

Revisão Teórica Sobre as Técnicas de Otimização Metaheurística na Síntese de Superfícies Seletivas de Frequência

Flávio Henry Cunha da Silva Ferreira, Fabrício José Brito Barros,
Jasmine Priscyla Leite de Araújo e Miércio Cardoso de Alcântara Neto

Resumo — Este artigo apresenta a progressão das técnicas de otimização computacionais híbridas e bioinspiradas para síntese de superfícies seletivas de frequência (FSS), incluindo a velocidade computacional e eficiência de tais processos. Duas metodologias serão analisadas. A primeira possui uma rede neural MLP (multi-layer perceptron, ou perceptron multicamada) combinada com um algoritmo genético (AG). A segunda baseia-se na rede neural GRNN (general regression neural network) combinada com o algoritmo de busca cuco multiobjetivo (MOCS), que se mostra mais eficiente para otimização das propriedades eletromagnéticas de FSS, pois possui velocidade de processamento consideravelmente maior que métodos mais comuns baseados em retropropagação e AGs.

Palavras-chave—*Superfícies seletivas de frequência; Otimização; Busca cuco; MLP; Algoritmos genéticos; GRNN*

Abstract — This paper presents the progression of hybrid and bioinspired computational optimization techniques for the synthesis of frequency selective surfaces (FSS), including computational speed and efficiency of such processes. Two methodologies will be analyzed. The first has an MLP neural network (multi-layer perceptron, or multilayer perceptron) combined with a genetic algorithm (AG). The second is based on the GRNN (general regression neural network) network combined with the multiobjective cuckoo search algorithm (MOCS), which is more efficient for optimizing the electromagnetic properties of FSS, since it has a considerably higher processing speed than more backpropagation and AGs.

Keywords—*Frequency selective surfaces; Optimization; Cuckoo search; MLP; Generic algorithms; GRNN*

I. INTRODUÇÃO

As superfícies seletivas de frequência (FSS, sigla em inglês) são filtros passivos de ondas eletromagnéticas (EM) que auxiliam na separação dos diversos sistemas que operam em rádio frequência (RF), possibilitando aplicações práticas e grande versatilidade de uso. Estudos deste tipo de estruturas, atualmente, se concentram na faixa de micro-ondas, porém, tais dispositivos podem ser projetados para outros sistemas e frequências mais elevadas, por exemplo, as ondas milimétricas (padrão IEEE 802.15.3c), ou ainda, em terahertz, possibilitando filtragem eficiente em frequências elevadas [1]. De modo geral, não existe uma solução analítica diretamente extraída a partir da resposta em frequência de um dispositivo; desta forma, a análise de suas características espectrais requer a aplicação de técnicas numéricas para análise de onda completa, tal como o método das

integrais finitas (FIT, sigla em inglês). Atrelado a isso, deve-se considerar o custo computacional exigido para a implementação destes métodos, conduzindo muitos pesquisadores ao método da tentativa e erro, até que se consiga o ajuste satisfatório de alguns requisitos do projeto dos dispositivos. Isso, obviamente, leva a um procedimento trabalhoso e dependente da intervenção humana. Não obstante, faz-se necessário o emprego de técnicas de otimização que acelerem a obtenção de estruturas periódicas de especificações de filtragem desejadas.

Neste íterim, autores têm realizado a junção de redes neurais artificiais e algoritmos de otimização natural, como os clássicos algoritmos genéticos e a otimização por enxame de partículas, ou ainda, os mais recentes algoritmos de busca cuco no projeto e otimização das superfícies seletivas de frequência [2]. Uma extensa pesquisa analítica tem sido realizada para descrever as propriedades de reflexão e transmissão das FSS com o emprego de técnicas de otimização para se obter as dimensões exatas, necessárias ao atendimento dos requisitos de projeto [3, 4, 5].

Neste estudo objetiva-se tecer uma discussão acerca do emprego de técnicas de otimização computacionais híbridas, incluindo custo computacional e a eficiência / eficácia de tais métodos. Especificamente, a discussão se concentra em torno de dois métodos de otimização híbrida, aplicados no projeto e síntese de FSS. A primeira utiliza rede neural MLP (*multi-layer perceptron*, ou perceptron multicamada) que se combina com um AG (algoritmo genético) e/ou uma PSO (otimização por enxame de partículas) [1, 6]. No segundo caso estudado, uma rede neural GRNN (*general regression neural network*, ou rede neural de regressão geral) é associada ao algoritmo de busca cuco multiobjetivo (MOCS), que se mostra mais eficiente para obter os resultados desejados, possui menor custo computacional e, em consequência, maior velocidade de processamento. Desta forma, trata-se de um método mais robusto em relação aos clássicos que associam RNA de retropropagação a um AG / PSO [7, 8]. Na segunda técnica, há uma substancial minimização no tempo demandado para o processamento dos dados EM, reduzindo desta forma o tempo total exigido pelo processo de otimização, o que pode ser atribuído ao fato da GRNN não necessitar da retropropagação. Seu aprendizado é realizado de maneira *forward*, ou seja, apenas em uma varredura do código [5, 7].

II. O MÉTODO HÍBRIDO MLP PARA SÍNTESE DE FSS

A. Contexto e Funcionamento

A perceptron multicamada (ou MLP) é uma rede neural baseada em retropropagação, amplamente utilizada em múltiplos contextos. Exemplos podem ser encontrados em áreas financeiras / econômicas (através da classificação de clientes por mérito e crédito) [6], na medicina - como diagnosticador de problemas na artéria coronária [6] - e até mesmo como um válido indicador da emoção humana (classificação de sentimentos através da música) [9].

Seu funcionamento é dado a partir de camadas de neurônios artificiais (neste caso, chamados de neurônios perceptron), em que cada um destes é regido por uma função de ativação e uma função de rede. A função de rede representa uma combinação linear das entradas do neurônio, e representa os dados que a rede neural fornece para a célula processar. A saída do neurônio é única, e relaciona a função de rede com a função de ativação – uma transformação linear ou não-linear. Diversas funções de ativação como a sigmoide e a tangente hiperbólica são comumente aplicadas, e cada uma destas provoca uma otimização diferente, dependendo da composição da rede neural [8].

Quanto aos tipos de camada, uma MLP possui a mesma classificação de três camadas de uma RNA genérica. Estas são: camada de entrada, camada oculta e camada de saída – sendo que somente a oculta pode possuir uma ou mais camadas, de acordo com a necessidade de otimização e/ou processamento [7, 10]. Cada um dos neurônios na camada de entrada é um atributo, que é processado na MLP e entregue a uma classe (saídas). No caso de uma FSS, as entradas são geralmente valores dimensionais da estrutura (periodicidade e dimensões de modelamento), sendo as saídas a frequência central de ressonância da estrutura e sua largura de banda [3, 5, 11].

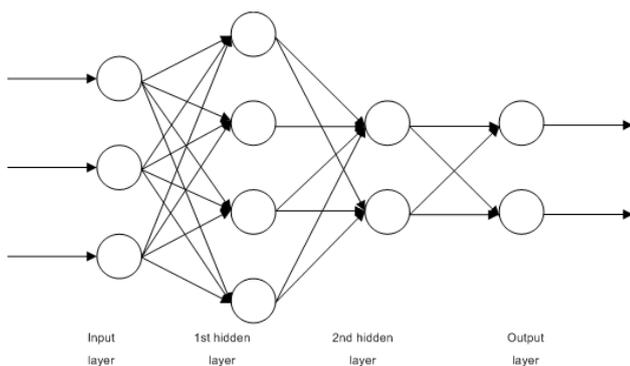


Fig. 1. Representação de uma MLP genérica

A rede MLP precisa ser treinada antes que esta seja executada para otimização. Ou seja, esta precisa aprender a partir de um banco de dados referencial o resultado desejado, para que o seu algoritmo conduza às saídas desejadas. Para o processo de aprendizado, há diversos tipos de algoritmo de treinamento, como os comumente utilizados *backpropagation* (ou retropropagação), RPROP (retropropagação resiliente) e Levenberg-Marquadt [7]. Em [10] pode-se notar o uso de um algoritmo RPROP para a rede MLP, e em [8] pode-se observar

que o escolhido foi o Levenberg-Marquadt, que possui convergência mais rápida em relação ao RPROP.

Embora a MLP já tenha sido implementada para trabalhos envolvendo FSS sem qualquer outro algoritmo auxiliador, como em [10], a utilização deste tipo de rede para otimização bioinspirada de FSS nota-se em [4, 5, 7, 8]: um compilado de trabalhos científicos publicados entre 2009 e 2014. Em suma, a MLP é aplicada para sintetizar a estrutura, servindo também como a função de ajuste (também chamada de função de fitness) para o algoritmo genético ou PSO selecionados para compor a solução híbrida proposta. Este método, porém, está tornando-se cada vez mais obsoleto devido às vantagens da GRNN e outros processos no estado da arte.

B. Progressão da MLP híbrida para FSS

Considerando o trabalho científico mais antigo mencionado em [10], a implementação da MLP sem métodos auxiliares bioinspirados fornece resultados satisfatórios, porém demorados. Quando limitado a 5000 épocas, o tempo de treinamento da rede neural varia entre 18 a 25 minutos, com um erro quadrático médio que chega a 10^{-4} – isto considerando apenas os resultados obtidos nas aplicações com FSS. O computador utilizado para a otimização possui CPU Pentium® M de 1,4 GHz com memória RAM de 512MB.

Em 2009, o método MLP híbrido em [7], abreviado como GA-ANN, obteve grande destaque por fornecer em consideravelmente menor tempo o treinamento e otimização das FSS. Combinando um algoritmo genético a uma MLP de 3 dimensões de entrada, 10 neurônios de camada oculta e 2 saídas (frequência ressonante e largura de banda), o tempo de treinamento em 20000 épocas é pouco menos de um minuto. Estes resultados refletem o treinamento com RPROP, que produziu também erro quadrático médio menor comparando ao trabalho anteriormente explicado – em torno de $2,8 \times 10^{-5}$. Mostra-se também a diferença entre o treinamento com retropropagação normal e RPROP, sendo RPROP o menor consumidor de tempo e portador do menor erro médio quadrático.

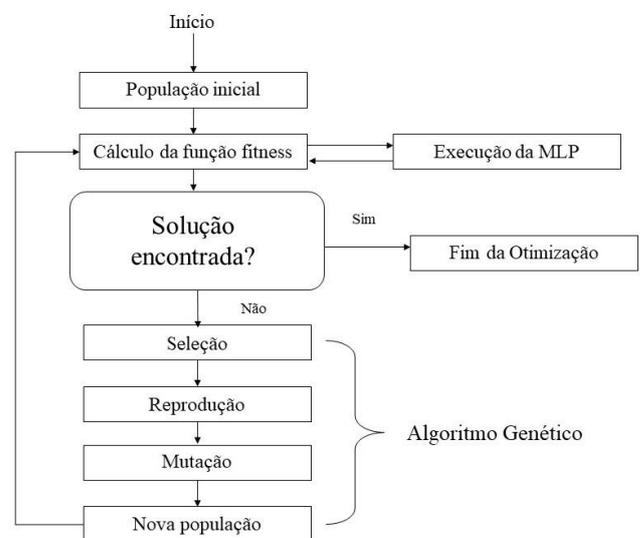


Fig. 2. Fluxograma da otimização GA-ANN

Na referência [8], um artigo mais recente de 2014, são ilustrados resultados melhores, já que além da implementação de uma AG, utilizou-se também do MoM (*method of moments*, ou método dos momentos) junto à MLP. O MoM contribui a uma convergência mais rápida ao melhor erro quadrático médio, a partir de estimativas estatísticas dos dados obtidos para otimização [12] – neste caso, obtidos no *software* “Ansoft Designer”. Observa-se um custo de processamento computacional significativamente menor com esta técnica, com apenas 15 épocas necessárias para treinamento e convergência. O erro quadrático médio é de 10^{-10} . O tempo de treinamento para a MLP também foi encurtado, com 38,3 segundos de execução (para estas especificações: CPU de 2.67 GHz, 4GB de memória RAM).

Em nenhum destes trabalhos o tempo total de execução da otimização híbrida é mencionado, já que o processo com o uso de MLP e algoritmo genético / PSO pode demorar vários minutos, ou até mesmo horas, para concluir [7, 8, 10].

III. O MÉTODO HÍBRIDO GRNN + MOCS

A. Resumo sobre a GRNN

Em 1991, E. Specht definiu e demonstrou uma rede neural de regressão geral, assim como seus benefícios e vantagens. Trata-se de uma rede neural estocástica, sendo necessária apenas uma passagem pelo banco de dados referencial para que esta seja treinada [13]. Outras vantagens salientadas pelo autor incluem a inexistência de convergência a um ponto de máximo ou mínimo local, e resultado convincente mesmo que hajam poucas amostras para treinamento.

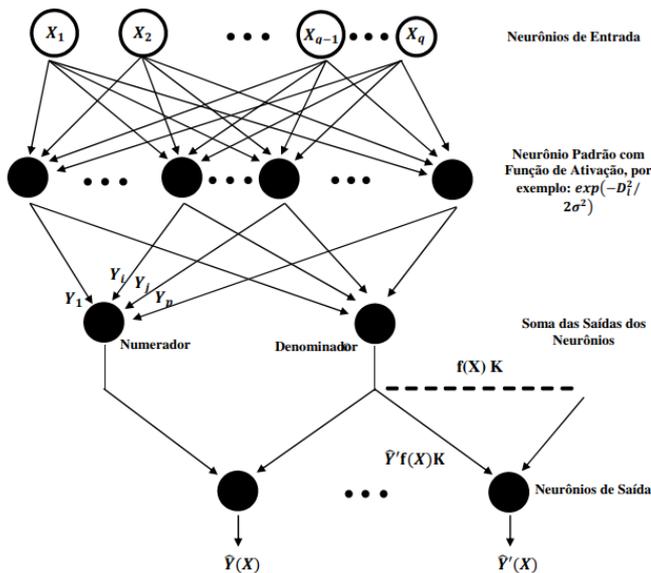


Fig. 3. Representação da rede neural de regressão generalizada

Esta é uma rede probabilística – pertencente ao grupo das redes denominadas radial de base (RBF) – pois sua definição e desenvolvimento partem de uma função distribuição de probabilidade e uma média condicional da saída em relação à entrada. Esta média condicional provê o nome, já que se trata de um progresso de regressão da saída Y dado que foi provocada

uma entrada X no sistema. A partir deste conceito, elimina-se a necessidade de treinamento iterativo (como exemplo, o *backpropagation*), já que a própria rede executa a regressão.

Para que o cálculo da regressão seja realizado, há dois somatórios que captam as saídas dos neurônios de camada oculta e transformam tais dados em numerador e denominador da função distribuição – demonstrados na Fig. 3. O resultado da divisão fornece o valor da saída após a regressão. Segundo as definições de [5] e a aplicação em [2, 7], a função de ativação exponencial recomendada e sua saída são dados por:

$$g_i = \exp\left(-\frac{D_i^2}{2\sigma^2}\right) \quad (1)$$

$$\hat{Y}(X) = \frac{\sum_{i=1}^n Y_i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^n \exp\left(-\frac{D_i^2}{2\sigma^2}\right)} \quad (2)$$

No qual σ representa a interpretação de densidade que a rede neural deve demonstrar – também chamado de parâmetro de suavidade (*smoothness parameter*). Para valores de σ próximos de zero, a estimativa de densidade pode tornar-se não-gaussiana; já para valores muito grandes, a densidade apresentada é gaussiana e suavizada, fazendo com que $Y(X)$ seja resultado de uma média das amostras Y_i [13]. Já o D_i é a distância entre a amostra de treinamento (X_i) e o ponto de predição de resultado (X), representado em [2, 5, 7] como:

$$D_i^2 = (X - X_i)^T (X - X_i) \quad (3)$$

B. Busca Cuco Simples (CS) e Multiobjetivo (MOCS)

O algoritmo de busca cuco, método bioinspirado introduzido por X.-S. Yang e S. Deb [14], trata-se de uma representação computacional baseada no mimetismo e parasitismo dos pássaros cuco – que colocam seus ovos nos ninhos de outros tipos de pássaro. Isto faz com que as espécies anfitriões possam ser enganadas, com três tipos de ação: ignorar os ovos de cuco, descartar estes ovos “alienígenas” ou abandonar completamente seu ninho e fazê-lo em outro lugar. Caso o ovo seja ignorado, os filhotes de cuco nascem e crescem mais rapidamente que os das outras aves, deslocando outros ovos para fora do ninho e aumentando sua partilha de comida [11, 14].

De acordo com os criadores da CS, o processo pode ser explanado a partir de três regras [14]:

- Cada cuco concebe um ovo por vez, e deposita-o em um ninho aleatório. Em cada ovo, considera-se uma solução em potencial – metaheurística.
- Os melhores ninhos são os portadores dos melhores ovos (soluções), e estes sobreviverão às próximas gerações – elitismo.
- O número de ninhos disponíveis é fixo e definido pelo escritor do código. A probabilidade do pássaro hospedeiro descobrir o ovo de cuco é definida como P_a

$\in [0, 1)$. Após isto, o pássaro pode escolher descartar este ovo ou abandonar seu ninho – descarte das piores soluções.

Os pássaros cuco neste método movem-se através dos designados Voos de Lévy. Tal artifício provê um padrão aleatório de voo no qual o cada cuco i trilhará para encontrar ninhos:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Lévy(\beta) \quad (4)$$

$$Lévy \cong u = t^{-(\beta+1)}; (1 < \beta \leq 3) \quad (5)$$

Início

Inicialize a função objetivo $f_1(x), \dots, f_k(x), x = (x_1, \dots, x_d)^T$

Gere uma população inicial de n ninhos hospedeiros x_i , e cada um com k ovos

quando ($t < \text{MaxGeração}$) ou (critério de parada)

Obtenha um Cuco (denominado i) aleatoriamente por voo de Lévy

Avalie e cheque se este é o valor ótimo de Pareto

Escolha um ninho entre n (denominado j) aleatoriamente

Avalie as k soluções para o j -ésimo ninho

se os ninhos i dominam as novas soluções dos ninhos j

Substitua o ninho i pelo novo conjunto de soluções do ninho j

fim

Abandone uma fração (p_a) dos piores ninhos

Mantenha as melhores soluções (ou conjunto de ninhos não dominados)

Classifique e encontre a atual solução ótima de Pareto

fim

Resultados pós processamento e visualizações

Fim

Fig. 4. Pseudocódigo da busca cuco multiobjetivo

Uma melhoria desta otimização focando em múltiplos objetivos, a busca cuco multiobjetivo (MOCS), foi desenvolvida pelos próprios criadores da CS [15]. Comparado ao algoritmo anterior, as regras são ligeiramente modificadas para que o multiobjetivo seja implementado. Isto, em modos práticos, significa que ao invés do pássaro cuco i por apenas um ovo por vez, este pode colocar um número k de ovos em um só ninho. Além disto, a terceira regra também é modificada de modo a gerar outro ninho com k ovos cada vez que P_a descartar um ninho com ovo “alienígena”. Desta forma, obtém-se mais soluções possíveis a cada ninho gerado.

Toda otimização multiobjetiva tem seus resultados avaliados através de uma Frente de Pareto – representando o valor de referência ou valor “ótimo” que as saídas devem alcançar [15, 16]. Sendo assim, em um código que implemente o MOCS, é geralmente feita uma comparação com a Frente de Pareto a partir das soluções obtidas. A fig. 4. elucida um pseudocódigo com os passos necessários para a criação de uma MOCS computacional [16].

C. Resultados da Otimização para FSS

Na referência [16], nota-se a comparação entre a GRNN + MOCS e outros tipos dois de otimização bioinspirada associadas à rede neural: a MOGA (*multiobjective genetic algorithm*, ou algoritmo genético multiobjetivo) e a MOBA (*multiobjective bat algorithm*, ou algoritmo de ecolocalização do morcego multiobjetivo). A atenção neste artigo voltou-se à busca cuco

multiobjetivo pois é o processo significativamente mais rápido dos três.

Há duas FSS definidas em [16]: uma de estrutura patch triangular e outra de estrutura patch com espiras triangulares, sendo o objetivo sintonizá-las com frequência ressonante em 11,0 GHz e largura de banda igual a 4,0 GHz – valores este tratados como a saída desejada da rede neural. Para tal, a rede neural possui 4 características da FSS como valores de entrada: a periodicidade das células ($T = T_x = T_y$), dimensões de altura/base do triângulo ($W = W_x = W_y$), permissividade relativa do substrato (ϵ_o) e a espessura do substrato (h). Os únicos valores fixos são a permissividade relativa igual a 4,4 e a espessura do substrato igual a 1,57 milímetros. Convém a análise de somente uma destas estruturas, apenas para verificar a progressão entre as técnicas híbridas utilizadas. Sabendo disto, os valores de processamento que serão mostrados fazem jus à FSS patch de espira triangular, representada na fig. 5.

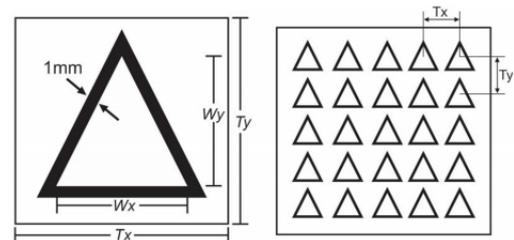


Fig. 5. Diagrama da FSS patch com espiras triangulares

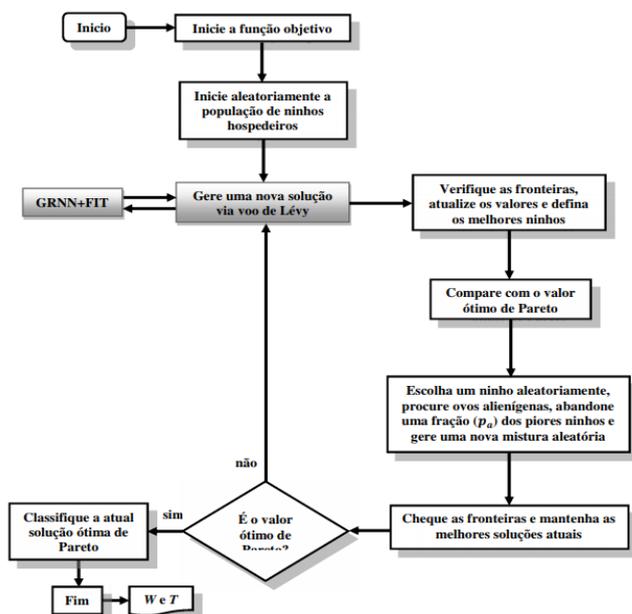


Fig. 6. Fluxograma da Otimização GRNN + MOCS

A rede neural de regressão generalizada utilizada em [3, 16] possui, portanto, 4 entradas, 5 neurônios de camada oculta e 2 saídas. Os resultados em [16] a seguir compilados mostram tempo e custo de processamento muito mais baixos comparados à técnica MLP híbrida. Nos resultados dos trabalhos citados que utilizam a rede neural MLP, nenhum dos autores especificou o tempo total do processo de otimização, por ser demasiadamente grande. Todavia, para a implementação com GRNN, a duração total da otimização em [16] para as duas estruturas não excede 15 minutos (dados obtidos com CPU de 2,53 GHz e memória RAM de 4GB).

A primeira bateria de testes traz o uso da GRNN com o algoritmo genético multiobjetivo. O tempo de otimização geral registrado para a espira triangular foi de 654,9 segundos com 44 iterações.

Em seguida, temos os resultados do método GRNN + MOBA. Para a espira triangular, este se mostrou pior, com tempo de execução igual a 733,9 segundos em 52 iterações.

Entretanto, os resultados observados com o modelo cuco para o tempo de otimização geral foram de apenas 71,8 segundos em 33 iterações – um grande salto em velocidade e poupança de processamento computacional. As dimensões encontradas da espira triangular patch são $T_x = T_y \approx 9,2 \text{ mm}$, e $W_x = W_y \approx 6,4 \text{ mm}$. O fluxograma deste tipo de otimização encontra-se na fig. 6.

CONCLUSÃO

As duas principais formas de otimização híbrida bioinspiradas para FSS foram postas em evidência neste artigo, a cada vez mais defasada que utiliza a rede neural perceptron multicamada (MLP), e a promissora e veloz baseada na rede neural de regressão generalizada com busca de cuco multiobjetivo (GRNN + MOCS). A progressão das melhorias de otimização das FSS foi analisada, em função do tempo. Nota-se que a otimização GRNN + MOCS possui resultados muito

melhores que a anterior, com treinamento rápido e duração total de apenas alguns minutos. Em 2006, apenas o treinamento da rede neural poderia tomar mais tempo (18 a 25 minutos) [10] do que todo o processo de otimização em 2015 [3, 16].

Vale ressaltar que a rede GRNN nunca havia sido utilizada para otimizar FSS, sendo [3, 16] estudos no estado da arte. Esta implementação híbrida, metaheurística e bioinspirada pode, portanto, ser descrita como uma técnica eficiente, promissora e ainda pouco utilizada, ideal para sua inclusão em trabalhos mais complexos envolvendo superfícies seletivas de frequência.

REFERÊNCIAS

- [1] A. L. P. S. Campos, “Estudo da Flexibilidade de Projeto de Superfícies Seletivas de Frequência”, II Congresso de Pesquisa e Inovação da Rede Norte Nordeste de Educação Tecnológica, 2007.
- [2] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, 2nd edition. Luniver Press, UK, 2010.
- [3] M. C. Alcântara Neto, J. P. L. Araújo, F. J. B. Barros, A. N. Silva, G. P. S. Cavalcante and A. G. D’Assunção, “Bioinspired multiobjective synthesis of X-band FSS via general regression neural network and cuckoo search algorithm”, *Microwave and Optical Technology Letters*, Vol. 57, No. 10, October 2015.
- [4] M. R. da Silva, C. de L. Nóbrega, P. H. da F. Silva, and Adaildo G. D’Assunção, “Optimization of FSS with Sierpinski Island fractal elements using population-based search algorithms and MLP neural network”, *Microwave and Optical Technology Letters*, Vol. 56, No. 4, April 2014.
- [5] P.H. da F. Silva, R. M. S. Cruz and A. G. D’Assunção, “Blending PSO and ANN for optimal design of FSS filters with Koch Island patch elements”, *IEEE Transactions on Magnetics*, Vol. 46, No. 8, August 2010.
- [6] M. Paliwal and U. A. Kumar, “Neural networks and statistical techniques: A review of applications”, Vol. 36, Issue 1, January 2009, pgs. 2-17.
- [7] R. M. S. Cruz, *Análise e Otimização de Superfícies Seletivas de Frequência Utilizando Redes Neurais Artificiais e Algoritmos de Otimização Natural*. Tese de Mestrado. Natal: UFRN, 2009.
- [8] W. C. de Araújo, H. W. C. Lins, A. G. D’Assunção Jr., J. L. G. Medeiros, and A. G. D’Assunção, “A bioinspired hybrid optimization algorithm for designing broadband frequency selective surfaces”, *Microwave and Optical Technology Letters*, Vol. 56, No. 2, February 2014.
- [9] Y.-P. Lin, C.-H. Wang, T.-L. Wu, S.-K. Jeng and J.-H. Chen, “Multilayer perceptron for EEG signal classification during listening to emotional music”.
- [10] P. L. da Silva, *Modelagem de Superfícies Seletivas de Frequência e Antenas de Microfita utilizando Redes Neurais Artificiais*. Natal: UFRN, 2006.
- [11] I. Fister Jr., I. Fister and X. S. Yang, “Discussion of ‘Estimation of reference evapotranspiration using neural networks and cuckoo search algorithm’ by Shahaboddin Shamshirband, Mohsen Amirmojahedi, Milan Gocic, Shatirah Akib, Dalibor Petkovic, Jamshid Piri and Slavisa Trajkovic”, *Journal of Irrigation and Drainage Engineering*, Vol. 144, No. 2, 2017.
- [12] X. Dardenne and C. Craeye, “Method of Moments simulation of infinitely periodic structures combining metal with connected dielectric objects”, *IEEE Transactions on Antennas and Propagation*, Vol. 56, No. 8, August 2008.
- [13] D. F. Specht, “A general regression neural network”, *IEEE Transactions on Neural Networks*, Vol. 2, No. 6, November 1991.
- [14] X.-S. Yang, and S. Deb, “Engineering optimisation by cuckoo search”, *Int. J. Mathematical Modelling and Numerical Optimisation*, Vol. 1, No. 4, pgs. 330–343, 2010.
- [15] X.-S. Yang, S. Deb, “Multiobjective cuckoo search for design optimization”, *Computers and Operations Research*, 2011.
- [16] M. C. Alcântara Neto, *Síntese de Superfícies Seletivas de Frequência para Micro-ondas Utilizando Otimização Multiobjetivo Bioinspirada*. Tese de Doutorado. Belém: UFPA, 2015.