

Meta-heurística Inspirada na Bioluminescência dos Vaga-lumes usando Aprendizagem Baseada em Oposição Elite

Izabele V. O. Leite¹, Marcos H. F. Marcone² e Fábio A. P. Paiva³

¹ Departamento de Engenharia Química
Universidade Federal do Rio Grande do Norte (UFRN) – Natal – RN – Brasil

² Escola de Ciências e Tecnologia
Universidade Federal do Rio Grande do Norte (UFRN) – Natal – RN – Brasil

³ Diretoria Acadêmica
Instituto Federal do Rio Grande do Norte (IFRN) – Parnamirim – RN – Brasil

{izaleite11, marcosmarcone48}@hotmail.com, fabio.procopio@ifrn.edu.br

Abstract. *Stagnation into local optimum is a frequent problem of metaheuristic methods, including ones nature-inspired such as Firefly Algorithm (FA). Although several approaches have been proposed, the problem remains an open question. This paper presents a FA variant that uses Elite Opposition-Based Learning (EOBL) and it is called FA-EOBL. The variant aims to generate diversity and increase the velocity of convergence of the original FA. Several experiments were performed with 12 benchmark functions. Generally, they are used to validate and compare new optimization algorithms. The results show the superiority of the FA-EOBL when it is compared to original FA.*

Resumo. *A estagnação em ótimo local é um problema frequente dos métodos meta-heurísticos, incluindo os inspirados na natureza como o Algoritmo do Vaga-lume (FA). Embora várias abordagens tenham sido propostas, o problema continua uma questão em aberto. Este trabalho apresenta uma variante do FA que utiliza Aprendizagem Baseada em Oposição Elite (EOBL), denominada FA-EOBL. A variante objetiva gerar diversidade e aumentar a velocidade de convergência do FA original. Diversos experimentos foram realizados com 12 funções de referência. Em geral, elas são usadas para validar e comparar novos algoritmos de otimização. Os resultados mostram a superioridade da FA-EOBL quando ela é comparada com o FA original.*

1. Introdução

A inteligência de enxames está relacionada ao comportamento coletivo de vários agentes naturais que interagem entre si e que seguem algumas regras. Um agente, quando observado individualmente, não pode ser considerado inteligente, porém um sistema com vários agentes apresenta um comportamento auto-organizável de inteligência coletiva.

Os algoritmos baseados em inteligência de enxames consistem em um conjunto de meta-heurísticas que se baseia em alguns seres vivos cujos comportamentos emergentes podem resultar em uma capacidade de resolver problemas complexos [Silva and Bastos Filho 2015]. *Firefly Algorithm* (FA), por exemplo, tem sido empregado em cenários reais de otimização [Basu and Mahanti 2011, Chatterjee et al. 2012,

Horng et al. 2012]. FA se baseia no processo de bioluminescência dos vaga-lumes. No algoritmo, existem dois pontos importantes: formulação da intensidade da luz e mudança da atratividade. A capacidade de atração de um vaga-lume é proporcional ao seu brilho. Assim os vaga-lumes menos brilhantes são atraídos pelos que apresentam o maior brilho.

Um problema enfrentado pelo FA é a estagnação em ótimos locais. Vários autores propuseram abordagens para minimizar esse problema como [Yu et al. 2013] com a variante *Self-Adaptive Step Firefly Algorithm* (SASFA), em que a configuração de passo baseia-se no histórico de informações e na situação de cada vaga-lume. Já [Goel and Panchal 2014] propuseram o *New Modified Firefly Algorithm* (NMFA) que consiste na definição dinâmica de valores do parâmetro α , definido na Equação (3), que decai linearmente, a cada iteração, de α_0 até α_∞ .

Neste trabalho, uma variante FA é proposta a fim de aumentar a velocidade de convergência do FA e, como consequência, melhorar o seu desempenho. A variante é denominada FA-EOBL e combina o FA original com a Aprendizagem Baseada em Oposição Elite (*Elite Opposition-Based Learning* – EOBL) [Zhou et al. 2012]. A estratégia é aplicada sobre cada vaga-lume de uma iteração t . O desempenho do novo algoritmo é avaliado por meio de experimentos computacionais com 12 funções de referência. Ao fim dos experimentos, foi observada a superioridade do algoritmo FA-EOBL em relação ao FA.

O trabalho está organizado da seguinte forma: na Seção 2, é apresentada uma breve introdução sobre a meta-heurística FA e a aprendizagem baseada em oposição elite; na Seção 3, a nova variante é apresentada para melhorar a performance do FA original. Na Seção 4, os experimentos são apresentados e os resultados são discutidos. Por fim, a Seção 5 apresenta as conclusões e os trabalhos futuros.

2. Fundamentação Teórica

Aqui são apresentadas algumas informações úteis para o entendimento da nova variante que é apresentada na Seção 3, como a meta-heurística inspirada na bioluminescência dos vaga-lumes e a aprendizagem baseada em oposição elite.

2.1. Meta-heurística Inspirada no Processo de Bioluminescência dos Vaga-lumes

O processo de bioluminescência é o principal elemento usado pelos vaga-lumes para a produção de *flashes*. Sabe-se que um vaga-lume utiliza *flashes* para 1) comunicar-se e atrair outros vaga-lumes, 2) atrair eventuais presas e; 3) proteger-se de potenciais predadores. Os *flashes* podem ser modelados e associados a uma função objetivo a ser otimizada, o que permite a formalização de novas meta-heurísticas como o Algoritmo do Vaga-lume (*Firefly Algorithm* – FA) [Yang 2009], que se baseia no processo de bioluminescência desses insetos coleópteros.

Para simplificar a modelagem do FA, [Yang 2009] considera três regras: 1) todos os vaga-lumes têm um único sexo e serão atraídos um pelo outro; 2) a capacidade de atração é proporcional ao brilho de cada vaga-lume e, portanto, o menos brilhante será movido em direção do mais brilhante; 3) o brilho de um vaga-lume é determinado pelo formato da função objetivo.

Baseado nessas três regras, os passos do FA podem ser sumarizados como mostrado no Algoritmo 1. Inicialmente, a população de vaga-lumes é gerada de forma

aleatória (linha 1). Em seguida, a função objetivo é calculada (linha 2). Na linha 3, o coeficiente de absorção do meio é definido. Caso a intensidade de luz do vaga-lume i seja menor que a do vaga-lume j , i é atraído por j (linhas 7–9). Na linha 10, o fator de atratividade é calculado pela equação $\beta(r) = \beta_0 e^{-\gamma r^2}$. Na linha 11, as novas soluções encontradas são avaliadas e as suas respectivas intensidades de luz são calculadas. Por fim, cada vaga-lume é ordenado de acordo com o seu brilho e aquele que apresentar a maior intensidade de luz é escolhido como a melhor solução.

Algoritmo 1 Pseudocódigo do Algoritmo do Vaga-lume original

- 1: Gere a população de vaga-lumes $x_i = (1, 2, \dots, n)$
 - 2: Determine a intensidade de luz I_i em x_i usando $f(x_i)$
 - 3: Defina o coeficiente de absorção de luz γ
 - 4: **enquanto** critério de convergência não for atingido **faça**
 - 5: **para** $i \leftarrow 1$ **até** n (para os N vaga-lumes) **faça**
 - 6: **para** $j \leftarrow i$ **até** n (para os N vaga-lumes) **faça**
 - 7: **se** $I_i < I_j$ **então**
 - 8: Mova $vagalume_i$ na direção do $vagalume_j$
 - 9: **fim se**
 - 10: Varie a atratividade, a partir da distância r , usando $\exp[-\gamma r^2]$
 - 11: Avalie as novas soluções e atualize as respectivas intensidades de luz
 - 12: **fim para**
 - 13: **fim para**
 - 14: Classifique os vaga-lumes e encontre o melhor
 - 15: **fim enquanto**
-

A atratividade de um vaga-lume é proporcional à intensidade de luz vista pelos vaga-lumes adjacentes. Assim, a sua atratividade β pode ser definida por $\beta = \beta_0 e^{-\gamma r^2}$, onde r é a distância entre os vaga-lumes, β_0 é a atratividade quando $r = 0$. Como é mais rápido e fácil calcular $1/(1 + \gamma r^2)$ do que uma função exponencial, se necessário, o cálculo de β pode ser substituído por

$$\beta = \frac{\beta_0}{1 + \gamma r^2}. \quad (1)$$

O parâmetro γ , definido na Equação (1), caracteriza a variação da atratividade e o seu valor é essencial para determinar a velocidade de convergência e o comportamento do FA. Em tese, $\gamma \in [0, \infty)$, porém, na prática, $\gamma = O(1)$ é determinado pelo tamanho característico Γ do sistema a ser otimizado. Assim, em muitas aplicações, γ varia de 0,01 a 100.

A distância entre dois vaga-lumes i e j , nas posições x_i e x_j , respectivamente, é dada por

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}, \quad (2)$$

onde $x_{i,k}$ é o k -ésimo componente da coordenada espacial x_i do i -ésimo vaga-lume e d é o número de dimensões. Já o movimento de um vaga-lume i , em direção do vaga-lume

mais brilhante j , é definido por

$$x_i = x_i + \beta_0 e^{-\gamma_{ij}^2} (x_j - x_i) + \alpha (\text{rand} - \frac{1}{2}), \quad (3)$$

onde o primeiro termo é a posição atual do vaga-lume i , o segundo termo representa a atratividade entre os vaga-lumes i e j e o terceiro termo é um valor aleatório, em que α é o parâmetro que define a aleatoriedade e rand é um valor aleatório, distribuído uniformemente no intervalo $[0, 1]$. Vale salientar que a atratividade β pode ser representada pela Equação (1).

2.2. Aprendizagem Baseada em Oposição Elite

Em geral, os algoritmos meta-heurísticos são iniciados com soluções aleatórias e, ao longo do tempo, eles procuram melhorá-las seguindo em direção da solução ótima. É comum que, na inicialização do algoritmo, essas soluções estejam distantes da ótima. O pior caso ocorre quando elas estão na posição oposta à da solução considerada ótima.

Uma alternativa para essa situação é buscar, simultaneamente, em todas as direções ou, simplesmente, na direção oposta. Para lidar com esta situação, a Aprendizagem Baseada em Oposição Elite (EOBL – *Elite Opposition-Based Learning*) [Zhou et al. 2012] pode ser usada. Antes de apresentar EOBL, o conceito de Aprendizagem Baseada em Oposição (OBL – *Opposition-Based Learning*) [Tizhoosh 2005] será explanado. A seguir, são definidos Número Oposto e Ponto Oposto.

Definição 1: Dado $x \in \mathbb{R}$, no intervalo $x \in [a, b]$, o número oposto \tilde{x} é definido na Equação (4).

$$\tilde{x} = a + b - x. \quad (4)$$

Definição 2: Dado $P = (x_1, x_2, \dots, x_n)$ como sendo um ponto no espaço n -dimensional, com $x_1, x_2, \dots, x_n \in \mathbb{R}$ e $x_i \in [a_i, b_i], \forall i \in \{1, 2, \dots, n\}$. Assim, o ponto oposto $\tilde{P} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$ é definido pelos seus componentes:

$$\tilde{x}_i = a_i + b_i - x_i. \quad (5)$$

Neste trabalho, o vaga-lume com o maior brilho é considerado o elite. Assim, supondo que o vaga-lume elite seja $X_e = (x_{e1}, x_{e2}, \dots, x_{en})$, a solução elite para o vaga-lume $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ pode ser definida como $\hat{X}_i = (\hat{x}_{i1}, \hat{x}_{i2}, \dots, \hat{x}_{in})$ e obtida pela Equação (6).

$$\hat{x}_{ij} = \delta \cdot (da_j + db_j) - x_{ej}, i = 1, 2, \dots, N; j = 1, 2, \dots, n \quad (6)$$

onde N é a quantidade de vaga-lumes, n é a dimensão de X , $\delta \in [0, 1]$ é um coeficiente generalizado que é usado para controlar a magnitude da oposição. $x_{ej} \in [a_i, b_i]$ e $[da_j, db_j]$ é o limite dinâmico, definido por

$$da_j = \min(x_{ij}), \quad db_j = \max(x_{ij}). \quad (7)$$

É possível que uma solução transformada por meio da oposição elite “salte” para fora do intervalo $[a_i, b_i]$. Se isso acontecer, a Equação (8) é usada para reiniciar \hat{x}_{ij} .

$$\hat{x}_{ij} = \text{rand}(da_j, db_j), \quad \hat{x}_{ij} < da_j \ || \ \hat{x}_{ij} > db_j. \quad (8)$$

3. Meta-heurística FA usando Aprendizagem Baseada em Oposição Elite

Uma das vantagens encontradas no FA é a sua fácil implementação. No entanto, a sua velocidade de convergência é considerada lenta e, muitas vezes, o algoritmo “cai” em ótimos locais [Goel and Panchal 2014].

A fim de adicionar um mecanismo que possibilite o FA gerar diversidade de soluções candidatas, este trabalho propõe uma nova variante FA que implementa o conceito de EOBL a fim de aumentar a velocidade de convergência do algoritmo original. A contribuição do trabalho é apresentada no pseudocódigo do Algoritmo 2.

Algoritmo 2 Pseudocódigo do Algoritmo do Vaga-lume usando EOBL

```

1: função EOBL( $vag^t$ )
2:   Obtenha limite dinâmico usando Equação (7)
3:   para  $k \leftarrow 1$  até  $dim_{vag^t}$  faça
4:     Calcule oposto usando Equação (6)
5:     se oposto está fora do espaço de busca então
6:       Reinicie solução elite usando Equação (8)
7:     fim se
8:   fim para
9:   retorne Solução baseada em oposição elite
10: fim função
11: Idem às linhas 1–3 do Algoritmo 1
12: enquanto critério de convergência não for atingido faça
13:   para  $i \leftarrow 1$  até  $n$  (para os  $N$  vaga-lumes) faça
14:      $vag^t \leftarrow$  EOBL( $vag^t$ )
15:     para  $j \leftarrow i$  até  $n$  (para os  $N$  vaga-lumes) faça
16:       Idem às linhas 7–11 do Algoritmo 1
17:     fim para
18:   fim para
19:   Classifique os vaga-lumes e encontre o melhor
20: fim enquanto

```

Na linha 14, onde EOBL é aplicada no vaga-lume corrente da iteração t , a aprendizagem baseada em oposição elite é executada pela rotina EOBL, sumarizada nas linhas 1 – 10 do Algoritmo 2. Na linha 2, o limite dinâmico de busca é obtido. Para cada dimensão do vaga-lume, calcula-se o seu correspondente oposto elite. Porém, pode ocorrer de a solução transformada ser posicionada fora do limite do espaço de busca. Caso isso ocorra, é necessário reiniciar a solução, conforme linha 6. Ao fim da geração do oposto elite, de cada uma das dimensões do vaga-lume, a nova solução baseada em oposição elite é retornada. As demais linhas do Algoritmo 2 são semelhantes às do FA original.

4. Experimentos Computacionais

Esta seção apresenta as funções de referência usadas para validar o algoritmo proposto. Também é apresentada a configuração dos valores atribuídos aos parâmetros. Finalmente, são discutidos as simulações e os resultados encontrados.

4.1. Funções de Referência

Na realização dos experimentos, foram escolhidas 12 funções de referência, apresentadas na Tabela 1, que são aplicadas em problemas de minimização e que já foram utilizadas em vários estudos de FA [Yu et al. 2013, Goel and Panchal 2014, Kalra and Arora 2016].

Tabela 1. Funções de referência utilizadas para avaliar FA e FA-EOBL.

F	Fórmula	Espaço de Busca	Ótima
f_1	$f_1(x) = \sum_{i=1}^d x_i^2$	$-100 \leq x_i \leq 100$	0
f_2	$f_2(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$-30 \leq x_i \leq 30$	0
f_3	$f_3(x) = \sum_{i=1}^{d-1} (x_i + 0.5)^2$	$-100 \leq x_i \leq 100$	0
f_4	$f_4(x) = \sum_{i=1}^d x_i^4$	$-100 \leq x_i \leq 100$	0
f_5	$f_5(x) = \sum_{i=1}^d x_i ^{i+1}$	$-500 \leq x_i \leq 500$	0
f_6	$f_6(x) = x_i^2 + \sum_{i=2}^d x_i^2$	$-10 \leq x_i \leq 10$	0
f_7	$f_7(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{d}} \sum_{i=1}^d x_i^2\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + \exp(1)$	$-32 \leq x_i \leq 32$	0
f_8	$f_8(x) = \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i) + 10]$	$-5.12 \leq x_i \leq 5.12$	0
f_9	$f_9(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-600 \leq x_i \leq 600$	0
f_{10}	$f_{10}(x) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^d x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^d x_i^2}$	$-100 \leq x_i \leq 100$	0
f_{11}	$f_{11}(x) = \sum_{i=1}^d x_i^2 + \left(\sum_{i=1}^d 0.5ix_i\right)^2 + \left(\sum_{i=1}^d 0.5ix_i\right)^4$	$-5 \leq x_i \leq 10$	0
f_{12}	$f_{12}(x) = \sum_{i=1}^d x_i \sin(x_i) + 0.1x_i $	$-10 \leq x_i \leq 10$	0

Para cada uma delas, a fórmula, o espaço de busca e a solução ótima são apresentados na Tabela 1. Essas funções são classificadas em dois grupos:

1. Grupo 1 – formado por funções unimodais que, geralmente são usadas para testar a capacidade do algoritmo em buscas locais. As funções unimodais usadas nos experimentos são: Esfera (f_1), Rosenbrock (f_2), *Step* (f_3), Schumer Steiglitz (f_4), *Powell Sum* (f_5) e *Cigar* (f_6).
2. Grupo 2 – formado por funções multimodais que possuem vários mínimos locais. Elas são usadas para verificar a habilidade do algoritmo para “escapar” de

ótimos locais. São elas: Ackley (f_7), Rastrigin (f_8), Griewank (f_9), Salomon (f_{10}), Zakharov (f_{11}) e Alpine (f_{12}).

4.2. Simulações e Resultados

Todas as rotinas foram implementadas em MATLAB R2013a. Os experimentos foram executados em um computador que usa um processador Intel Core i7, com 2,4 GHz de frequência, 8 GB de memória RAM e *Windows 10 Home Single Language* (64 bits).

A performance da variante FA–EOBL é avaliada comparando os seus resultados com os do FA, usando 12 funções de referência apresentadas. Em todos os experimentos, são realizadas 20 execuções independentes. O tamanho da população de vaga-lumes é fixado em $N = 20$, o número de dimensões em $D = 30$ e a quantidade de iterações em 1.000. Os valores atribuídos aos parâmetros dos algoritmos foram os mesmos usados por [Kalra and Arora 2016]: parâmetro de aleatoriedade α definido em 0.1, coeficiente de absorção do ar γ igual a 1.0 e intensidade de luminosidade mínima $\beta_{min} = 0.2$. Neste trabalho, o teste de Wilcoxon é utilizado para verificar se os resultados da variante são estatisticamente significativos quando comparados aos do FA, isto é, se *p-value* é menor que o nível de significância determinado. Nos experimentos, o nível de significância de 0.05 é aplicado a fim de avaliar os resultados encontrados pela variante e pelo FA.

A Figura 1 apresenta o comportamento de convergência médio do FA original e da variante FA–EOBL para as doze funções de referência avaliadas. Os gráficos das subfiguras 1(a) – 1(f) correspondem ao grupo de funções unimodais. Em 1(a), FA–EOBL apresenta uma estagnação entre as iterações 690–830 e, novamente, perto da iteração 960, porém a sua velocidade de convergência e os seus resultados superam o FA original. A subfigura 1(b) mostra que, até as primeiras iterações, FA–EOBL apresenta convergência mais rápida que FA, todavia perto da iteração 90, a variante estagna e permanece assim até o fim das 1.000 iterações. Rosenbrock (f_2) foi a função em que os algoritmos apresentaram o pior desempenho, embora FA–EOBL tenha superado FA. Na subfigura 1(c), FA–EOBL encontra a solução ótima da função *Step* (f_3) perto da iteração 90 e FA próximo da iteração 400. Pelo fato de ambos terem encontrado a solução ótima da função, naturalmente, o teste de Wilcoxon não indica significância estatística, embora a velocidade de convergência da variante supere a do FA, conforme Tabela 2. Por fim, as subfiguras 1(d), 1(e) e 1(f) mostram que FA–EOBL possui um comportamento bastante ativo durante as 1.000 iterações, enquanto que FA apresenta uma convergência mais lenta.

Nos gráficos das subfiguras 1(g) – 1(l) são apresentados os experimentos realizados nas funções multimodais. Na subfigura 1(g), há um comportamento ativo dos dois algoritmos, apesar de FA–EOBL ser mais rápida. É observado também que FA–EOBL apresenta uma ligeira estagnação entre as iterações 810–910, mas isso não comprometeu, significativamente, o desempenho da variante. A subfigura 1(h) mostra a estagnação do FA desde a iteração 400, ao passo que FA–EOBL apresenta um comportamento ativo e uma rápida estagnação entre as iterações 790–980. A subfigura 1(i) não apresenta pontos de estagnação de nenhum dos dois algoritmos, mas observa-se claramente que FA–EOBL é mais rápida. Na subfigura 1(j), FA estagna na iteração 300 e permanece assim até o fim do número máximo de iterações, enquanto FA–EOBL, perto da iteração 600, apresenta um “pico” no valor de *fitness*, isto é, um valor pior dos que vinham sendo encontrados pelo algoritmo até então. No entanto, nas iterações seguintes, o valor de *fitness* volta a ser

melhorado. Na subfigura 1(k), nas primeiras iterações, FA não apresenta melhorias significativas no valor de *fitness* e, a partir da iteração 450 estagna, ao passo que FA–EOBL permanece bastante ativo durante as 1.000 iterações. Na última subfigura, FA apresenta uma velocidade de convergência muito lenta, quando comparada ao da FA–EOBL, que apresenta 2 trechos de estagnação: entre as iterações 410–600 e entre as iterações 500–1.000. Em nenhum dos experimentos realizados nas funções multimodais a solução global foi encontrada. Porém, observa-se claramente a superioridade da variante quando comparada ao FA.

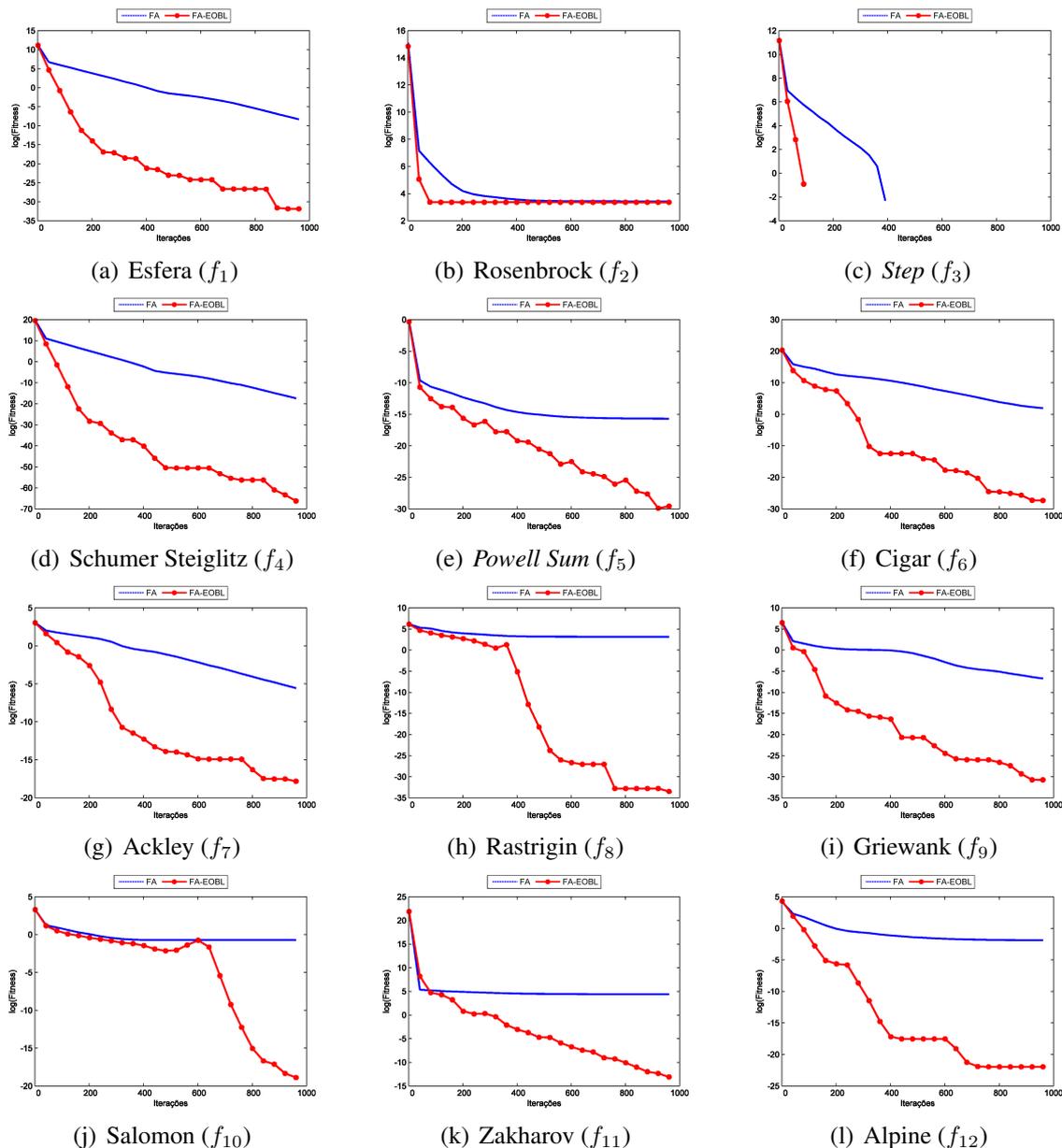


Figura 1. Convergência média dos algoritmos nas funções avaliadas.

A Tabela 2 apresenta os resultados numéricos da comparação entre FA e FA-EOBL. Para cada função avaliada, é mostrada a melhor solução, a pior solução, a média das soluções, o desvio padrão e o *p-value* resultante do Teste de Wilcoxon. Ainda na

Tabela 2. Comparação entre FA e FA-EOBL para as 12 funções avaliadas.

F	FA			FA-EOBL			p-Value
	Pior	Melhor	Média (Desvio)	Pior	Melhor	Média (Desvio)	
f_1	2.58e-04	4.82e-05	1.23e-04 (4.78e-05)	1.01e-14	2.27e-46	1.33e-15 (2.82e-15)	1.90e-06
f_2	86.94	25.49	30.69 (13.28)	28.70	28.70	28.70 (3.91e-04)	8.30e-03
f_3	0	0	0 (0)	0	0	0 (0)	1
f_4	2.44e-08	8.06e-10	7.19e-09 (6.40e-09)	1.29e-29	7.32e-115	1.71e-30 (3.79e-30)	1.90e-06
f_5	5.68e-07	2.21e-08	1.51e-07 (1.49e-07)	2.09e-13	5.02e-16	4.53e-14 (6.60e-14)	1.90e-06
f_6	24.08	0.85	5.50 (5.43)	2.62e-11	1.24e-44	1.31e-12 (5.87e-12)	1.90e-06
f_7	3.93e-03	1.87e-03	2.60e-03 (5.74e-04)	6.34e-08	8.88e-16	1.00e-08 (1.98e-08)	1.90e-06
f_8	36.81	11.93	23.28 (6.44)	0	0	0 (0)	1.90e-06
f_9	7.99e-03	2.71e-04	8.68e-04 (1.68e-03)	6.47e-13	0	4.40e-14 (1.46e-13)	1.90e-06
f_{10}	0.59	0.29	0.48 (0.07)	5.37e-08	3.19e-16	4.21e-09 (1.21e-08)	1.90e-06
f_{11}	134.70	31.17	81.99 (30.80)	2.53e-06	1.97e-07	9.46e-07 (6.58e-07)	1.90e-06
f_{12}	0.41	0.01	0.15 (0.12)	3.69e-09	2.53e-37	2.90e-10 (8.85e-10)	1.90e-06

Tabela 2, os valores em negrito representam a melhor média e o desvio padrão dos experimentos. Em relação aos valores médios encontrados, FA-EOBL obtém resultados melhores em todos os experimentos. O desvio padrão, que mede a dispersão em torno da média, mostra que a variante proposta obteve valores mais satisfatórios que os do FA original em todos os experimentos. Isso comprova a qualidade e a estabilidade de suas soluções.

5. Conclusões

Este trabalho apresentou uma combinação da meta-heurística Algoritmo do Vagalume (FA) com uma estratégia derivada da OBL, chamada EOBL. O resultado da combinação foi uma variante denominada FA-EOBL. O objetivo do trabalho foi potencializar a diversidade das soluções e aumentar a velocidade de convergência do algoritmo original.

Para validar a variante proposta, o seu desempenho foi comparado ao do FA original. A comparação foi estabelecida por meio da execução de vários testes computacionais utilizando 12 funções de referência, das quais 6 são unimodais e 6 multimodais. Em 91,66% dos experimentos, observou-se que a variante FA-EOBL apresentou melhores resultados quando eles foram comparados aos do FA original. Apenas na função *Step* (f_3)

a variante não superou o FA original porque ambos encontraram o ponto ótimo global.

Os resultados apontados neste artigo são promissores para o contexto das meta-heurísticas bioinspiradas. Como trabalhos futuros, pretende-se avaliar a combinação do FA com uma outra variante da OBL chamada *Quasi Opposition-Based Learning*.

Referências

- Basu, B. and Mahanti, G. K. (2011). Fire fly and artificial bees colony algorithm for synthesis of scanned and broadside linear array antenna. *Progress In Electromagnetics Research B*, 32:169–190.
- Chatterjee, A., Mahanti, G. K., and Chatterjee, A. (2012). Design of a fully digital controlled reconfigurable switched beam concentric ring array antenna using firefly and particle swarm optimization algorithm. *Progress In Electromagnetics Research B*, 36:113–131.
- Goel, S. and Panchal, V. (2014). Performance evaluation of a new modified firefly algorithm. In *3rd International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions)*, pages 1–6.
- Hornig, M.-H., Lee, Y.-X., Lee, M.-C., and Liou, R.-J. (2012). Firefly meta-heuristic algorithm for training the radial basis function network for data classification and disease diagnosis. In *Theory and new applications of swarm intelligence*. InTech.
- Kalra, S. and Arora, S. (2016). Firefly algorithm hybridized with flower pollination algorithm for multimodal functions. In *Proceedings of the International Congress on Information and Communication Technology*, pages 207–219. Springer.
- Silva, E. H. and Bastos Filho, C. J. (2015). Pso efficient implementation on gpus using low latency memory. *IEEE Latin America Transactions*, 13(5):1619–1624.
- Tizhoosh, H. R. (2005). Opposition-based learning: a new scheme for machine intelligence. In *Computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce, international conference on*, volume 1, pages 695–701.
- Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms*, pages 169–178.
- Yu, S., Yang, S., and Su, S. (2013). Self-adaptive step firefly algorithm. *Journal of Applied Mathematics*, 2013.
- Zhou, X., Wu, Z., and Wang, H. (2012). Elite opposition-based differential evolution for solving large-scale optimization problems and its implementation on gpu. In *13th Int. Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, pages 727–732.