Algoritmo de Polinização das Flores Modificado usando Aprendizagem Baseada em Oposição

Izabele V. O. Leite[1], Marcos H. F. Marcone[2], Fábio A. P. Paiva[3]

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN)
Caixa Postal 59143-455 – Parnamirim – RN – Brasil

izaleite11@hotmail.com[1], marcosmarcone48@gmail.com[2],
fabio.procopio@ifrn.edu.br[3]

Abstract. *This paper presents a new variant for a metaheuristic known as Flower Pollination Algorithm (FPA). The variant uses the concept of opposition-based learning on the flower that has the worst fitness value. The modified FPA aims to preserve the solutions diversity in order to avoid premature convergence. To evaluate the performance of the variant, it was compared to standard FPA considering problem dimensionality and number of fitness function evaluations. Several computational experiments were performed to optimize four benchmark functions. The superiority of the modified FPA has been evidenced when its results are compared to standard FPA.*

Resumo. *Este artigo apresenta uma nova variante da meta-heurística conhecida como Algoritmo de Polinização das Flores (FPA). A variante utiliza o conceito de aprendizagem baseada em oposição sobre a flor que apresenta o pior valor de fitness. O FPA modificado tem como objetivo preservar a diversidade de soluções a fim de evitar convergência prematura. Para avaliar o desempenho da variante, ela foi comparada com o FPA padrão considerando dimensionalidade do problema e número de avaliações da função objetivo. Diversos experimentos computacionais foram realizados para otimizar quatro funções de referência. A superioridade do FPA modificado foi evidenciada quando os seus resultados foram comparados com o FPA padrão.*

1. Introdução

O processo de otimização consiste na tarefa de encontrar a melhor solução para um determinado problema. Em geral, as técnicas de otimização são usadas quando não existem alternativas simples para se resolver um problema [Viali 2011]. Muitas técnicas convencionais de otimização não apresentam bons resultados quando são usadas em problemas não-lineares e multimodais e, por isso, a tendência atual vem sendo a utilização de algoritmos meta-heurísticos inspirados na natureza, os quais têm apresentado resultados bastante promissores.

A inspiração baseada na natureza tem sido uma característica constante dos algoritmos de otimização propostos recentemente. Eles vêm sendo aplicados em diversos problemas do mundo real e se destacam aqueles que se baseiam: a) no comportamento social e cooperativo de pássaros e de peixes [Kennedy e Eberhart 1995], b) na bioluminescência produzida pelos vaga-lumes [Yang 2009], c) na capacidade de ecolocalização dos morcegos [Yang 2010] e; d) no esquema cooperativo de aranhas sociais [Cuevas *et al* 2013].

Além da bioluminescência dos vaga-lumes e da ecolocalização dos morcegos, Yang (2012) também estudou o processo de reprodução das flores com o objetivo de propor um novo algoritmo de otimização conhecido como Algoritmo de Polinização das Flores (FPA – *Flower Pollination Algorithm*). Na modelagem do novo algoritmo, Yang considerou que uma flor pode ser usada para representar uma solução de um determinado problema. A transferência do pólen, quando ocorre a curtas distâncias, pode ser usada para implementar um processo de otimização local, ao passo que a transferência a longas distâncias pode implementar a otimização global.

Um problema comumente enfrentado pelos algoritmos meta-heurísticos é a convergência prematura. Ela ocorre quando o algoritmo fica "preso" em ótimos locais e, devido a sua baixa diversidade, ele para de buscar a solução ótima global. A convergência prematura pode impedir que o algoritmo obtenha bons resultados, tornando-o ineficiente.

Vários trabalhos foram propostos a fim de melhorar o desempenho do algoritmo FPA padrão e, em geral, são propostas híbridas. El-sharkay (2015) propôs o algoritmo híbrido ABCFP combinando FPA com o Algoritmo de Colônia de Abelhas (ABC). Chakraborty Saha e Dutta (2014) propuseram a variante DE-FPA que combina o algoritmo de Evolução Diferencial (DE – *Differential Evolution*) com o FPA. Zhou Wang e Luo (2016) propuseram uma variante FPA baseada em oposição elite. A diversificação é aumentada usando a aprendizagem baseada em oposição elite, ao passo que a intensificação é implementada por meio de uma estratégia gulosa auto-adaptativa.

Neste artigo, uma variante do FPA é apresentada e consiste em utilizar o conceito de aprendizagem baseada em oposição, o qual é aplicado sobre a flor que apresenta o pior valor de *fitness*. A modificação proposta no FPA padrão tem como objetivo preservar a diversidade de soluções e evitar que o algoritmo fique "preso" em ótimos locais. Para avaliar o desempenho da variante proposta, diversos experimentos computacionais foram realizados na otimização de quatro funções de referência. Após os experimentos, no que diz respeito à qualidade e à estabilidade das soluções encontradas, observou-se um melhor desempenho da nova variante quando os seus resultados foram confrontados com o FPA padrão.

O trabalho está organizado como segue. Na Seção 2, é apresentado o algoritmo de otimização baseado no processo de polinização das flores. Na Seção 3, é apresentada uma nova variante do algoritmo de polinização das flores o qual aplica o conceito de aprendizagem baseada em oposição. Na Seção 4, os experimentos computacionais e os resultados obtidos são comentados. E, por fim, na Seção 5, são apontadas as considerações finais e as propostas de trabalhos futuros.

2. Algoritmo de Polinização das Flores

Na natureza, estima-se que existem mais de 250 mil tipos de plantas floríferas e que, aproximadamente, 80% de todas as espécies de plantas do planeta produzam flores [Glover 2007]. A principal função das flores é a reprodução que, em geral, ocorre quando agentes polinizadores como insetos, pássaros e morcegos transferem pólen.

A polinização pode ser abiótica ou biótica. Cerca de 90% das plantas floríferas são do tipo biótico, ou seja, o pólen é transferido por um polinizador. Já os outros 10%, são do tipo abiótico, em que o vento e a difusão na água são os responsáveis pela polinização. A polinização pode ocorrer via autopolinização ou polinização cruzada. A

autopolinização ocorre em uma flor bissexuada e não são necessários agentes polinizadores. Já a polinização cruzada é observada em flores unissexuadas em que os grãos de pólen são levados de uma flor para outra [Sousa *et al* 2002].

No Algoritmo de Polinização das Flores, assume-se que cada planta possui somente uma flor e que cada flor produz somente um gameta. O FPA implementa a polinização global e a polinização local. No primeiro tipo de polinização, os agentes carregam o pólen ao longo de grandes espaços de busca, o que garante a polinização e a reprodução da planta mais saudável. O indivíduo mais saudável é representado por g_*. Matematicamente, a polinização global pode ser representada por

$$x_i^{t+1} = x_i^t + L(x_i^t - g_*), \quad (1)$$

onde x_i^t é o pólen i (ou vetor solução x_i) na iteração t. Já g_* é a melhor solução encontrada entre todas, até o momento. O parâmetro L é a força de polinização, que é um valor gerado pela distribuição de Lévy [Viswanathan *et al* 2002]. Já a polinização local pode ser representada pela Equação 2, como segue

$$x_i^{t+1} = x_i^t + \varepsilon(x_j^t - x_k^t), \quad (2)$$

onde x_j^t e x_k^t são pólens de diferentes plantas da mesma espécie, na mesma iteração. Por fim, ε corresponde a uma distribuição normal uniforme no intervalo [0,1].

Na natureza, as flores vizinhas são mais susceptíveis à polinização local que as mais distantes. Por esse motivo, é utilizada uma probabilidade de proximidade p para escolher entre realizar a polinização global ou intensificar a polinização local. Na versão original do FPA, bem como na variante apresentada neste trabalho, o valor de p é definido em 0,8, o que favorece a busca local.

No Algoritmo 1, é apresentado o pseudocódigo do FPA. Ele é iniciado com a geração aleatória da população de flores (linha 01). Na linha 02, a melhor solução na população inicial é selecionada. A taxa de probabilidade é definida na linha 03 e é utilizada para escolher entre a polinização global ou a local. As instruções dentro da estrutura de repetição representam a evolução das flores ao longo do tempo e do espaço de busca. Assim, enquanto um determinado critério de convergência não é atingido (linhas 04–18), elas são executadas. Na linha 06, a taxa de probabilidade p é comparada com um valor gerado aleatoriamente e, quando ele for maior que p, será realizada uma busca global (linhas 07–08). Caso contrário, é realizada uma polinização local (linhas 10–12). Depois da polinização, se as novas soluções encontradas forem melhores, elas são atualizadas na população (linhas 14–15). Ao fim, a solução g_* que é a melhor solução da iteração, é selecionada (linha 17).

O FPA tem sido utilizado em diversas aplicações do mundo real. Na área de sistemas de potência, Abdelaziz *et al* (2016) utilizaram FPA para determinar o carregamento ideal de geradores em sistemas de energia elétrica. O método proposto pelos autores tem como objetivo minimizar simultaneamente o custo do consumo de combustível e o nível de emissão de poluentes, ao mesmo tempo que satisfaz a demanda de carga e algumas restrições operacionais.

No trabalho de Saxena e Kothari (2016), o FPA é usado para otimizar um arranjo de antenas lineares. A meta-heurística é aplicada ao arranjo linear para obter posições otimizadas das antenas a fim de atingir um padrão de arranjo com nível de lobo lateral mínimo e colocação de nulos profundos nas direções desejadas.

Uma nova variante do FPA foi apresentada por Zhou e Wang (2016) para resolver o problema do planejamento de caminho de veículos submarinos não-tripulados em duas e em três dimensões. A fim de aumentar a capacidade de intensificação do FPA, a meta-heurística Otimização por Enxame de Partículas (PSO – *Particle Swarm Optimization*) foi aplicada ao processo de busca local da nova variante. Na hibridização, também observou-se uma maior capacidade de busca global do FPA.

01: Gere a população inicial de flores/pólens (gametas) x_i (i= 1, 2, ..., n)
02: Ache a melhor solução g_* na população inicial
03: Defina a probabilidade $p \in [0,1]$
04: **enquanto** critério de convergência não for atingido **faca**
05: **para** *i* de 1 **ate** *n* **faca**
06: **se** $(rand > p)$ **entao**
07: Gere um L obedecendo a distribuição de Lévy
08: Faça polinização global usando $x_i^{t+1} = x_i^t + L(\mathrm{g}_* - x_i^t)$
09: **senao**
10: Gere ε a partir da distribuição uniforme entre [0,1]
11: Selecione aleatoriamente j e k entre todas as soluções
12: Faça polinização local usando $x_i^{t+1} = x_i^t + \varepsilon(\, x_j^t - x_k^t)$
13: **fimse**
14: Calcule as novas soluções
15: Se as novas soluções são melhores, atualize-as na população de flores
16: **fimpara**
17: Encontre a melhor solução atual g_*
18: **fimenquanto**

Algoritmo 1. Pseudocódigo do Algoritmo de Polinização das Flores.

3. FPA Modificado usando Aprendizagem Baseada em Oposição

Em geral, os algoritmos meta-heurísticos são iniciados com soluções aleatórias e, ao longo do tempo, eles procuram melhorá-las seguindo na direção da solução ótima. É comum que as soluções iniciais geradas aleatoriamente estejam distantes da ótima. O pior caso ocorre quando elas estão na posição oposta à da solução ótima. Em casos como esse, a otimização pode demandar muito tempo ou até mesmo inviabilizar o processo. Uma alternativa para isso seria realizar uma busca, simultaneamente, em todas as direções ou, simplesmente, realizar a busca na direção oposta. Essa última alternativa pode ser realizada usando o conceito de Aprendizagem Baseada em Oposição [Tizhoosh 2005].

Com a implementação da *Opposition-Based Learning* (OBL) é possível diminuir o tempo de computação e aumentar as chances de se encontrar uma solução mais próxima da ideal. Isso se dá porque sempre que uma solução x é buscada, é feita uma estimativa $\tilde{x}$. Assim, se $x \in \Re$ é um número real, definido dentro de um intervalo $x \in [a, b]$, então o oposto de x é definido por

$$\tilde{x} = a + b - x, \tag{3}$$

onde a é o limite inferior e b o superior. Considerando $P(x_1, \dots, x_n)$ como sendo um ponto em um sistema de coordenadas n-dimensional, $x_1, \dots, x_n \in \mathfrak{N}$ e $x_i \in [a_i, b_i]$, o oposto de P é definido pelas coordenadas $\tilde{x}_1, \dots, \tilde{x}_n$, de forma análoga, em que

$$\tilde{x_i} = a_i + b_i - x_i \qquad i = 1, \dots, n. \tag{4}$$

Este trabalho apresenta uma modificação no FPA padrão, a qual consiste na utilização do conceito de OBL. Nas linhas 17 – 20 do Algoritmo 2, a contribuição do trabalho é apresentada. Na linha 17, a flor que apresenta o pior valor de *fitness* é selecionada para aplicação da OBL. Nas linhas 18 – 20, o algoritmo verifica se, após essa aplicação, o valor de *fitness* da pior flor foi melhorado. Em caso positivo, o valor original é substituído pelo novo valor encontrado após a aplicação da OBL.

```
01: Gere a população inicial de flores/pólens (gametas) x_i (i= 1, 2, ..., n)
02: Ache a melhor solução g_* na população inicial
03: Defina a probabilidade p ∈ [0, 1]
04: enquanto critério de convergência não for atingido faca
05:     para i de 1 ate n faca
06:         se (rand > p) entao
07:             Gere um L obedecendo a distribuição de Lévy
08:             Faça polinização global usando x_i^{t+1} = x_i^t + L(g_* − x_i^t)
09:         senao
10:             Gere ε a partir da distribuição uniforme entre [0,1]
11:             Selecione aleatoriamente j e k entre todas as soluções
12:             Faça polinização local usando x_i^{t+1} = x_i^t + ε( x_j^t − x_k^t)
13:         fimse
14:         Calcule as novas soluções
15:         Se as novas soluções são melhores, atualize-as na população de flores
16:     fimpara
17:     Aplique OBL sobre a pior flor x_p
18:     se fitness (x̃_p) < fitness (x_p) entao
19:         x_p ← x̃_p
20:     fimse
21:     Encontre a melhor solução atual g_*
22: fimenquanto
```

Algoritmo 2. Pseudocódigo do Algoritmo de Polinização das Flores modificado.

A cada iteração, OBL é utilizada para aumentar a diversidade do algoritmo a fim de evitar que ele fique "preso" em ótimos locais. Quando se considera uma direção da busca e a posição oposta àquela da solução encontrada, é possível que a melhor solução seja alcançada em um tempo menor.

4. Experimentos Computacionais e Resultados Numéricos

Os experimentos computacionais foram executados em um computador que utiliza processador Intel Core i7 com 2,5 GHz de frequência, 8 GB de memória RAM e sistema operacional Windows 10 Pro, 64 bits. O FPA padrão e o FPA modificado foram

implementados na linguagem de programação Matlab R2013a e não foram utilizadas técnicas de processamento paralelo.

A fim de avaliar o desempenho dos algoritmos, foram escolhidas quatro funções de referência, bastante conhecidas na literatura dos algoritmos meta-heurísticos: Esfera, *Powell Sum*, Griewank e Csendes. Todas elas são aplicadas a problemas de minimização e são descritas a seguir. As funções Esfera e *Powell Sum* são unimodais e, normalmente, são usadas para testar a habilidade do algoritmo em buscas locais. Já Griewank e Csendes possui vários mínimos locais e são utilizadas para verificar a habilidade do algoritmo em buscas globais.

- Função Esfera – é caracterizada por ser simples, convexa e unimodal:

$$f_1(x) = \sum_{i=1}^{d} x_i^2 \tag{5}$$

- Função *Powell Sum* – é uma função unimodal definida de acordo com a formulação abaixo:

$$f_2(x) = \sum_{i=1}^{d} |x_i|^{i+1} \tag{6}$$

- Função Griewank – possui vários mínimos locais regularmente distribuídos:

$$f_3(x) = \frac{1}{4000} \sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{7}$$

- Função Csendes – é uma função multimodal definida como segue:

$$f_4(x) = \sum_{i=1}^{d} x_i^6 \left[2 + sen\left(\frac{1}{x_i}\right)\right] \tag{8}$$

Na Tabela 1, são apresentadas as fronteiras do espaço de busca, a região de inicialização do algoritmo, a solução global e a solução ótima, para cada uma das quatro funções de referência.

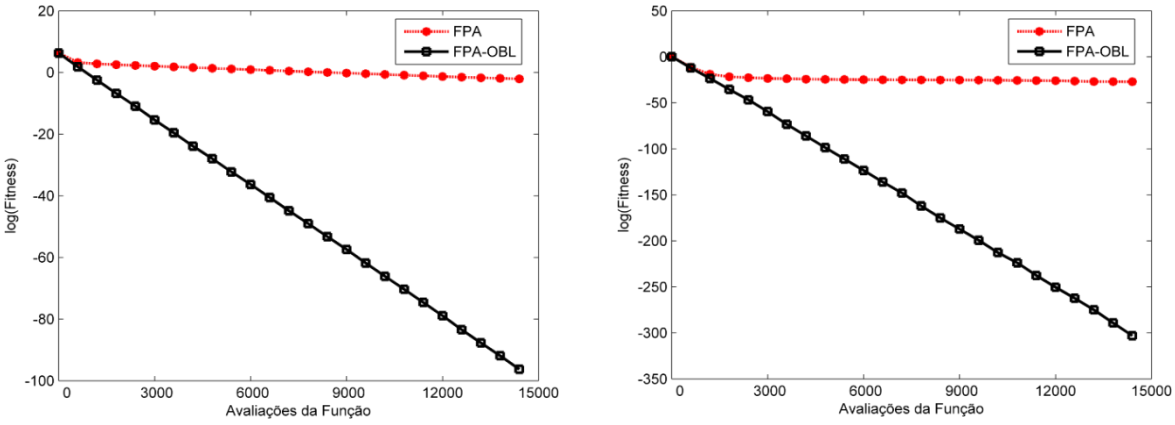
Tabela 1. Características das funções de referência.

Função	Espaço de busca	Faixa de Inicialização	Solução Global	Solução Ótima
f_1	$-5{,}12 \leq x_i \leq 5{,}12$	$2{,}06 \leq x_i \leq 5{,}12$	$f(x^*) = 0$	$x^* = (0, \ldots, 0)$
f_2	$-100 \leq x_i \leq 100$	$50 \leq x_i \leq 100$	$f(x^*) = 0$	$x^* = (0, \ldots, 0)$
f_3	$-600 \leq x_i \leq 600$	$300 \leq x_i \leq 600$	$f(x^*) = 0$	$x^* = (0, \ldots, 0)$
f_4	$-1 \leq x_i \leq 1$	$0.5 \leq x_i \leq 1$	$f(x^*) = 0$	$x^* = (0, \ldots, 0)$

Para realizar os experimentos, são definidas 3 configurações. Elas consistem na variação da dimensionalidade das funções e do número máximo de iterações. Apenas a quantidade de flores é fixada em 30. A dimensionalidade varia em 30, 60 e 90 dimensões. Para 30 dimensões, o número máximo de iterações é 5.000; para 60 dimensões, são 10.000 iterações e; para 90 dimensões, o número máximo de iterações é 15.000. Após a realização de 30 execuções independentes, são registrados a melhor solução, a pior solução, a média do valor de *fitness* e o desvio padrão.

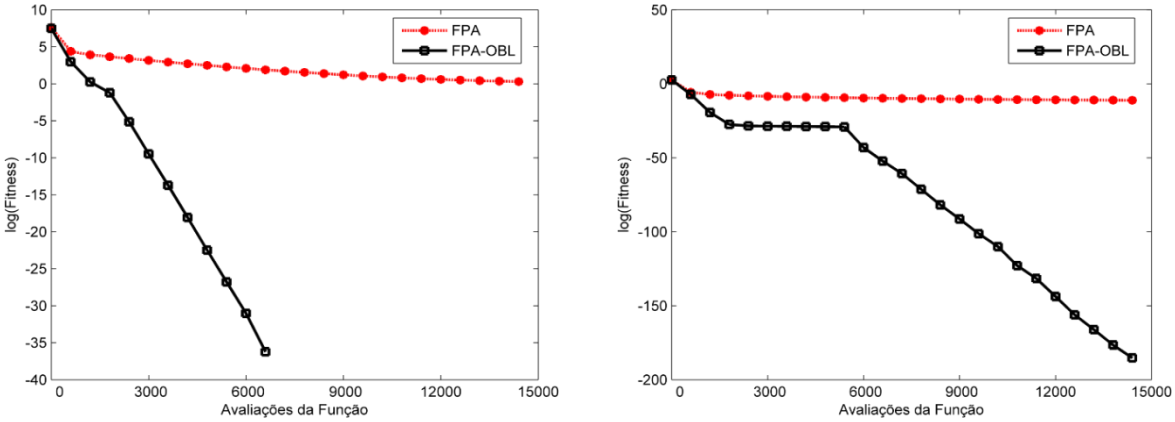
As Figuras 1(a), 1(b), 2(a) e 2(b) apresentam o comportamento de convergência dos algoritmos para as quatro funções de referência f_1, f_2, f_3 e f_4, respectivamente, durante 15.000 iterações, em 90 dimensões. Na otimização da função f_1, como visto na Figura 1(a), é observado um suave comportamento de estagnação do algoritmo FPA padrão, no entanto, o FPA modificado se apresenta ativo durante as 15.000 iterações. Uma situação semelhante a essa é apresentada na Figura 1(b), em que o FPA modificado supera o FPA padrão quando são consideradas a velocidade de convergência e a qualidade das soluções. Vale salientar que, apesar do desempenho satisfatório que o FPA modificado obteve, as soluções ótimas globais não foram encontradas nas funções f_1 e f_2.

Na Figura 2(a), função f_3, o FPA padrão novamente apresenta um desempenho inferior ao do FPA modificado. Durante todo o processo de busca, é observado que o algoritmo padrão tem uma baixa velocidade de convergência, apresentando poucas melhorias significativas entre as iterações 12.000 – 15.000. Por outro lado, o FPA modificado apresenta um comportamento de convergência bastante ativo e, próximo da iteração 7.000, atinge a solução ótima. Na Figura 2(b), quando comparado com o FPA modificado, o FPA padrão apresentou um comportamento de estagnação, além de soluções insatisfatórias e velocidade de convergência lenta. Por outro lado, a variante proposta estagnou, aproximadamente, entre as iterações 1.500 – 5.800, no entanto, após esse momento, ela continuou com um comportamento ativo de convergência. Apesar do desempenho satisfatório em relação ao FPA padrão, a variante proposta não encontrou a solução ótima global da função f_4 durante as 15.000 iterações. A Tabela 2 apresenta os resultados numéricos dos experimentos computacionais realizados.



(a) Função Esfera (f_1) **(b) Função *Powell Sum* (f_2)**

Figura 1. FPA padrão e FPA modificado, durante 15.000 iterações, em 90 dimensões.



(a) Função Griewank (f_3) **(b) Função Csendes (f_4)**

Figura 2. FPA padrão e FPA modificado, durante 15.000 iterações, em 90 dimensões.

Tabela 2. Resultados dos algoritmos nas funções de referência avaliadas.

Fun	Dim	Iter	FPA padrão			FPA modificado com OBL		
			Média (Desvio)	Melhor	Pior	Média (Desvio)	Melhor	Pior
f_1	30	5.000	5,31E-03 (3,46E-03)	4,70E-04	1,22E-02	**9,97E-16** (9,34E-16)	6,43E-17	3,72E-15
	60	10.000	3,08E-02 (1,98E-02)	4,95E-03	9,16E-02	**5,35E-30** (1,26E-29)	7,36E-32	5,03E-29
	90	15.000	0,10 (7,56E-02)	3,11E-02	0,37	**2,54E-44** (4,18E-44)	6,80E-47	1,51E-43
f_2	30	5.000	1,48E-13 (7,33E-13)	1,40E-19	4,02E-12	**9,37E-46** (2,02E-45)	1,09E-49	7,89E-45
	60	10.000	2,19E-10 (1,19E-09)	2,63E-20	6,54E-09	**2,99E-91** (9,97E-91)	3,72E-35	5,35E-90
	90	15.000	1,54E-12 (3,60E-12)	6,78E-19	1,21E-11	**6,20E-137** (1,60E-136)	5,61E-142	6,92E-136
f_3	30	5.000	0,73 (0,24)	0,30	1,07	**9,20E-09** (3,34E-08)	1,21E-13	1,72E-09
	60	10.000	1,14 (0,14)	0,93	1,70	**0** (0)	0	0
	90	15.000	1,27 (0,15)	1,01	1,72	**0** (0)	0	0
f_4	30	5.000	3,76E-07 (3,69E-07)	1,60E-09	1,52E-06	**4,04E-33** (2,21E-32)	7,33E-42	1,21E-31
	60	10.000	4,17E-06 (3,15E-06)	1,00E-06	1,27E-05	**1,67E-66** (9,18E-66)	5,43E-79	5,03E-65
	90	15.000	1,30E-05 (1,00E-05)	4,55E-06	5,22E-05	**1,73E-86** (9,48E-86)	2,30E-116	5,19E-85

Em todas as funções avaliadas, como se observa na Tabela 2, os valores médios das soluções encontradas pelo algoritmo FPA modificado são melhores que os do FPA padrão. Na função f_3, o FPA modificado encontrou a solução ótima global. Em relação ao desvio padrão, que mostra a variação existente em relação à média, o algoritmo FPA modificado obteve valores melhores em todos os experimentos. A superioridade do FPA modificado também pode ser evidenciada a partir da análise dos piores e dos melhores resultados, em que os piores valores do FPA modificado são menores que os dos melhores do FPA padrão.

5. Conclusões

Este trabalho apresentou uma variante da meta-heurística baseada no processo de polinização das flores. A variante consiste na combinação do FPA padrão com a aprendizagem baseada em oposição e tem o objetivo de gerar diversidade de soluções e evitar que o processo de convergência estagne em ótimos locais.

O desempenho do FPA padrão e do FPA modificado foram comparados por meio de vários experimentos computacionais, realizados utilizando quatro funções de referência: Esfera, *Powell Sum*, Griewank e Csendes. Após os experimentos, observou-se que o algoritmo FPA modificado obteve resultados superiores ao algoritmo padrão no que diz respeito à qualidade e à estabilidade das soluções encontradas. Isso ocorreu porque a aprendizagem baseada em oposição, além de diminuir o tempo de computação, preservou a diversidade do algoritmo, tornando o processo de convergência ativo.

Como trabalhos futuros, pretende-se avaliar o desempenho do FPA modificado em aplicações reais da área de telecomunicações. Além disso, também se pretende combinar a variante apresentada com um operador de mutação conhecido como Cauchy a fim de aumentar ainda mais a diversidade do algoritmo.

Referências

ABDELAZIZ, A. Y.; ALI, E. S.; ELAZIM, SM Abd. Implementation of flower pollination algorithm for solving economic load dispatch and combined economic emission dispatch problems in power systems. **Energy**, v. 101, p. 506-518, 2016.

CHAKRABORTY, Dwaipayan; SAHA, Sankhadip; DUTTA, Oindrilla. DE-FPA: A hybrid differential evolution-flower pollination algorithm for function minimization. In: **High Performance Computing and Applications (ICHPCA), 2014 International Conference on**. IEEE, 2014. p. 1-6.

CUEVAS, Erik et al. A swarm optimization algorithm inspired in the behavior of the social-spider. **Expert Systems with Applications**, v. 40, n. 16, p. 6374-6384, 2013.

EL-SHARKAY. A hybrid Bee colony algorithm for solving unconstrained optimization problems. **International Journal of Engineering Trends and Technology (IJETT)**, vol. 28, no. 9, pp. 480-484, 2015.

GLOVER, Beverley Jane. **Understanding flowers and flowering: an integrated approach**. Oxford, UK: Oxford University Press, 2007.

KENNEDY, J.; EBERHART, R. **Particle swarm optimization**. In Proceedings of IEEE International Conference on Neural Networks, 1995.

SAXENA, Prerna; KOTHARI, Ashwin. Linear antenna array optimization using flower pollination algorithm. **SpringerPlus**, v. 5, n. 1, p. 1, 2016.

SOUSA, Vanessa Ribeiro et al. Biologia floral do cerrado: polinização e floração. 2002.

TIZHOOSH, Hamid R. Opposition-Based Learning: A New Scheme for Machine Intelligence. In: **CIMCA/IAWTIC**. 2005. p. 695-701

VIALI, L. **Métodos quantitativos: estatística e matemática aplicadas.** 2011.

VISWANATHAN, G. M. et al. Lévy flight random searches in biological phenomena. **Physica A: Statistical Mechanics and Its Applications**, v. 314, n. 1, p. 208-213, 2002.

YANG, Xin-She. A new metaheuristic bat-inspired algorithm. In: **Nature inspired cooperative strategies for optimization (NICSO 2010)**. Springer Berlin Heidelberg, 2010. p. 65-74.

YANG, Xin-She. Firefly algorithms for multimodal optimization. In: **International Symposium on Stochastic Algorithms**. Springer Berlin Heidelberg, 2009. p. 169-178.

YANG, Xin-She. Flower pollination algorithm for global optimization. In: **International Conference on Unconventional Computing and Natural Computation**. Springer Berlin Heidelberg, 2012. p. 240-249.

ZHOU, Yongquan; WANG, Rui. An Improved Flower Pollination Algorithm for Optimal Unmanned Undersea Vehicle Path Planning Problem. **International Journal of Pattern Recognition and Artificial Intelligence**, v. 30, n. 04, p. 1659010, 2016.

ZHOU, Yongquan; WANG, Rui; LUO, Qifang. Elite opposition-based flower pollination algorithm. **Neurocomputing**, v. 188, p. 294-310, 2016.