante destacar que el elevado componente estocástico de los AHO, empleados aquí como cajas negras de Python, implica que en ocasiones y usando un AHO específico, el mismo problema programado resulte en soluciones de configuraciones distintas (si bien aproximadas) para los stubs.

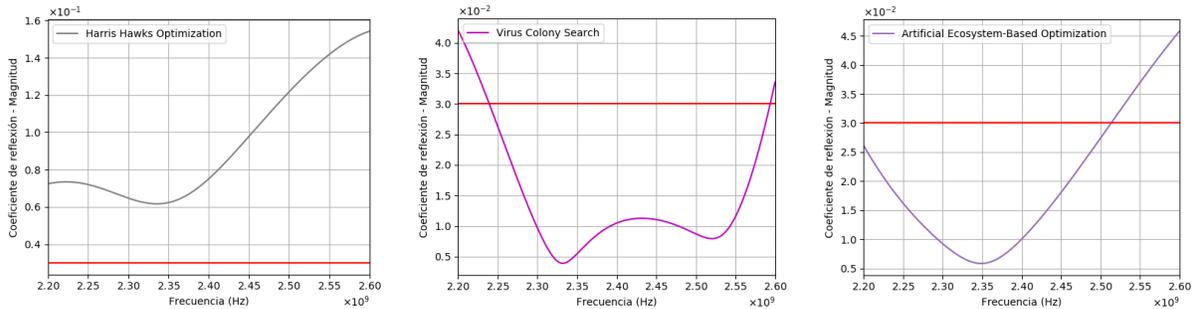
Para estimular el uso, prueba y validación de resultados de *Eureka MSMN*, la



(a) Nelder-Mead. $PCRA = 7,3 \%$ (b) Evolución diferencial. $PCRA = 39,8 \%$ (c) Recocido dual. $PCRA = 100 \%$



(d) Fuerza bruta. $PCRA = 22,3 \%$ (e) Optimización de ballenas. $PCRA = 0 \%$ (f) Optimizador de equilibrio. $PCRA = 61,2 \%$



(g) Optimización de halcones Harris. $PCRA = 0 \%$ (h) Búsqueda de colonias de virus. $PCRA = 88,7 \%$ (i) Optimización basada en ecosistemas artificiales. $PCRA = 78,7 \%$

Figura 22: Espectros de $|\Gamma(f)|$ y $PCRA(\%)$ para cada uno de los 9 AHO del Ejemplo 3 de diseño de redes MSMN. Las líneas horizontales rojas representan el $|\Gamma|_{Máx.} = 0,03$ especificado

herramienta está disponible en el repositorio <https://github.com/hidalgoabraham/eureka-msmn>, en donde se incluyen código fuente, archivo ejecutable y manuales de instalación y de usuario.

5. Reconocimientos

La presente investigación fue reconocida con Mención Honorífica por la Escuela de Ingeniería de Telecomunicaciones de la Facultad de Ingeniería

de la Universidad de Carabobo, Venezuela, según el Acta de Examen No. 10231.

6. Referencias

- [1] C. A. Balanis, *Antenna Theory*, 3 ed. John Wiley & Sons, 2005.
- [2] A. Bianchi, *Sistemas de Ondas Guiadas*. Marcombo, 1980.
- [3] R. Collin, *Foundations for Microwave Engineering*, 2 ed. John Wiley & Sons, 2001.

- [4] D. Pozar, *Microwave Engineering*, 4 ed. John Wiley & Sons, 2011.
- [5] P. H. Smith, "Electronic Applications of the Smith Chart," in *Waveguide, Circuit and Component Analysis*, 1st ed. McGraw-Hill Book Company, 1969, ch. 9.
- [6] M. Alsharif, A. Kelechi, M. Albreem, S. Chaudhry, M. Zia, and S. Kim, "Sixth Generation (6G) Wireless Networks: Vision, Research Activities, Challenges and Potential Solutions," *Symmetry*, vol. 12, no. 4, p. 676, 2020. <https://doi.org/10.3390/sym12040676>
- [7] C. Regoli, "Impedance matching by using a multi-stub system," in *Proceedings of the 7th WSEAS International Conference on Simulation, Modeling and Optimization (2007)*, ser. SMO'07. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2007, p. 341–345.
- [8] N. I. Dib, A. H. Sharaqa, and R. A. Formato, "Variable z_0 applied to biogeography based optimized multi-stub matching network," in *IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting (2013)*, 2013, pp. 2129–2130. 10.1109/APS.2013.6711723
- [9] M. E. Aydemir, M. Ilarslan, K. Buyukatak, and E. Duymaz, "Swarm intelligence in solution of impedance matching problem in satellite transmitters," in *Proceedings of 5th International Conference on Recent Advances in Space Technologies - RAST2011*, 2011, pp. 436–440. 10.1109/RAST.2011.5966873
- [10] K. R. Chowdhary, *Fundamentals of Artificial Intelligence*, 1st ed. New Delhi: Springer, ch. 9, p. 239. <https://doi.org/10.1007/978-81-322-3972-7>
- [11] P. M. Rabanal, "Algoritmos heurísticos y aplicaciones a métodos formales," Tesis doctoral, Universidad Complutense de Madrid, Madrid, España, 2010.
- [12] N. V. Thieu and S. Mirjalili, "MEALPY: a Framework of The State-of-The-Art Meta-Heuristic Algorithms in Python," Oct. 2022. 10.5281/zenodo.7255425. [Online]. Available: <https://doi.org/10.5281/zenodo.7255425>
- [13] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965. <http://dx.doi.org/10.1093/comjnl/7.4.308>
- [14] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997. <https://doi.org/10.1023/A:1008202821328>
- [15] Y. Xiang, D. Y. Sun, W. Fan, and X. G. Gong, "Generalized simulated annealing algorithm and its application to the Thomson model," *Physics Letters A*, vol. 233, no. 3, pp. 216–220, 1997. [https://doi.org/10.1016/S0375-9601\(97\)00474-X](https://doi.org/10.1016/S0375-9601(97)00474-X)
- [16] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- [17] B. S. Afshin Faramarzi Mohammad Heidarinejad and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowledge-Based Systems*, vol. 191, p. 105190, 2020. <https://doi.org/10.1016/j.knsys.2019.105190>
- [18] A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019. <https://doi.org/10.1016/j.future.2019.02.028>
- [19] M. D. Li, H. Zhao, X. W. Weng, and T. Han, "A novel nature-inspired algorithm for optimization: Virus colony search," *Advances in Engineering Software*, vol. 92, pp. 65–88, 2016. <https://doi.org/10.1016/j.advengsoft.2015.11.004>
- [20] L. Zhao, W. Wang and Z. Zhang, "Artificial Ecosystem-Based Optimization: A Novel Nature-Inspired Meta-Heuristic Algorithm," *Neural Computing and Applications*, vol. 32, pp. 9383–9425, 2019. <https://doi.org/10.1007/s00521-019-04452-x>
- [21] E. Alvarado and Á. Vargas, "Desarrollo de un software educativo didáctico interactivo para sistemas de ondas guiadas versión 3," Trabajo especial de grado, Escuela de Telecomunicaciones, Facultad de Ingeniería, Universidad de Carabobo, Carabobo, Venezuela, 2016.