

Uma Nova Variante do Algoritmo do Morcego Baseada em uma Modificação no Operador de Mutaç o de Michalewicz

Izabele V. O. Leite¹, Marcos H. F. Marcone², F bio A. P. Paiva³

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

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

³ Diretoria Acad mica
Instituto Federal do Rio Grande do Norte (IFRN) – Parnamirim – RN – Brasil

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

Abstract. *Metaheuristics have been widely used to solve optimization problems. In this paper, a new variant of the metaheuristic inspired by echolocation of bats is presented. The proposed variant is the result of the combination between the original Bat Algorithm (BA) and a modification in the Michalewicz mutation operator. To evaluate the performance of this new variant, it was compared to the original BA and a variant of the literature named Modified BA (MBA). The computational experiments were performed with 30 independent runs to optimize six benchmark functions. The superiority of the proposed variant has been evidenced when its results are compared to the original BA and to the MBA variant results.*

Resumo. *As meta-heur sticas v m sendo bastante utilizadas para resolver problemas de otimiza o. Neste trabalho,   apresentada uma nova variante da meta-heur stica inspirada na ecolocaliza o dos morcegos. A variante proposta   o resultado da combina o entre o Bat Algorithm (BA) original e uma modifica o no operador de muta o de Michalewicz. Para avaliar o desempenho da nova variante, ela foi comparada com o BA original e com uma variante da literatura chamada Modified BA (MBA). Os experimentos computacionais foram realizados com 30 execu es independentes a fim de otimizar seis fun es de refer ncia. A superioridade da variante proposta foi evidenciada ao comparar os seus resultados com os do BA original e com os do MBA.*

1. Introdu o

Os problemas de otimiza o est o presentes em diversas aplica es reais como telecomunica es [Silva and Martins 2013], sistemas de pot ncia [Chiaradonna et al. 2015], sinaliza o de tr fego [Matos and Carvalho 2016] e muitos outros. A otimiza o   um processo que visa encontrar a melhor solu o para um determinado problema. Assim, h  uma necessidade constante em se desenvolver algoritmos eficientes e robustos capazes de resolver problemas reais. Para resolver problemas desse tipo, os algoritmos meta-heur sticos t m sido amplamente utilizados.

As meta-heur sticas podem ser usadas para orientar uma heur stica a fim de diversificar e de intensificar o espa o de busca do problema. Algumas meta-heur sticas se

inspiraram na natureza e, como um exemplo, podem ser citadas aquelas que se baseiam na inteligência de enxames. Por conta disso, elas são conhecidas como algoritmos baseados na inteligência de enxames. Normalmente, os sistemas de inteligência de enxame são implementados por uma população de agentes simples que interagem uns com os outros e com o seu próprio ambiente. A inteligência de enxames implementa a coletividade de grupos formados por agentes da natureza como pássaros, vaga-lumes, morcegos e outros.

Nos últimos anos, várias meta-heurísticas baseadas na inteligência de enxames têm sido propostas. O Algoritmo do Morcego (*Bat Algorithm* – BA) [Yang 2010], por exemplo, é inspirado na ecolocalização dos morcegos e, para equilibrar a diversificação e a intensificação, ele usa uma técnica de ajuste de frequência. O BA foi usado para resolver diversos problemas reais como veículos aéreos não-tripulados [Wang et al. 2012], classificação de dados [Mishra et al. 2012] e sistemas de distribuição [Reddy and Manoj 2012].

Um problema comumente enfrentado pelos algoritmos meta-heurísticos é conhecido como convergência prematura e ela ocorre quando o algoritmo fica “preso” em um ótimo local. A meta-heurística baseada na ecolocalização dos morcegos é poderosa em buscas locais, porém, às vezes, ela pode “cair” em um ótimo local e não conseguir “escapar” fazendo com que não realize uma boa busca global.

Vários trabalhos foram apresentados a fim de melhorar o desempenho do BA original. O *Hybrid Bat Algorithm* (HBA) [Fister Jr et al. 2013] é o resultado de uma hibridação do BA com as estratégias da Evolução Diferencial [Storn and Price 1997]. Em [Yilmaz and Kucuksille 2013], é apresentado o *Improved Bat Algorithm* com três modificações: Modificação no Fator de Peso de Inércia, Modificação da Frequência Adaptativa e Modificação da Abelha Escoteira. A fim de melhorar o mecanismo de exploração do BA original, o *Modified Bat Algorithm* (MBA) [Yilmaz et al. 2014] propõe uma modificação nas equações que modelam a taxa de emissão de pulso e a sonoridade de cada morcego. Em [Wang et al. 2012] e em [Zhang and Wang 2012], as propostas usaram estratégias de mutação da Evolução Diferencial [Storn and Price 1997] como uma alternativa para melhorar o algoritmo original.

Este trabalho utiliza a mutação de Michalewicz [Michalewicz 1996] a fim de apresentar uma variante da meta-heurística inspirada na ecolocalização dos morcegos. O operador de mutação garante que, no início do processo de otimização, o algoritmo realize buscas globais e, ao longo da evolução das iterações, as buscas passem a ser locais. Para melhorar a performance do BA original, o operador de Michalewicz foi modificado a fim de que a mutação ocorra em todas as dimensões do vetor que representa a posição do morcego, ao invés de apenas uma dimensão. Seis funções de referência são usadas para comparar a variante proposta e o BA original. Após os experimentos, os resultados mostraram que o desempenho da variante apresentada superou o do BA.

O trabalho está organizado da seguinte forma: na Seção 2, é apresentada uma breve introdução sobre a meta-heurística BA e o operador de mutação de Michalewicz; na Seção 3, a nova variante é apresentada para melhorar a performance do BA original. Na Seção 4, os experimentos são apresentados e os resultados são discutidos. Por fim, a Seção 5 apresenta as considerações finais e os trabalhos futuros.

2. Fundamentação Teórica

Aqui são apresentadas algumas informações necessárias para o entendimento da variante proposta na Seção 3, como a meta-heurística inspirada na ecolocalização de morcegos e o operador de mutação de Michalewicz.

2.1. Metaheurística Inspirada na Ecolocalização de Morcegos

Os algoritmos meta-heurísticos bioinspirados têm recebido muita atenção por parte dos cientistas da computação. Em [Yang 2010], é proposto um desses algoritmos o qual é inspirado na ecolocalização de morcegos e, por isso, é conhecido como *Bat Algorithm* (BA). A ecolocalização é uma sofisticada capacidade biológica que os morcegos utilizam para detectar presas e evitar obstáculos. Durante o voo, os morcegos emitem ondas sonoras e ouvem ecos para, assim, construir um mapa sônico para representar suas proximidades. Dessa forma, eles podem calcular o quão longe estão de um determinado objeto e, além disso, eles são capazes de distinguir um obstáculo de uma presa, mesmo no escuro [Nakamura et al. 2012].

Na meta-heurística BA, todos os morcegos usam a ecolocalização para calcular distâncias. O i -ésimo morcego possui uma posição x_i (vetor solução), uma velocidade v_i e uma frequência f_i . Para cada iteração t , o i -ésimo 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 i -ésimo 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_{min} e f_{max} são, respectivamente, as frequências mínima e máxima. β é um valor aleatório gerado a partir de uma distribuição uniforme $\in [0, 1]$ e x_g^{t-1} 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 aleatório $\in [-1, 1]$ e \bar{A}^t é o valor médio do som produzido por todos os morcegos. Além da velocidade e da frequência, as ondas sonoras emitidas pelos morcegos possuem outras duas características: a taxa de pulso e a sonoridade. A taxa de pulso é a medida da variação das perturbações que se propagam através de um meio. Já a sonoridade, também conhecida como amplitude, é a magnitude de oscilação de uma onda [Cordeiro et al. 2012]. Quando um morcego identifica uma presa, de acordo com a sua proximidade, a sua taxa de pulso (r) é aumentada e sua sonoridade (A) é diminuída. No BA, implementa-se a sonoridade $A_i \in [A_0, A_{min}]$ e a taxa $r_i \in [0, 1]$, 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 $0 < \alpha < 1$ e $\gamma > 0$, tem-se

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

O pseudocódigo do BA pode ser sumarizado pelo Algoritmo 1. Ele é inicializado com uma geração aleatória da população de morcegos (linha 1). Na linha 2, a frequência inicial f_i é determinada para a posição X_i . A linha 3 inicializa a taxa de pulso r_i e a sonoridade A_i . As linhas 4–16 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 6). 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_i , novas soluções são geradas em torno da melhor solução (linha 9). Na linha 11, a solução atual é avaliada e é verificado se ela será aceita. Em caso positivo, a taxa r_i é aumentada e a sonoridade A_i é diminuída (linha 12). Por fim, o bando é ordenado e o melhor morcego é selecionado (linha 14).

Algoritmo 1 Pseudocódigo do Algoritmo do Morcego

- 1: Inicializa a população de morcegos $X_i \ i = (1, 2, \dots, N)$ e V_i
 - 2: Define a frequência de pulso f_i em X_i
 - 3: Inicialize a taxa de pulso r_i e a sonoridade A_i
 - 4: **while** ($t \leq num_max_iter$) **do**
 - 5: **for** $i = 1 : X$ **do**
 - 6: Gere novas soluções ajustando frequência e atualizando velocidades e posições
 - 7: **if** ($rand > r_i$) **then**
 - 8: Selecione a melhor solução
 - 9: Gere uma solução nas proximidades da melhor solução
 - 10: **end if**
 - 11: **if** ($rand < A_i \ \& \ fit(x_i) < fit(x_g^{t-1})$) **then**
 - 12: Aumente r_i e diminua A_i
 - 13: **end if**
 - 14: Classifique os morcegos e encontre o melhor
 - 15: **end for**
 - 16: **end while**
-

2.2. Operador de Mutação de Michalewicz

O processo de mutação consiste na geração de novas soluções a partir de uma solução inicial. Assim, dada uma solução $x \in \mathfrak{R}$, o valor correspondente de x mutado pode ser expresso por $x' = mut(x)$, onde $mut(\cdot)$ é a função de mutação.

Os operadores de mutação são parte integrante das técnicas de computação evolutiva e podem ser usados para evitar a perda da diversidade de uma população. Eles podem introduzir novos indivíduos a uma população adicionando variabilidade à população e impedindo a estagnação em ótimos locais [Dumitrescu et al. 2000]. Assim, a inclusão de um operador de mutação ao BA pode aumentar a sua capacidade de busca.

Michalewicz desenvolveu um operador de mutação [Michalewicz 1996] para ser usado, inicialmente, em um Algoritmo Genético, o qual é chamado de operador de mutação não-uniforme de Michalewicz. O operador pode ser usado para trabalhar modificando uma dimensão do vetor que representa a posição do morcego x_{id} , como mostrado

$$mut(x_{id}) = \begin{cases} x_{id} + \Delta(t, u - x_{id}) : & b_a = 1 \\ x_{id} + \Delta(t, x_{id} - l) : & b_a = 0, \end{cases} \quad (5)$$

onde t é o número da iteração atual, u e l são os limites superior e inferior do valor da dimensão do morcego, respectivamente; b_a é um bit gerado aleatoriamente. A função $\Delta(t, y)$ retorna um valor no intervalo $[0, y]$, tal que a probabilidade de $\Delta(t, y)$ ser próximo de 0 aumenta à medida que t aumenta. Essa propriedade faz com que, nas primeiras iterações (isto é, quando t é pequeno), o algoritmo realize buscas globais, ao passo que quando t aumenta, as buscas passam a ser locais. A função $\Delta(t, y)$ é dada por:

$$\Delta(t, y) = y \left(1 - a^{(1-t/T)^b} \right), \quad (6)$$

onde a é um número aleatório gerado a partir de uma distribuição uniforme no intervalo $[0, 1]$, T é o número máximo de iterações do algoritmo e b é um parâmetro de tunelamento, cujo valor padrão é 5, como descrito em [Michalewicz 1996].

3. Uma Nova Variante BA Baseada no Operador Modificado de Michalewicz

O Algoritmo do Morcego é uma técnica capaz de encontrar bons resultados em um tempo de convergência rápido, porém ele pode convergir prematuramente. Diz-se que o algoritmo convergiu prematuramente quando a solução que ele propõe se aproxima de um ótimo local, ao invés do ótimo global. A convergência prematura ocorre devido à diminuição da diversidade no espaço de busca [Evers and Ghaliya 2009], o que leva o algoritmo a um estado de estagnação.

Para reduzir os problemas que podem ser causados pela convergência prematura, este trabalho propõe a combinação do operador de mutação de Michalewicz com o Algoritmo do Morcego. No entanto, ao contrário da abordagem original do operador, que altera apenas uma dimensão do vetor (x_{id}), como visto na Equação 5, neste trabalho, todas as dimensões do vetor x_i são alteradas, como segue:

$$mut(x_i) = \begin{cases} x_i + \Delta(t, U - x_i) : & num_{sorteado} = 0 \\ x_i + \Delta(t, x_i - L) : & num_{sorteado} = 1, \end{cases} \quad (7)$$

onde t é o número da iteração atual, U e L são os vetores que correspondem aos limites superior e inferior de cada dimensão do morcego, respectivamente; $num_{sorteado}$ tem 50% de probabilidade de ser 0 e 50% de probabilidade de ser 1; $\Delta(t, y)$ é uma função que retorna um valor no intervalo $[0, y]$ e pode ser calculada a partir da Equação 6.

O pseudocódigo do BA baseado no operador modificado de Michalewicz pode ser resumido pelo Algoritmo 2. A contribuição do trabalho é apresentada nas linhas 11–13. As demais são semelhantes ao Algoritmo 1 e já foram descritas na Subseção 2.1. Na linha 11, o valor de *fitness* do morcego atual é comparado com o seu valor de *fitness* quando a mutação de Michalewicz é aplicada. Então, caso essa mutação apresente um valor de

fitness melhor que o valor de *fitness* original, a mutação é confirmada e o morcego atual assume a sua nova posição (linha 12). A verificação na linha 11 é realizada para que a aplicação da mutação só seja efetivada se houver uma melhoria no valor de *fitness*.

Algoritmo 2 Pseudocódigo do Algoritmo do Morcego Modificado com Michalewicz

```
1: Inicializa a população de morcegos  $X_i$   $i = (1, 2, \dots, N)$  e  $V_i$ 
2: Define a frequência de pulso  $f_i$  em  $X_i$ 
3: Inicialize a taxa de pulso  $r_i$  e a sonoridade  $A_i$ 
4: while ( $t \leq num\_max\_iter$ ) do
5:   for  $i = 1 : X$  do
6:     Gere novas soluções ajustando frequência e atualizando velocidades e posições
7:     if ( $rand > r_i$ ) then
8:       Selecione a melhor solução
9:       Gere uma solução nas proximidades da melhor solução
10:    end if
11:    if  $fit(mutacao\_michalewicz(x_i)) < fit(x_i)$  then
12:       $x_i \leftarrow mutacao\_michalewicz(x_i)$ , conforme Equação 7
13:    end if
14:    if ( $rand < A_i \ \& \ fit(x_i) < fit(x_g^{t-1})$ ) then
15:      Aumente  $r_i$  e diminua  $A_i$ 
16:    end if
17:    Classifique os morcegos e encontre o melhor
18:  end for
19: end while
```

4. Experimentos Computacionais

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

4.1. Funções de Referência e Configurações dos Parâmetros

É comum usar funções de referência com a hipótese de que a dificuldade apresentada por elas corresponde àquelas encontradas em aplicações reais. Assim, elas são frequentemente usadas para validar e comparar os algoritmos de otimização, bem como para validar novas abordagens de otimização global [Dieterich and Hartke 2012].

Para realizar os experimentos, foram escolhidas seis funções de referência que são aplicadas em problemas de minimização e que já foram utilizadas em vários estudos de BA [Fister Jr et al. 2013, Kabir and Alam 2014, Shan et al. 2016]. Para cada uma delas, a fórmula, o espaço de busca e a solução ideal são apresentados na Tabela 1. Esfera (f_1), Rosenbrock (f_2) e *Noisy Quartic* (f_3) são funções unimodais. Geralmente, elas são usadas para testar a capacidade do algoritmo em executar buscas locais. Por outro lado, Griewank (f_4), Rastrigin (f_5) e Ackley (f_6) têm vários mínimos locais e são usadas para verificar a habilidade do algoritmo para realizar buscas globais.

Os valores atribuídos aos parâmetros foram os mesmos usados por [Zhu et al. 2016]: sonoridade A_i e taxa de pulso r_i são fixados em 0.5, sonoridade

Tabela 1. Funções de Referência Avaliadas.

Fun	Fórmula	Espaço de Busca	Sol. Ótima
f_1	$f_1(x) = \sum_{i=1}^d x_i^2$	$-100 \leq x_i \leq 100$	$f(x^*) = 0$
f_2	$f_2(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$-30 \leq x_i \leq 30$	$f(x^*) = 0$
f_3	$f_3(x) = \sum_{i=1}^d ix_i^4 + rand(0, 1)$	$-1.28 \leq x_i \leq 1.28$	$f(x^*) = 0$
f_4	$f_4(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-600 \leq x_i \leq 600$	$f(x^*) = 0$
f_5	$f_5(x) = \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i) + 10]$	$-5.12 \leq x_i \leq 5.12$	$f(x^*) = 0$
f_6	$f_6(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + \exp(1)$	$-32.76 \leq x_i \leq 32.76$	$f(x^*) = 0$

inicial A_0 é igual a 2 e sonoridade mínima A_{min} é igual a 0. Os valores atribuídos às constantes γ e α são 0.05 e 0.95, respectivamente.

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).

Nos experimentos, são realizadas 30 execuções independentes. Em cada execução, são fixados o número de dimensões, o número de iterações e a quantidade de morcegos em 30, 900 e 50, respectivamente. Essa configuração também foi a mesma definida nos experimentos realizados em [Zhu et al. 2016]. Para avaliar os resultados da variante proposta, ela foi comparada com o BA original e com a variante da literatura *Modified Bat Algorithm* (MBA) [Yilmaz et al. 2014]. O critério usado para escolher a variante MBA como um parâmetro de avaliação da variante proposta, é o fato de MBA modificar equações que interferem no mecanismo de exploração do BA original.

As figuras 1, 2 e 3 comparam o processo de convergência médio do BA com o do BA modificado com o operador de mutação de Michalewicz (BA-Michalewicz), medido em 30 execuções independentes. O processo ocorre no espaço de busca das funções Esfera, Rosenbrock, *Noisy Quartic*, Griewank, Rastrigin e Ackley, respectivamente.

Nas subfiguras 1(a) e 1(b), que correspondem às funções Esfera e Rosenbrock, é observado que a velocidade de convergência da variante proposta, em relação ao BA original, é bastante superior. Na função Esfera, o BA original se mantém estagnado, praticamente, desde o início das primeiras iterações até o final. Isso mostra que ele convergiu prematuramente e a que a busca pela solução ótima global foi ineficaz. Isso também ocorre na função Rosenbrock. O BA original permanece estagnado, aproximadamente, a partir da iteração 60 até o final. Embora a velocidade de convergência da variante proposta, durante a otimização da função Esfera, seja reduzida a partir da iteração 360 e, para

a função Rosenbrock, a partir da 540, a nova variante encontra soluções muito melhores que as do algoritmo original.

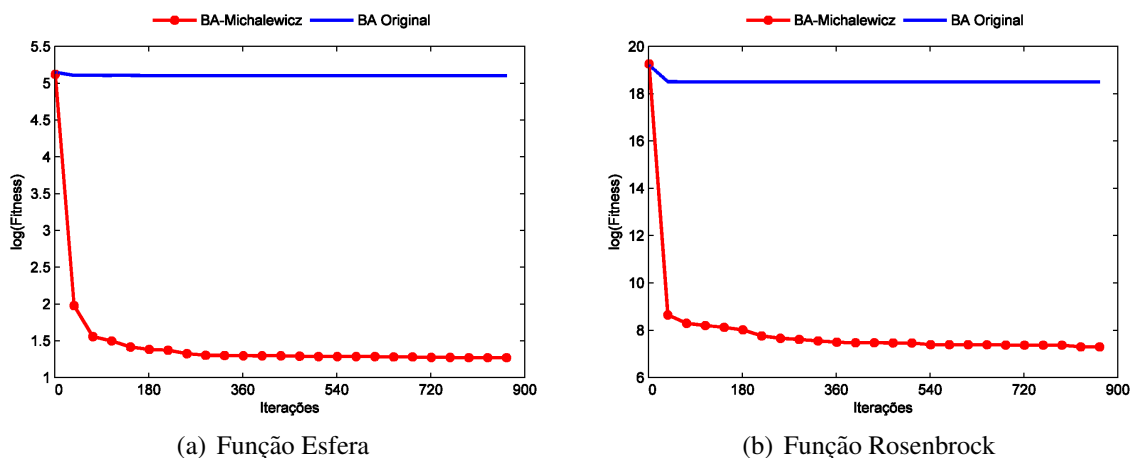


Figura 1. Processo de convergência médio nas funções Esfera e Rosenbrock.

Na função *Noisy Quartic* (subfigura 2(a)) e na função Griewank (subfigura 2(b)), os resultados da variante proposta também superam os do BA original. Na subfigura 2(a), o BA original estagna, praticamente, no início das iterações. E, na subfigura 2(b), ele inicia o seu processo de estagnação próximo da iteração de número 50. Durante todo o processo de busca, é observado que o algoritmo original tem uma baixa velocidade de convergência, apresentando poucas melhorias significativas em seus resultados. Por outro lado, a variante BA-Michalewicz apresenta um comportamento de convergência mais ativo e apresenta melhores soluções que o algoritmo original.

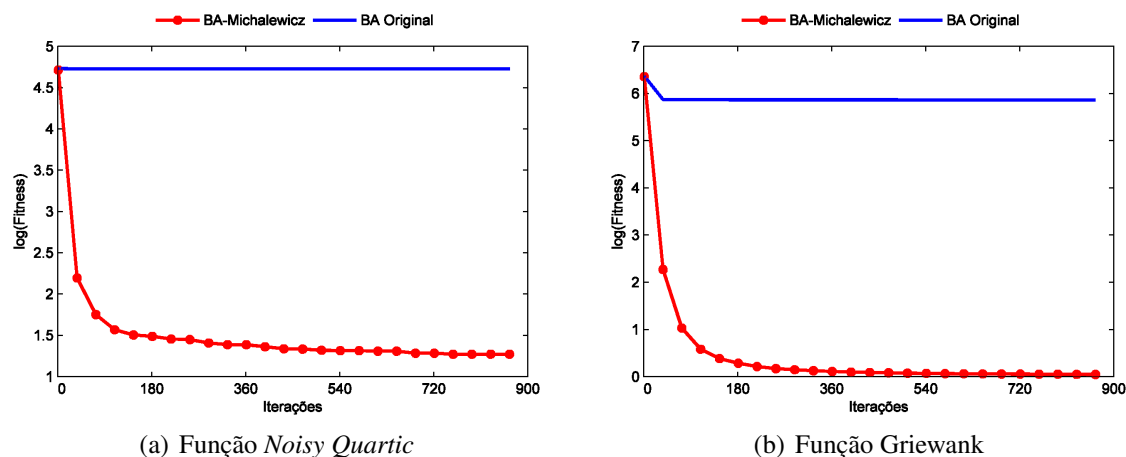


Figura 2. Processo de convergência médio nas funções *Noisy Quartic* e Griewank.

Na subfigura 3(a), que corresponde à função Rastrigin, o BA original estagna, mais uma vez, próximo das primeiras iterações e, na função Ackley (subfigura 3(b)), próxima da iteração 60. Além disso, o algoritmo original obtém soluções insatisfatórias quando comparado à variante proposta, a qual apresenta um comportamento de convergência bastante ativo durante o processo de otimização.

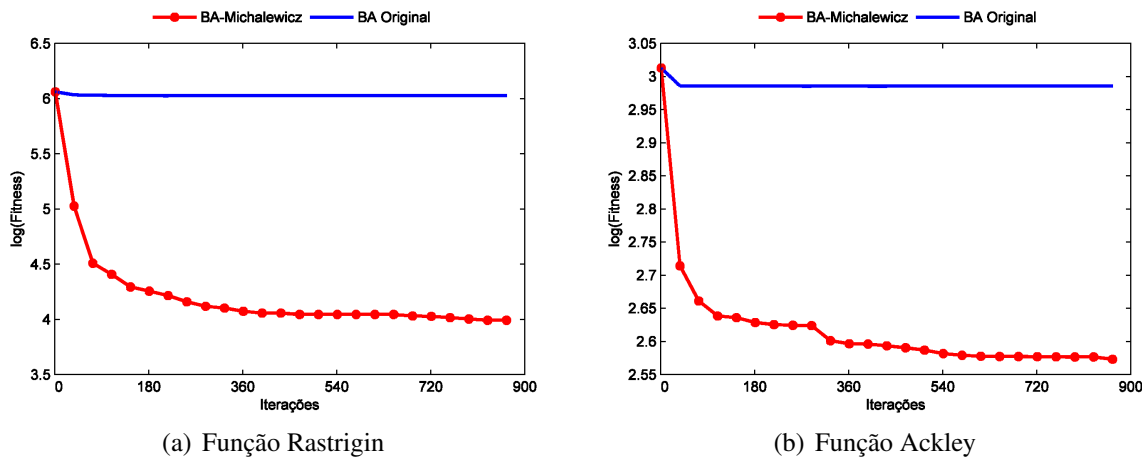


Figura 3. Processo de convergência médio nas funções Rastrigin e Ackley.

Na Tabela 2, são apresentados os resultados dos experimentos computacionais realizados. Em todas as funções avaliadas, os valores médios das soluções encontradas pela variante proposta são melhores que os do BA original e da variante MBA. Nota-se também que, em várias funções, a média da variante proposta é mais satisfatória que os melhores resultados do BA e da variante MBA, comprovando mais uma vez a superioridade do algoritmo proposto.

Em relação ao desvio padrão, que mede a dispersão em torno da média, a variante BA proposta obteve valores mais satisfatórios que o BA original e que o MBA em 66,37% dos experimentos. Apenas nos experimentos com as funções f_5 e f_6 (destacados com um *), o desvio padrão da variante proposta não foi melhor que o do MBA, embora o valor médio das soluções tenha sido. Isso comprova a estabilidade das soluções encontradas.

Tabela 2. Comparação entre BA original, MBA e o BA-Michalewicz.

Fun	BA original		MBA		BA-Michalewicz	
	Melhor	Média (Desvio)	Melhor	Média (Desvio)	Melhor	Média (Desvio)
f_1	1.14e+02	1.64e+02 (2.22e+01)	1.19e+01	2.30e+01 (5.95e+00)	7.36e-03	3.55e+00 (2.39e+00)
f_2	5.17e+07	1.08e+08 (3.61e+07)	1.11e+06	5.56e+06 (3.11e+06)	8.72e-02	1.42e+03 (1.55e+03)
f_3	5.90e+01	1.12e+02 (2.34e+01)	2.26e+00	4.69e+00 (1.71e+00)	4.06e-01	3.56e+00 (1.37e+00)
f_4	2.15e+02	3.51e+02 (9.23e+01)	4.89e+01	8.07e+01 (1.88e+01)	1.00e+00	1.04e+00 (1.83e-02)
f_5	3.47e+02	4.13e+02 (2.48e+01)	1.23e+02	1.54e+02 (1.68e+01)	1.84e-03	5.41e+01 (4.77e+01)*
f_6	1.77e+01	1.97e+01 (4.69 e-01)	1.19e+01	1.49e+01 (1.21e+00)	6.86e-02	1.30e+01 (7.77e+00)*

Wilcoxon é um teste estatístico não-paramétrico utilizado para comparar duas

amostras independentes. A hipótese nula H_0 afirma que as duas amostras vêm da mesma população, enquanto a hipótese alternativa H_1 afirma que uma delas possui valores maiores que a outra. Quando o p -value é menor que o nível de significância, decide-se rejeitar H_0 , isto é, há diferença significativa entre as amostras.

O Teste de Wilcoxon, com nível de significância de 0.05, é aplicado a fim de comparar os resultados das soluções encontradas pela variante proposta e pelo BA original. Ele é usado para verificar se os resultados da variante proposta são estatisticamente significativos quando comparados aos do BA original. Na otimização das funções f_1 , f_2 , f_3 , f_4 e f_5 o valor do p -value foi 1.86e-09 e, para a função f_6 , 4.65e-08, ou seja, inferior ao do nível de significância 0.05 estabelecido no teste. Assim, a hipótese H_0 foi rejeitada garantindo que não foram encontradas igualdades entre as soluções apresentadas pelos BA original e pela variante proposta neste trabalho.

Para cada uma das funções de referência avaliada, a Tabela 2 (melhores resultados são destacados em negrito) mostra a melhor solução encontrada, a média do valor de *fitness* e o desvio padrão do BA original, da variante MBA e da variante proposta, quando eles são executados 30 vezes. Os resultados mostram que a nova variante supera o BA original e a variante MBA. Ao comparar a variante proposta com o BA original e com o MBA, os resultados mostram sua superioridade no que diz respeito à estabilidade e à velocidade de convergência.

5. Conclusões

Este trabalho apresentou uma variante da meta-heurística inspirada no processo de ecolocalização dos morcegos conhecida como *Bat Algorithm* (BA). A nova variante consiste na combinação do BA original com o operador de mutação de Michalewicz. No entanto, este operador foi modificado para que, ao contrário da sua proposta original que altera apenas uma dimensão do vetor, todas as dimensões sejam alteradas. A proposta da variante é gerar diversidade para as soluções e reduzir os problemas causados pela convergência prematura, caracterizado pela estagnação do enxame em ótimos locais.

O desempenho do BA original, da variante MBA e da variante proposta foi comparado por meio de vários experimentos computacionais que foram realizados utilizando seis funções de referência: Esfera, Rosenbrock, *Noisy Quartic*, Griewank, Rastrigin e Ackley. Após os experimentos, observou-se que a nova variante apresentou um bom comportamento médio de convergência e obteve resultados superiores aos algoritmos citados no que diz respeito à qualidade e à estabilidade das soluções encontradas. Isso ocorreu porque o operador de mutação de Michalewicz modificado preservou a diversidade do algoritmo original, tornando o seu processo de convergência mais ativo.

Os resultados apontados neste artigo são promissores para o contexto das meta-heurísticas. Diante disso, o objetivo de trabalhos futuros é avaliar a adaptação de um conceito da área de Inteligência Computacional – que tem sido usado para aumentar a eficiência dos algoritmos meta-heurísticos – denominada *Elite Opposition-Based Learning* (EOBL) em uma outra meta-heurística bioinspirada conhecida como Algoritmo do Vaga-lume (*Firefly Algorithm* – FA). Assim como o Algoritmo do Morcego, FA também tem sido bastante utilizado para resolver problemas de otimização do mundo real.

Referências

- Chiaradonna, S., Di Giandomenico, F., and Murru, N. (2015). On enhancing efficiency and accuracy of particle swarm optimization algorithms. *International Journal of Innovative Computing, Information and Control (IJICIC)*, 11(4):1165–1190.
- Cordeiro, J. A., Parpinelli, R. S., and Lopes, H. S. (2012). Análise de sensibilidade dos parâmetros do bat algorithm e comparação de desempenho.
- Dieterich, J. M. and Hartke, B. (2012). Empirical review of standard benchmark functions using evolutionary global optimization. *arXiv preprint arXiv:1207.4318*.
- Dumitrescu, D., Lazzarini, B., Jain, L. C., and Dumitrescu, A. (2000). *Evolutionary computation*. CRC press.
- Evers, G. I. and Ghahia, M. B. (2009). Regrouping particle swarm optimization: A new global optimization algorithm with improved performance consistency across benchmarks. In *SMC*, pages 3901–3908. IEEE.
- Fister Jr, I., Fister, D., and Yang, X.-S. (2013). A hybrid bat algorithm. *arXiv preprint arXiv:1303.6310*.
- Kabir, M. W. U. and Alam, M. S. (2014). Bat algorithm with self-adaptive mutation: a comparative study on numerical optimization problems. *International Journal of Computer Applications*, 100(10).
- Matos, S. A. and Carvalho, A. B. (2016). Traffic lights signaling optimization as a many-objective optimization problem. In *XIII Encontro Nacional de Inteligência Artificial e Computacional*, pages 421–432.
- Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag.
- Mishra, S., Shaw, K., and Mishra, D. (2012). A new meta-heuristic bat inspired classification approach for microarray data. *Procedia Technology*, 4:802–806.
- Nakamura, R. Y., Pereira, L. A., Costa, K., Rodrigues, D., Papa, J. P., and Yang, X.-S. (2012). Bba: a binary bat algorithm for feature selection. In *25th Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 291–297.
- Reddy, V. U. and Manoj, A. (2012). Optimal capacitor placement for loss reduction in distribution systems using bat algorithm. *IOSR journal of Engineering*, 2(10):23–27.
- Shan, X., Liu, K., and Sun, P.-L. (2016). Modified bat algorithm based on lévy flight and opposition based learning. *Scientific Programming*, 2016.
- Silva, C. R. M. and Martins, S. R. (2013). An adaptive evolutionary algorithm for uwb microstrip antennas optimization using a machine learning technique. *Microwave and Optical Technology Letters*, 55(8):1864–1868.
- Storn, R. and Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359.
- Wang, G., Guo, L., Duan, H., Liu, L., and Wang, H. (2012). A bat algorithm with mutation for uav path planning. *The Scientific World Journal*, 2012.

- Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)*, pages 65–74. Springer.
- Yilmaz, S. and Kucuksille, E. U. (2013). Improved bat algorithm (iba) on continuous optimization problems. *Lecture Notes on Software Engineering*, 1(3):279.
- Yilmaz, S., Kucuksille, E. U., and Cengiz, Y. (2014). Modified bat algorithm. *Elektronika ir Elektrotechnika*, 20(2):71–78.
- Zhang, J. W. and Wang, G. G. (2012). Image matching using a bat algorithm with mutation. In *Applied Mechanics and Materials*, volume 203, pages 88–93.
- Zhu, B., Zhu, W., Liu, Z., Duan, Q., and Cao, L. (2016). A novel quantum-behaved bat algorithm with mean best position directed for numerical optimization. *Computational intelligence and neuroscience*, 2016.