

DARLYNE FONTES VIRGINIO
KADYDJA KARLA NASCIMENTO CHAGAS
MÁRCIO ADRIANO DE AZEVEDO
WYLLYS ABEL FARKATT TABOSA

IFRN 10 ANOS

DE CRIAÇÃO
em mais de
UM SÉCULO
de História



editoraifrn

ALGORITMO DO MORCEGO MODIFICADO COM VOO DE LÉVY E APRENDIZAGEM BASEADA EM OPOSIÇÃO ELITE

Fábio Augusto Procópio de Paiva¹¹⁰
Izabele Vitória Oliveira Leite¹¹¹
Marcos Henrique Fernandes Marcone¹¹²

INTRODUÇÃO

Otimização é uma área que, invariavelmente, é utilizada durante tomadas de decisões em quase todos os domínios do mundo real (SHAREEF; MOHIDEEN; ALI, 2015), inclusive na ciência e na engenharia. Neste cenário, os algoritmos de otimização meta-heurísticos são bastante utilizados por serem mais poderosos que os métodos convencionais (YANG, 2010) que são baseados nas lógicas formais ou na programação matemática. Além disso, o tempo necessário para executar um algoritmo meta-heurístico é menor que o dos algoritmos exatos.

110 Doutor (2016) e mestre (2007) em Engenharia Elétrica e de Computação pela Universidade Federal do Rio Grande do Norte (UFRN) e bacharel (1999) em Sistemas de Informação pela Universidade Potiguar (UnP). Professor do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN), atuando nos cursos técnicos e superiores da área de Sistemas de Informação. Suas áreas de interesse incluem: Meta-heurísticas Bioinspiradas, Sistemas de Recomendação, Banco de Dados NoSQL e NewSQL e Internet das Coisas.
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN)
Caixa Postal 59143-455 – Diretoria Acadêmica – Parnamirim/RN – Brasil
E-mail: fabio.procopio@ifrn.edu.br

111 Técnica em Informática (2016) pelo Instituto Federal do Rio Grande do Norte (IFRN), *campus* Parnamirim. Atualmente, está cursando o Bacharelado em Engenharia Química pela Universidade Federal do Rio Grande do Norte (UFRN). Suas áreas de interesse incluem: Meta-heurísticas, Algoritmos Bioinspirados, Gerenciamento de Resíduos Sólidos e Controle de Qualidade.
Universidade Federal do Rio Grande do Norte (UFRN)
Caixa Postal 59078-970 – Centro de Tecnologia – Natal/RN – Brasil
E-mail: izaleite11@hotmail.com

112 Técnico em Informática (2016) pelo Instituto Federal do Rio Grande do Norte (IFRN), *campus* Parnamirim. Atualmente, está cursando o Bacharelado em Ciência e Tecnologia pela Universidade Federal do Rio Grande do Norte (UFRN) e é monitor bolsista da disciplina de Álgebra Linear na Escola de Ciências e Tecnologia (ECT), UFRN. Suas áreas de interesse incluem: Meta-heurísticas, Algoritmos Bioinspirados e Matemática Aplicada à Física.
Universidade Federal do Rio Grande do Norte (UFRN)
Caixa Postal 59078-970 – Escola de Ciências e Tecnologia – Natal/RN – Brasil
E-mail: marcosmarcone48@gmail.com

Nos últimos anos, esses algoritmos vêm sendo amplamente empregados para resolver problemas complexos de otimização. Apesar disso, até onde nossas pesquisas evoluíram, foram encontrados poucos trabalhos publicados por pesquisadores do IFRN. Algumas publicações são ANDRADE (2018), DANTAS (2018), PAIVA *et al* (2018), SOUZA *et al* (2018), PAIVA *et al* (2017), PAIVA *et al* (2016), PAIVA *et al* (2015), PEREIRA (2015), SANTOS *et al* (2014), SOUZA *et al* (2012), GOLDBARG *et al* (2011) e SILVA *et al* (2009).

Os algoritmos inspirados na natureza vêm sendo frequentemente aplicados com sucesso na área de otimização. A maioria deles se baseia em algumas características de sucesso do sistema biológico e, por isso, são chamados de Algoritmos Bioinspirados (FISTER-JR *et al*, 2013). Uma classe especial desse tipo de algoritmo foi inspirada na inteligência dos enxames e, portanto, são conhecidos como algoritmos baseados em inteligência de enxames.

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

Os algoritmos baseados em inteligência de enxames consistem em um conjunto de meta-heurísticas inspirado em alguns seres vivos cujos comportamentos emergentes podem resultar em uma capacidade de resolver problemas complexos (SILVA; BASTOS-FILHO, 2015). O Algoritmo do Morcego (YANG, 2010) é um método meta-heurístico que faz alusão ao processo de ecolocalização dos morcegos, que tem sido utilizado para otimizar problemas de engenharia (WANG *et al*, 2012; BEKDAS; NIGDELI, 2016) e de ciência (YILMAZ; KUCUKSILLE, 2013; WANG; GUO, 2013; AFRABANDPEY *et al*, 2014; SHAN; LIU; SUN, 2016). No entanto, assim como muitas meta-heurísticas, o Algoritmo do Morcego enfrenta um problema quando estagna em ótimos locais, o que faz com que a busca global não seja tão bem-sucedida.

Vários trabalhos, com diferentes abordagens, já foram apresentados a fim de melhorar o desempenho do BA. O *Improved Bat Algorithm* (IBA) (YILMAZ; KUCUKSILLE, 2013) propõe três modificações no algoritmo base: inclusão de um fator ponderado de inércia, frequência adaptativa e hibridização com o algoritmo Colônia de Abelhas Artificiais (ABC – *Artificial Bee Colony*) incluindo a fase de abelhas escoteiras. A variante *Bat Algorithm with Harmony Search* (HSBA) (WANG; GUO, 2013) é uma hibridização que combina o algoritmo de Busca Harmônica (HS – *Harmony Search*) com o BA original, a qual consiste em adicionar uma operação para ajustar a afinação do HS, funcionando como um operador de mutação. A variante *Bat Algorithm based on Chaotic Map* (CBA) (AFRABANDPEY *et al*, 2014) usa sequências caóticas para inicializar a frequência, a intensidade e a taxa de emissão de pulso. Uma outra variante proposta (SHAN; LIU; SUN, 2016) combina a trajetória do voo de Lévy, o operador de mutação diferencial e a Aprendizagem Baseada em Oposição (OBL – *Opposition-Based Learning*).

Baseado no conceito da OBL, proposto por Tizhoosh (TIZHOOSH, 2005), outro conceito da área de Inteligência Computacional que também vem sendo utilizado em abordagens de otimização é chamado Aprendizagem Baseada em Oposição Elite

(EOBL – *Elite Opposition-Based Learning*) (ZHOU; WU; WANG, 2012). Ao contrário da OBL, que utiliza indivíduos comuns da população na tentativa de encontrar uma solução mais próxima da ótima global, EOBL utiliza o melhor indivíduo da população atual, isto é, o indivíduo elite. Zhou, Wu e Wang (ZHOU; WU; WANG, 2012) observaram que a solução oposta encontrada a partir de um indivíduo elite tem uma chance maior de estar mais próxima da solução ótima global, quando comparada à abordagem usada pela OBL.

A fim de melhorar os resultados do BA, este trabalho apresenta uma nova variante combinando duas estratégias que podem ser usadas em algoritmos meta-heurísticos: o voo de Lévy e a EOBL, que são implementadas para melhorar a performance do algoritmo, balanceando as tarefas de diversificação e de intensificação. Os resultados dos experimentos computacionais mostraram que a variante proposta apresenta melhores resultados quando comparada ao BA original e a três variantes da literatura.

O artigo está organizado como segue: na Seção 2, é apresentada uma breve introdução sobre a meta-heurística BA, o funcionamento do mecanismo do voo de Lévy e a aprendizagem baseada em oposição elite; na Seção 3, a nova variante que se propõe a melhorar a performance do BA original é apresentada. Na Seção 4, os experimentos são apresentados e os resultados são discutidos. Por fim, a Seção 5 apresenta as conclusões e os trabalhos futuros.

FUNDAMENTAÇÃO TEÓRICA

Aqui, são apresentadas algumas informações úteis para o entendimento do trabalho apresentado, como a meta-heurística inspirada na ecolocalização de morcegos, o voo de Lévy e a Aprendizagem Baseada em Oposição Elite.

META-HEURÍSTICA INSPIRADA NA ECOLOCALIZAÇÃO DOS MORCEGOS

A ecolocalização é uma sofisticada capacidade biológica que os morcegos utilizam para detectar presas e evitar obstáculos. Durante o voo, eles emitem sons e ouvem ecos para, assim, construir um mapa sônico para representar suas proximidades. Eles são capazes de distinguir um obstáculo de uma presa, mesmo no escuro.

O Algoritmo do Morcego (BA – *Bat Algorithm*) é uma meta-heurística inspirada no processo de ecolocalização dos morcegos (YANG, 2010). No BA, todos os morcegos usam a ecolocalização para calcular distâncias. O morcego possui uma posição (vetor solução), uma velocidade e uma frequência. Para cada iteração, o morcego se move em direção do melhor morcego encontrado até o momento. A fim de implementar a busca global, são atualizadas a posição, a velocidade e a frequência do morcego, de acordo com a Equação (1):

$$\begin{aligned} f_i &= f_{min} + (f_{max} - f_{min}) \cdot \beta \\ v_i^t &= v_i^{t-1} + (x_i^{t-1} - x_g^{t-1}) \cdot f_i \\ x_i^t &= x_i^{t-1} + v_i^t, \end{aligned} \quad (1)$$

onde f e β são, respectivamente, as frequências mínima e máxima. β é um valor aleatório gerado a partir de uma distribuição uniforme $[0, 1]$ e x_g representa a melhor solução global corrente.

Já a busca local é implementada por meio da geração de uma nova solução candidata próxima da melhor solução atual, como segue:

$$x_{nova}^t = x_g^{t-1} + \epsilon \bar{A}^t, \quad (2)$$

onde ϵ é um valor $[-1, 1]$ e \bar{A} é o valor médio do som produzido pelos morcegos.

De acordo com a proximidade do alvo, os morcegos diminuem a sonoridade A , e aumentam a taxa de emissão de pulsos r , como segue:

$$\begin{aligned} A_i^{t+1} &= \alpha A_i^t \\ r_i^{t+1} &= r_i^0 [1 - \exp(-\gamma t)], \end{aligned} \quad (3)$$

onde α e γ são valores constantes. Para qualquer i e t , tem-se

$$\begin{aligned} A_i^t &\rightarrow 0 \\ r_i^t &\rightarrow 0, \text{ onde } t \rightarrow \infty. \end{aligned} \quad (4)$$

O pseudocódigo do BA pode ser sumarizado pelo Algoritmo 01. Na linha 01, o algoritmo é inicializado com uma geração aleatória da população de morcegos. Na linha 02, a frequência inicial f é determinada para a posição i . A linha 03 inicializa a taxa de pulso r e a sonoridade A . As linhas 04 – 17 representam a evolução dos morcegos ao longo do tempo. Nesta evolução, novas soluções são estabelecidas e a frequência, a velocidade e a posição de cada morcego são atualizadas (linha 06). Na próxima linha, a taxa de emissão de pulsos é comparada com um número real aleatório gerado a partir de uma distribuição uniforme no intervalo $[0, 1]$. Quando o número gerado for maior que r , novas soluções são geradas em torno da melhor solução (linha 09). Na linha 11, a solução atual é avaliada e é verificado se ela será aceita. Em caso positivo, a taxa r é aumentada e a sonoridade A é diminuída (linha 12). Por fim, o bando é ordenado e o melhor morcego é selecionado (linha 14).

```

01: Gere a população inicial de morcegos  $X_i$  ( $i = 1, 2, \dots, N$ ) e  $V_i$ 
02: Determine a frequência de pulso  $f_i$  em  $X_i$ 
03: Inicialize a taxa de pulso  $r_i$  e a amplitude  $A_i$ 
04: enquanto critério de convergência não for atingido faça
05:   para  $i \leftarrow 1$  até  $X$  faça
06:     Gere novas soluções ajustando a frequência e atualize as velocidades e posições
07:     se ( $rand > r_i$ ) então
08:       Selecione uma solução
09:       Gere uma solução próxima da melhor solução
10:     fim se
11:     se ( $rand < A_i$ ) e  $(f(x_i) < f(x_g^{t-1}))$  então
12:       Aumente  $r_i$  e diminua  $A_i$ 
13:     fim se
14:     Classifique os morcegos e encontre o melhor deles
15:   fim para
16:    $t \leftarrow t + 1$ 
17: fim enquanto

```

Algoritmo 01: Pseudocódigo do Algoritmo do Morcego.
Fonte: Yang (2010).

VOO DE LÉVY

O voo de Lévy se baseia no comportamento observado de pássaros e de insetos enquanto voam. Na literatura, ele tem sido estudado de várias formas (YANG; DEB, 2010) e é definido como um tipo de processo estocástico não-gaussiano que possui uma caminhada aleatória, cujos tamanhos dos passos são baseados na distribuição estável de Lévy. O voo de Lévy é descrito na Equação (5):

$$s = \frac{U}{|V|^{1/\lambda}}, \quad (6)$$

$$U \sim N(0, \sigma^2), V \sim N(0, 1), \quad (7)$$

onde σ^2 é a função gama padrão, cuja distribuição é válida para grandes passos e λ , como definido em Yang (2012).

Segundo Yang (2012), a geração de tamanhos de passos pseudoaleatórios que obedeçam, corretamente, a Equação (5) não é uma tarefa trivial. Existem alguns métodos para geração de números aleatórios e o mais eficiente é o algoritmo de Mantegna (MANTEGNA, 1994), o qual usa duas distribuições gaussianas e, conforme equação:

$$\sigma^2 = \left[\frac{\Gamma(1 + \lambda)}{\lambda \Gamma((1 + \lambda)/2)} \cdot \frac{\sin(\pi\lambda/2)}{2^{(\lambda-1)/2}} \right]^{1/\lambda}. \quad (8)$$

(7)

onde s significa que as amostras são geradas a partir de uma distribuição normal gaussiana, com média zero e variância σ^2 , calculada por

$$L \sim \frac{\lambda \Gamma(\lambda) \sin\left(\frac{\pi\lambda}{2}\right)}{\pi} \frac{1}{s^{1+\lambda}}, (s \gg s_0 > 0), \quad (5)$$

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 essas soluções estejam distantes daquela considerada ótima e o pior caso ocorre quando elas estão na posição oposta à da solução 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; WU; WANG, 2012) pode ser usada.

Antes de introduzir EOBL, o conceito de Aprendizagem Baseada em Oposição (OBL – *Opposition-Based Learning*) será explanado. Na área de Inteligência Computacional, OBL tem sido empregada com o objetivo de aumentar a eficiência dos algoritmos. Para um determinado problema, OBL avalia uma solução e a sua respectiva solução oposta possibilitando que uma solução candidata seja encontrada próxima à global.

O conceito de OBL pode ser aplicado não apenas para gerar soluções aleatórias iniciais, mas também a cada iteração do algoritmo no conjunto corrente de soluções (RAHNAMAYAN; TIZHOOSH; SALAMA, 2008). A seguir, são definidos Número Oposto e Ponto Oposto.

Definição 1: Dado , no intervalo , o número oposto é definido na Equação (9) e a mesma definição pode ser estendida, de forma similar, para problemas multidimensionais:

$$\tilde{x} = a + b - x. \quad (9)$$

Definição 2: Dado como sendo um ponto no espaço -dimensional, com , , ..., e [,], {1, 2, ..., }. Assim, o ponto oposto é definido pelos seus componentes:

$$\tilde{x}_i = a_i + b_i - x_i. \quad (10)$$

Uma vez que a definição de OBL foi apresentada, um exemplo será usado para explicar EOBL. Neste artigo, o morcego com o melhor valor de *fitness* é visto como o indivíduo elite. Supondo que o indivíduo elite seja , a solução baseada em oposição elite para o indivíduo pode ser definida como e pode ser obtida pela Equação (11):

$$x_{ij} = \delta \cdot (da_j + db_j) - x_{ej}, i = 1, 2, \dots, N; j = 1, 2, \dots, n, \quad (11)$$

onde é o tamanho da população, é a dimensão de , , (é o limite dinâmico da -ésima variável de decisão. O limite dinâmico pode ser obtido pela seguinte equação:

$$d_{aj} = \min(x_{ij}) \text{ e } d_{bj} = \max(x_{ij}). \quad (12)$$

O limite dinâmico pode fazer com que salte para fora de (. Se isso acontecer, a Equação (13) será usada para reiniciar :

ALGORITMO DO MORCEGO USANDO VOO DE LÉVY E EOBL

O Algoritmo do Morcego é poderoso na tarefa de intensificação, porém ele pode facilmente “cair” em ótimos locais e, portanto, convergir prematuramente. Para aumentar as chances de o algoritmo “escapar” de ótimos locais, este artigo propõe uma variante do BA, resumizada no Algoritmo 02, que utiliza duas técnicas conhecidas como voo de Lévy e Aprendizagem Baseada em Oposição Elite, que oferecem ao algoritmo uma maior capacidade de balanceamento da diversificação e da intensificação.

```
01: Gere a população inicial de morcegos  $X_i$  ( $i = 1, 2, \dots, N$ ) e  $V_i$ 
02: Determine a frequência de pulso  $f_i$  em  $X_i$ 
03: Inicialize a taxa de pulso  $r_i$  e a amplitude  $A_i$ 
04: enquanto critério de convergência não for atingido faça
05:   para  $i \leftarrow 1$  até  $X$  faça
06:     Idem às linhas 6 – 13 do Algoritmo 01
07:      $x_i \leftarrow Levy(x_i)$ 
08:      $x_i \leftarrow EOBL(x_i)$ 
09:   Classifique os morcegos e encontre o melhor deles
10:   fim para
11:    $t \leftarrow t + 1$ 
12: fim enquanto
```

Algoritmo 02: Pseudocódigo do Algoritmo do Morcego Modificado com o Voo de Lévy e EOBL.
Fonte: Autoria própria (2018).

A variante pode ser usada como uma alternativa para lidar com o problema da convergência prematura e, assim, obter melhor desempenho. A variante proposta pode ser resumida no Algoritmo 02, cuja contribuição está nas linhas 07 e 08. Até onde esta pesquisa avançou, nenhuma variante BA foi encontrada com a proposta de combinar Voo de Lévy e EOBL e, portanto, isso garante o ineditismo deste trabalho.

A partir de uma solução, o voo de Lévy é usado para gerar novas soluções, conforme equação a seguir

$$x_i^t = x_i + \gamma Levy(\lambda), \quad x_{ij} < d_{a,j} \parallel x_{ij} > d_{b,j}, \quad (14)$$

onde γ é um parâmetro que controla o tamanho do passo aleatório e λ é um parâmetro de distribuição do voo de Lévy que, neste trabalho, corresponde à dimensão do problema.

Na linha 07, o voo de Lévy é aplicado sobre o morcego atual e, em seguida, na linha 08, EOBL é aplicada sobre o morcego que foi modificado com o voo de Lévy.

EXPERIMENTOS COMPUTACIONAIS

Esta seção apresenta as funções de referência usadas para validar o algoritmo proposto. Também são apresentados a configuração dos experimentos e os valores definidos para os parâmetros do algoritmo. Por fim, várias simulações são apresentadas e a análise dos resultados é realizada.

FUNÇÕES DE REFERÊNCIA

É bastante frequente o emprego de funções de referência com o pressuposto de que a dificuldade delas corresponde àquelas encontradas em aplicações de problemas contínuos do mundo real. Para realizar os experimentos, foram escolhidas dez funções de referência, aplicadas a problemas de minimização. Elas vêm sendo usadas em vários estudos do BA (YILMAZ; KUCUKSILLE, 2013; WANG; GUO, 2013; AFRABANDPEY *et al*, 2014; SHAN; LIU; SUN, 2016) e são apresentadas na Tabela 01. Para cada uma delas, são apresentadas a sua formulação e o espaço de busca.

As funções de referência avaliadas nos experimentos são classificadas em dois grupos distintos. Abaixo, eles são apresentados e as funções agrupadas dentro deles:

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 (), Schwefel 2.22 (), Schwefel 1.2 (), Schwefel 2.21 (), Rosenbrock () e *Noisy Quartic* ().
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: Rastrigin (), Ackley (), Griewank () e Xin–She Yang 4 ().

Tabela 01: Características das funções de referência.

Fonte: Autoria própria (2018).

Formulação	Espaço de Busca	Sol. Ótima
$f_1(x) = \sum_{i=1}^d x_i^2$	$-5.12 \leq x_i \leq 5.12$	$\mathbf{x}^* = (0, \dots, 0)$
$f_2(x) = \sum_{i=1}^d x_i + \prod_{i=1}^d x_i $	$-10 \leq x_i \leq 10$	$\mathbf{x}^* = (0, \dots, 0)$
$f_3(x) = \sum_{i=1}^d \left(\sum_{j=1}^i x_j \right)^2$	$-100 \leq x_i \leq 100$	$\mathbf{x}^* = (0, \dots, 0)$
$f_4(x) = \max_{i=1, \dots, d} x_i $	$-100 \leq x_i \leq 100$	$\mathbf{x}^* = (0, \dots, 0)$
$f_5(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$-30 \leq x_i \leq 30$	$\mathbf{x}^* = (1, \dots, 1)$
$f_6(x) = \sum_{i=1}^d ix_i^4 + rand(0,1)$	$-1.28 \leq x_i \leq 1.28$	$\mathbf{x}^* = (0, \dots, 0)$
$f_7(x) = \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i) + 10]$	$-5.12 \leq x_i \leq 5.12$	$\mathbf{x}^* = (0, \dots, 0)$

Formulação	Espaço de Busca	Sol. Ótima
$f_7(x) = \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i) + 10]$	$-5.12 \leq x_i \leq 5.12$	$x^* = (0, \dots, 0)$
$f_8(x) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + \exp(1)$	$-32 \leq x_i \leq 32$	$x^* = (0, \dots, 0)$
$f_9(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-600 \leq x_i \leq 600$	$x^* = (0, \dots, 0)$
$f_{10}(x) = \left[\sum_{i=1}^n \sin^2(x_i) - \exp(-\sum_{i=1}^n x_i^2) \right] * \exp(-\sum_{i=1}^n \sin^2 \sqrt{ x_i })$	$-10 \leq x_i \leq 10$	$x^* = (0, \dots, 0)$

EXPERIMENTOS E RESULTADOS

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

Os valores atribuídos aos parâmetros foram os mesmos usados por Zhu (2016): sonoridade e taxa de pulso são fixados em 0.5, sonoridade inicial é igual a e sonoridade mínima é igual a . Os valores atribuídos às constantes e são 0.05 e 0.95, respectivamente.

Na Tabela 2, a variante proposta é comparada com outras 3 da literatura. No entanto, as variantes da literatura não foram implementadas nestes experimentos e, por isso, os resultados de cada uma delas foram extraídos diretamente de Zhu (2016). Assim, para que nenhum algoritmo se beneficie de uma configuração que possa vir a ser a mais adequada, foi usada a mesma configuração definida em Zhu (2016).

São realizadas 30 execuções independentes, em 30 dimensões, durante 900 iterações. O tamanho do bando de morcegos é fixado em . As figuras 1, 2, 3, 4 e 5 mostram o processo de convergência médio dos algoritmos.

A subfigura 1a mostra a otimização da função Esfera, na qual é observado que o BA para de evoluir nas primeiras iterações e permanece estagnado até o fim das iterações. Já o BA modificado apresenta uma convergência ativa e obtendo boas soluções. Para a função Schwefel 2.22, subfigura 1b, o comportamento do BA tem uma evolução ativa até próximo da iteração 150; em seguida, permanece estagnado até próximo da iteração 700; na sequência, BA apresenta uma leve melhoria, no entanto, volta a estagnar.

Nas subfiguras 2a (Schwefel 1.2) e 2b (Schwefel 2.21), a variante proposta apresenta, mais uma vez, um comportamento ativo desde o início até o fim de suas iterações. Por outro lado, BA estagna no início das primeiras iterações. Nas subfiguras 3a (Rosenbrock) e 3b (*Noisy Quartic*), novamente BA apresenta um comportamento de estagnação no início das iterações até o final. No entanto, para a função Rosenbrock, a variante pro-

posta apresenta um comportamento ativo até a iteração 150 e, na sequência, ela estagna, embora a qualidade das soluções encontradas supere a do BA. Já na função *Noisy Quartic*, a variante apresenta uma alta velocidade de convergência nas primeiras iterações e, depois da iteração 300, converge mais lentamente.

As figuras 4 e 5 referem-se às funções multimodais. Para (função Rastrigin), subfigura 4a, a variante proposta apenas depois da iteração 720 apresentou uma velocidade de convergência mais lenta. Já na subfigura 4b, função Ackley, o algoritmo proposto apresentou um comportamento de convergência ativo durante todas as iterações. Por outro lado, em ambas as funções, o algoritmo original passou praticamente todas as iterações estagnado. Por fim, nas subfiguras 5a (função Griewank) e 5b (função Xin-She Yang 4), BA apresentou um comportamento ativo nas primeiras iterações, mas, em seguida, “caiu” em ótimos locais. Todavia, a variante apresentada, nas primeiras iterações apresentou alta velocidade de convergência e, ao longo das iterações, comportou-se mais suavemente.

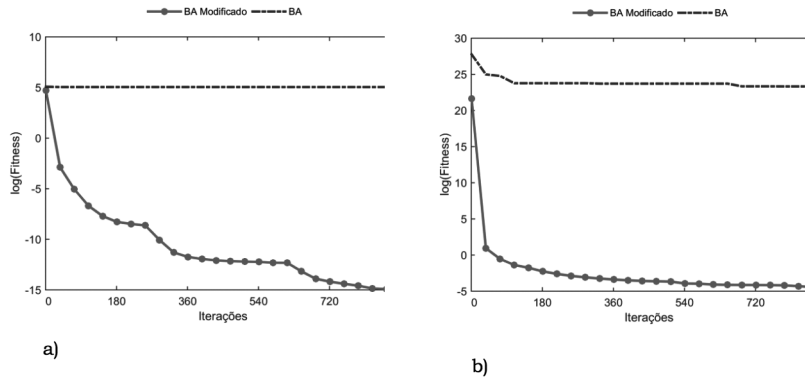


Figura 1: Comportamento médio do BA e da variante BA + Levy + EOBL: a) Função f_1 ; b) Função f_2 .
Fonte: Autoria própria (2018).

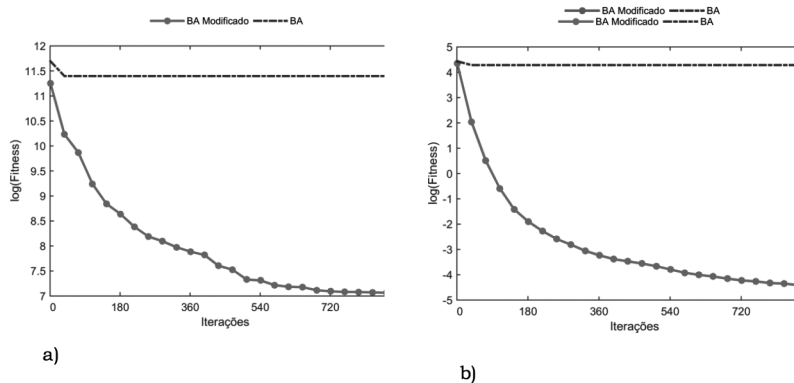


Figura 2: Comportamento médio do BA e da variante BA + Levy + EOBL: a) Função f_3 ; b) Função f_4 .
Fonte: Autoria própria (2018).

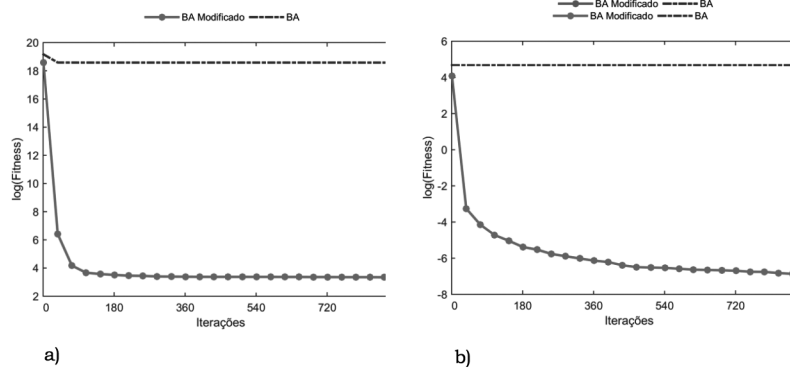


Figura 3: Comportamento médio do BA e da variante BA + Levy + EOBL: a) Função f_5 ; b) Função f_6 .
Fonte: Autoria própria (2018).

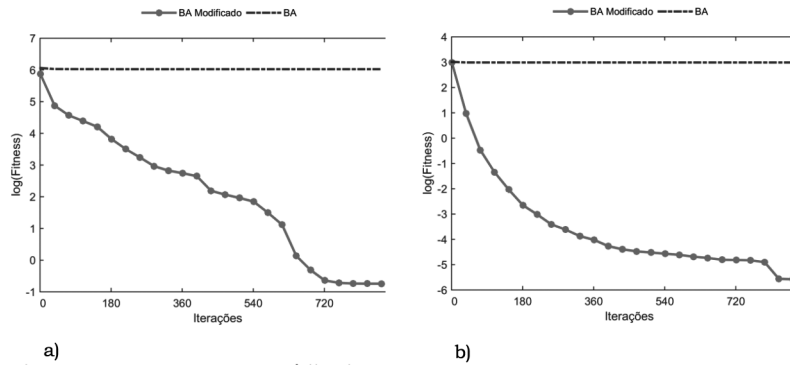


Figura 4: Comportamento médio do BA e da variante BA + Levy + EOBL: a) Função f_7 ; b) Função f_8 .
Fonte: Autoria própria (2018).

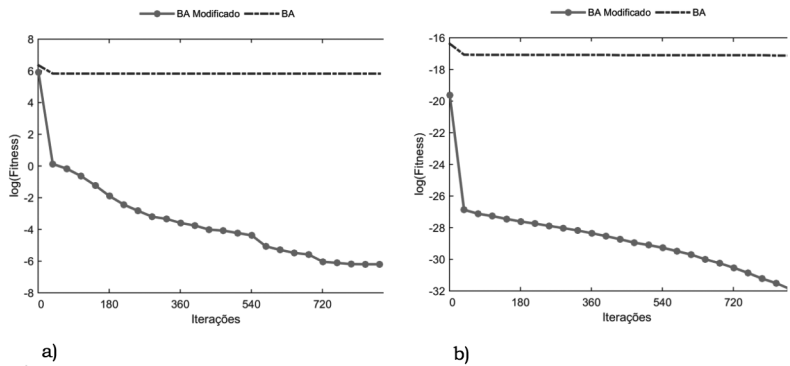


Figura 5: Comportamento médio do BA e da variante BA + Levy + EOBL: a) Função f_9 ; b) Função f_{10} .
Fonte: Autoria própria (2018).

A Tabela 02 apresenta resultados numéricos da comparação entre o BA, a variante proposta e outras 3 variantes da literatura: IBA, HSBA e CBA. Os valores em negrito representam as melhores soluções encontradas. Os valores referentes às três variantes da literatura foram extraídos de Zhu (2016). Sobre os valores médios encontrados, a variante proposta obteve resultados melhores em todas as funções. O desvio padrão, que mede a dispersão em torno da média, mostrou que variante proposta obteve valores mais satisfatórios que os do BA e das outras variantes em todos os experimentos. Isso comprova a qualidade e a estabilidade de suas soluções.

Tabela 02: Comparação de desempenho de diversos algoritmos.
Fonte: Autoria própria (2019).

FUNÇÃO	BA	IBA	HSBA	CBA	BA+LEVY+EOBL
f_1	1.55e+02 (23.60)	1.34e-03 (1.63e-04)	0.14 (2.90e-02)	1.34e-03 (1.60e-04)	3.28e-07 (1.09e-06)
f_2	1.35e+10 (4.51e+10)	12.32 (25.65)	1.60 (0.20)	1.10e+05 (4.99e+05)	1.11e-02 (2.59e-02)
f_3	8.91e+04 (3.13e+04)	1.95e+03 (1.43e+03)	4.01 (1.28)	7.42e+03 (3.68e+03)	1.16e+03 (1.30e+03)
f_4	72.52 (8.38)	23.25 (4.57)	1.10 (1.52)	32.38 (7.62)	1.10e-02 (1.51e-02)
f_5	4.13e+02 (2.49e+01)	1.02e+02 (1.40e+02)	64.32 (34.54)	1.15e+02 (1.35e+02)	0.47 (2.58)
f_6	1.08e+02 (26.08)	3.74e-02 (1.11e-02)	0.13 (5.03e-02)	3.51e-02 (1.35e-02)	1.02e-03 (6.61e-04)
f_7	4.13e+02 (24.98)	1.40e+02 (38.51)	36.74 (10.77)	1.69e+02 (44.03)	0.47 (2.58)
f_8	19.85 (0.29)	14.86 (1.36)	0.63 (0.59)	17.11 (1.60)	2.94e-03 (6.64e-03)
f_9	3.38e+02 (65.04)	63.57 (16.63)	9.12e-03 (2.14e-03)	1.21e+02 (38.23)	2.02e-03 (6.60e-03)
f_{10}	3.66e-08 (3.853e-08)	1.35e-03 (1.42e-04)	0.10 (0.13)	1.34e-03 (1.79e-04)	8.54e-15 (2.50e-14)

CONCLUSÕES

Este trabalho propôs uma variante do Algoritmo do Morcego a fim de melhorar o desempenho do algoritmo original. Para aumentar as chances de o algoritmo “escapar” de ótimos locais, a proposta combinou o BA com duas estratégias que atuam no processo de exploração do espaço de busca: voo de Lévy e EOBL.

A performance do algoritmo original e a da variante proposta foram comparadas por meio de vários experimentos computacionais. A partir dos resultados obtidos, observou-se que o novo algoritmo superou o BA original e três variantes da literatura, no que diz respeito à qualidade e à estabilidade das soluções encontradas. Também foram realizados outros experimentos – embora eles não tenham sido descritos no texto – que combinaram o BA original com o voo de Lévy e a OBL. A partir da análise desses experimentos, observou-se que a combinação voo de Lévy e OBL apresentou uma performance bastante inferior àquela que foi obtida da combinação proposta neste trabalho, isto é, voo de Lévy e EOBL.

Outra meta-heurística que vem apresentando bons resultados é a *Flower Pollination Algorithm* (FPA). Como trabalhos futuros, pretende-se aplicar ao FPA uma nova abordagem de serendipidade, um conceito específico da área de sistemas de recomendação. Esse método se baseia nas dimensões acaso e sagacidade e pode ser utilizado para aumentar a diversidade do algoritmo.

REFERÊNCIAS

AFRABANDPEY, Hodayun et al. A novel bat algorithm based on chaos for optimization tasks. In: **Iranian Conference on Intelligent Systems (ICIS)**, 2014. IEEE, 2014. p. 1-6.

ANDRADE, Romerito C.; GOLDBARG, Marco C.; GOLDBARG, Elizabeth F. G.. An algorithm based on Voronoi diagrams for the multi-stream multi-source multicast routing problem. **International Journal of Innovative Computing and Applications**, v. 9, n. 4, p. 216-229, 2018.

BEKDAS, Gebrail; NIGDELI, Sinan Melih. Bat algorithm for optimization of reinforced concrete columns. **Proceedings in Applied Mathematics and Mechanics (PAMM)**, v. 16, n. 1, p. 681-682, 2016.

DANTAS, Lucas H. A.. **Uma abordagem metaheurística para o problema de alocação de horário escolar no IFRN**. 2018. 76 f. Dissertação (Mestrado em Engenharia de Software), Instituto Metrópole Digital, Universidade Federal do Rio Grande do Norte, Natal, 2018.

FISTER-JR, Iztok *et al.* A brief review of nature-inspired algorithms for optimization. **arXiv preprint arXiv:1307.4186**, 2013.

GOLDBARG, Marco C.; TAUMATURGO, Camila N. O.; GOLDBARG, Elizabeth F. G.. **Algoritmo Busca Tabu para a solução do Nonograma**. XLIII Simpósio Brasileiro de Pesquisa Operacional, 2011.

MANTEGNA, Rosario Nunzio. Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes. **Physical Review E**, v. 49, n. 5, p. 4677, 1994.

PAIVA, Fábio A. P.; COSTA, José A. F.; SILVA, Cláudio R. M.. An alternative approach for particle swarm optimisation using serendipity. *International Journal of Bio-Inspired Computation*, v. 11, n. 2, p. 81-90, 2018.

PAIVA, Fábio A. P.; COSTA, José A. F.; SILVA, Cláudio R. M.. A Serendipity-Based Approach to Enhance Particle Swarm Optimization Using Scout Particles. **IEEE Latin America Transactions**, v. 15, n. 6, p. 1101-1112, 2017.

PAIVA, Fábio; COSTA, José; SILVA, Cláudio. A SERENDIPITY-BASED PSO APPROACH TO DELAY PREMATURE CONVERGENCE USING SCOUT PARTICLES. **International Journal of Innovative Computing, Information and Control**, v. 12, n. 4, 2016.

PAIVA, Fábio A. P.; Costa, José A. F.; SILVA, Cláudio R. M.. **Uma Meta-Heurística Alternativa de Inteligência de Enxames Baseada em Serendipidade Guiada**. Em: Anais do XII Congresso Brasileiro de Inteligência Computacional, 2015.

PEREIRA, Francisco E. M.. **Implementação de um Algoritmo Paralelo e Escalável do Ant Colony System aplicado ao Problema do Roteamento de Veículos**. 2015. 56 f. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas), Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, Pau dos Ferros, 2015.

RAHNAMEYAN, Shahryar; TIZHOOSH, Hamid R.; SALAMA, Magdy MA. Opposition-based differential evolution. **IEEE Transactions on Evolutionary computation**, v. 12, n. 1, p. 64-79, 2008.

SANTOS, João P. Q.; MELO, Jorge D.; DUARTE-NETO, Adrião D.; ALOISE, Daniel. Reactive search strategies using reinforcement learning, local search algorithms and variable neighborhood search. **Expert Systems with Applications**, v. 41, n. 10, p. 4939-4949, 2014.

SHAN, Xian; LIU, Kang; SUN, Pei-Liang. Modified bat algorithm based on lévy flight and opposition based learning. **Scientific Programming**, 2016.

SHAREEF, SK Subhani; MOHIDEEN, E. Rasul; ALI, Layak. Directed Firefly algorithm for multimodal problems. Em: **IEEE International Conference on Compu-**

tational Intelligence and Computing Research (ICCIC). IEEE, 2015. p. 1-6.

SILVA, Eric H. M.; BASTOS-FILHO, Carmelo J. A.. PSO efficient implementation on GPUs using low latency memory. **IEEE Latin America Transactions**, v. 13, n. 5, p. 1619-1624, 2015.

SILVA, Ana; MARINHO, Eberton; OLIVEIRA, Wagner; ALOISE, Daniel. **Genetic algorithm and memetic algorithm with vocabulary building for the SONET ring assignment problem**. Em: Anais do *European Chapter on Metaheuristics*, 2009.

SOUZA, Givanaldo R.; GOLDBARG, Elizabeth F.G.; GOLDBARG, Marco C.; CANUTO, Anne M. P.. **A Multiagent Approach for Metaheuristics Hybridization Applied to the Traveling Salesman Problem**. Em: Anais do XII Simpósio Brasileiro de Redes Neurais, 2012.

SOUZA, Givanaldo R.; GOLDBARG, Elizabeth F. G.; CANUTO, Anne M. P.; GOLDBARG, Marco C.; RAMOS, Iloneide C.. MAHM: a PSO-based multiagent architecture for hybridisation of metaheuristics. **Swarm Intelligence: Principles, current algorithms and methods**, v. 1, p. 237-264, 2018.

TIZHOOSH, Hamid R. Opposition-based learning: a new scheme for machine intelligence. Em: **International conference on Computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce**. IEEE, 2005. p. 695-701.

WANG, Gaige et al. A bat algorithm with mutation for UCAV path planning. **The Scientific World Journal**, 2012.

WANG, Gaige; GUO, Lihong. A novel hybrid bat algorithm with harmony search for global numerical optimization. **Journal of Applied Mathematics**, 2013.

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

YANG, Xin-She. Flower pollination algorithm for global optimization. Em: **International conference on unconventional computing and natural computation**. Springer, Berlin, Heidelberg, 2012. p. 240-249.

YANG, Xin-She; DEB, Suash. Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization. Em: **Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)**. Springer, Berlin, Heidelberg, 2010. p. 101-111.

YILMAZ, Selim; KUCUKSILLE, Ecir U. Improved bat algorithm (IBA) on continuous

optimization problems. **Lecture Notes on Software Engineering**, v. 1, n. 3, p. 279, 2013.

ZHOU, Xinyu; WU, Zhijian; WANG, Hui. Elite opposition-based differential evolution for solving large-scale optimization problems and its implementation on GPU. Em: **13th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)**, 2012. p. 727-732.

ZHU, Binglian et al. A novel quantum-behaved bat algorithm with mean best position directed for numerical optimization. **Computational intelligence and neuroscience**, 2016.