



**UNIVERSIDADE ESTADUAL DA PARAÍBA  
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ  
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**ALLAN VILAR DE CARVALHO**

**PROPOSTA DE UMA VARIANTE BA: ANÁLISE DE SENSIBILIDADE  
DAS ETAPAS ACRESCENTADAS E AVALIAÇÃO DO DESEMPENHO**

**PATOS - PB  
2016**

**ALLAN VILAR DE CARVALHO**

**PROPOSTA DE UMA VARIANTE BA: ANÁLISE DE SENSIBILIDADE  
DAS ETAPAS ACRESCENTADAS E AVALIAÇÃO DO DESEMPENHO**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Wellington Candeia de Araujo.

**PATOS – PB  
2016**

É expressamente proibida a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano da dissertação.

C331p Carvalho, Allan Vilar de  
Proposta de uma variante BA [manuscrito] : análise de sensibilidade das etapas acrescentadas e avaliação do desempenho / Allan Vilar De Carvalho. - 2016.  
20 p. : il. color.

Digitado.  
Trabalho de Conclusão de Curso (Graduação em Computação)  
- Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas, 2016.  
"Orientação: Prof. Dr. Wellington Candeia de Araujo, CCEA".

1. Bat Algorithm. 2. Algoritmo Evolutivo. 3. Algoritmo Bioinspirado. 4. Problemas de Otimização. I. Título.  
21. ed. CDD 005.3

Allan Vilar de Carvalho

**PROPOSTA DE UMA VARIANTE BA: ANÁLISE DE SENSIBILIDADE  
DAS ETAPAS ACRESCENTADAS E AVALIAÇÃO DO DESEMPENHO**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de Bacharel em Ciência da Computação.

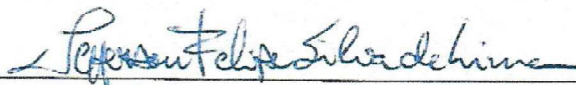
Aprovado em 21 de dezembro de 2016

BANCA EXAMINADORA



---

Wellington Candeia de Araujo  
(Orientador)



---

Jefferson Felipe Silva de Lima  
(Examinador)



---

Rodrigo Alves Costa  
(Examinador)

A minha família, pela dedicação, companheirismo e amizade, DEDICO.

## **AGRADECIMENTOS**

Ao professor Dr. Wellington Candeia de Araujo pelas leituras sugeridas ao longo dessa orientação e pela dedicação.

Ao meu pai Adão Vilar de Carvalho, a minha mãe Acioneide Torres Vilar de Carvalho, as minhas irmãs Arilânia Vilar de Carvalho e Alânia Vilar de Carvalho.

A minha namorada Fiderlane Islane Gomes dos Santos, pela compreensão por minha ausência em várias ocasiões.

Aos colegas de classe pelos momentos de amizade e apoio.

# PROPOSTA DE UMA VARIANTE BA: Análise de Sensibilidade das Etapas Acrescentadas e Avaliação do Desempenho

Allan Vilar de Carvalho\*

## RESUMO

Este artigo apresenta uma análise aprimorada nas novas etapas propostas e inseridas no algoritmo bioinspirado BA (*Bat Algorithm*) aplicado em diversos trabalhos ao longo dos anos, por meio da utilização de 5 funções *benchmark* de otimização extensivamente utilizadas na validação de algoritmos, com o objetivo de melhorá-lo para que futuramente seja aplicado a problemas do mundo real. Além da variação do parâmetro que armazena a quantidade de iterações nas novas soluções não aceitas onde ocasionou o aprimoramento das novas etapas e o encontro de uma quantidade agora padrão que obtém melhores resoluções. As avaliações consistiram na média e desvio padrão das soluções e no custo computacional que comprovam largamente as melhorias ocasionadas no BA canônico.

**Palavras-Chave:** *Bat Algorithm*. Algoritmo Evolutivo. Algoritmo Bioinspirado. Problemas de Otimização.

## 1 INTRODUÇÃO

Algoritmos evolutivos são considerados por muitos pesquisadores, poderosos para resolver diversos problemas de otimização existentes que são difíceis de solucionar por causa da elevada complexidade. Alguns problemas são complicados de resolver com algoritmos evolutivos tradicionais (AFRABANDPEY, 2014), por isso a realização de pesquisas por cientistas e pesquisadores de computação para encontrar alternativas para melhorar a resolução de tais algoritmos é fundamental, como por exemplo: estudos na forma de inicialização dos parâmetros e adição de novas etapas, ambas extremamente importantes para proporcionar melhorias no desempenho.

Existem muitos exemplos desse tipo de algoritmo inspirado na natureza, como: Algoritmo *Firefly* (FA) nomeado por Yang (2007) e fundamentado com a bioluminescência dos vaga-lumes, Otimização de Enxame de Partículas (PSO), inspirado no comportamento coletivo de cardumes de peixes e bandos de pássaros proposto por Kennedy e Eberhart (1995) e Otimização Colônia de Formigas (ACO) motivado pelo comportamento de formigas e desenvolvido por Marco Dorigo (1992).

O BA é uma recente edição de algoritmo evolutivo e bioinspirado e desenvolvido por Xin-She Yang (2010), aplicado em áreas de processamento de imagens, mineração de dados e

---

\* Aluno de Graduação em Bach. em Ciência da Computação na Universidade Estadual da Paraíba – Campus VII.  
Email: allanvilcarvalho@gmail.com

em diversos trabalhos de otimização elaborados ao longo dos anos, baseia-se no comportamento dos morcegos, mais precisamente na ecolocalização para detectar distâncias e encontrar presas (no nosso contexto, soluções).

Este trabalho tem como objetivo principal, apresentar novos testes com um número maior de funções *benchmark* e iterações nas novas soluções não aceitas pela nova variante BA proposta por (VILAR, 2016), para executar tarefas de otimização com melhor desempenho comparado a versão padrão, que possui a taxa de convergência lenta e baixa precisão (XIE, 2013).

O esboço deste trabalho consiste, na primeira parte após esta da introdução apresenta uma breve explicação sobre o comportamento dos morcegos. Na segunda parte definição do algoritmo de morcego padrão, relatando as abstrações para sua criação. Na terceira parte consta exemplos de variantes BA. Na quarta apresentação da Nova Variante com ilustração do pseudo-código. Na quinta parte expõe a metodologia utilizada. Na sexta explicação dos experimentos realizados nos algoritmos. Na sétima apresentados os resultados e a análise dos mesmos e por fim na parte oitava a conclusão.

## **2 COMPORTAMENTO DOS MORCEGOS**

Os morcegos são animais que possuem uma aptidão extraordinária chamada ecolocalização, onde utilizam para caçar, desviar de obstáculos no escuro, dizer a forma e a textura de uma presa, saber qual a direção da presa e até a sua velocidade (TOPAL, 2014). Representam 20% das espécies de mamíferos, cerca de 996 espécies e são os únicos mamíferos com asas (YANG, 2010).

A espécie *Microchiroptera* usa de uma forma mais abrangente a ecolocalização, conseguindo emitir pulsos de som, com uma frequência superior à outras espécies. Os pulsos de som duram em torno de 8 a 10 milésimos de segundo e corresponde de 25 kHz a 150 kHz de frequência, enquanto a espécie *Microchiroptera* emite cerca de 10 a 20 pulsos de som podendo chegar a 200 pulsos quando estão perto da presa. Normalmente o morcego quando busca algo sua frequência de emissão do som é baixa diferentemente do comprimento de onda, ambos são valores inversamente proporcionais.

A partir de estudos, a *Microchiroptera* consegue usar o tempo de atraso da emissão e detecção do eco. A diferença de tempo entre suas duas orelhas e as variações de intensidade dos ecos para construir um cenário tridimensional (YANG, 2010). Diante disso, foi desenvolvido o algoritmo de morcego, mostrado na próxima seção.



### 3 ALGORITMO DE MORCEGO PADRÃO

#### 3.1 Teoria

O BA é composto das vantagens de outros algoritmos, a exemplo do PSO e Recozimento Simulado. Foi desenvolvido para fins de otimização (RAGHAVAN, 2015) onde o problema básico é maximizar ou minimizar uma função, como exemplo: função Griewank, da forma mais precisa para encontrar a solução ideal.

Com inspiração na ecolocalização dos morcegos, mais precisamente na espécie *Microchiroptera* onde a ideia base é uma população de morcegos virtuais voando em um ambiente com “x” dimensões aleatoriamente para detectar uma presa (solução), originaram-se as seguintes abstrações para a criação:

- **Abstração 1:** Os morcegos utilizam-se da ecolocalização para medir a distância e “saber” distinguir a presa dos obstáculos;
- **Abstração 2:** Todos os morcegos voam aleatoriamente com velocidade  $v_i$  (velocidade de movimentação no espaço) na posição  $x_i$  (localização no espaço) com uma frequência fixa  $f_{min}$  (frequência dos impulsos emitidos);
- **Abstração 3:** Para localização da presa os morcegos ajustam a frequência  $f_i$  e o volume sonoro  $A_0$  (volume do eco emitido);
- **Abstração 4:** A frequência dos impulsos emitido e a taxa de emissão de pulso  $r_i$  podem ser ajustadas de acordo com a proximidade na presa.

O trecho de viagem dos pulsos do morcego  $i$  pode ser alterado ao se variar a frequência  $f_i$ , que é estabelecida aleatoriamente e de modo uniforme a partir do intervalo  $[f_{min}, f_{max}]$  (PRAJNA, 2014). Toda frequência equivale a um comprimento de onda entre o intervalo  $[\lambda_{min}, \lambda_{max}]$ , que são valores inversamente proporcionais pelo fato que  $\lambda$  e  $f$  estão relacionados, já que  $\lambda f = V$ , normalmente a velocidade do som no ar é  $V = 340$  m/s. Diante das informações citadas podemos afirmar que uma frequência  $[20$  kHz,  $500$  kHz] corresponde a uma gama de comprimento de onda  $[0,7$ mm,  $17$ mm] e quanto mais baixa a frequência maior será o comprimento de onda, diferentemente de uma frequência alta que terá o comprimento de onda menor acarretando em um percurso de onda também menor.

A partir do pseudo-código torna-se relativamente simples a implementação do algoritmo do morcego que tecnicamente apresenta sintonia de frequência e controle de parâmetros (AL-MURAEB, 2016).

O pseudo-código BA é mostrado na Figura 1:

Figura 1 - Pseudo-código do BA.

```
Função Objetivo  $f(x)$ ,  $x = (x_1, x_2, \dots, x_d)$ 
Inicializa a Frequência  $F_i$ , Velocidade  $V_i$ , Posição  $X_i$  e Fitness  $FIT_i$  dos morcegos
Define a taxa de pulso  $R_i$  e volume  $A_i$ 
Localiza e armazena, o melhor Fitness e a melhor Posição
Enquanto (condição)
    Para todos os morcegos
        Gera uma nova solução  $S_i$  atualizando a:
            • Frequência
            • Velocidade
            • Posição
        Se ( $R_i < RAND$ )
            Atualiza a nova solução
        Fim
        Calcula o Fitness da nova solução
        Se ( $(FIT_{new} \leq FIT_i) \& (A_i > RAND)$ )
            Atualiza o morcego  $i$ 
        Fim
        Atualiza o melhor Fitness e a melhor Solução (localização de morcego)
    Fim
Fim
```

Fonte: Elaborada pelo autor.

Como mostra a Figura 1 no início do algoritmo a frequência, velocidade, posição, taxa de pulso e volume são inicializados de forma aleatória e dentro de limites previamente definidos continuando com o armazenamento do melhor *fitness* (parâmetro que define a qualidade do morcego) e melhor posição. Seguinte a inicialização dos parâmetros gera-se uma nova solução a partir da atualização da frequência e da velocidade utilizando-se funções apresentadas na seção seguinte. Em seguida verifica-se a taxa de pulso acarretando na atualização da nova solução. Prosseguindo-se com o cálculo do fitness e finalizando-se com a verificação da qualidade da nova solução responsável pela atualização do morcego.

### 3.2 Morcegos Virtuais

O BA é composto por um conjunto de morcegos virtuais que possuem o objetivo de encontrar algo específico similar ao morcego do mundo real. São inicializados aleatoriamente em um ambiente virtual e voam com uma velocidade definida a partir da frequência dos impulsos emitidos, de acordo com as informações seguintes.

### 3.2.1 Movimentação

No algoritmo algumas etapas são consideradas básicas, sendo elas: atualização da frequência, velocidade e posição no ambiente de busca, como mostrado na Figura 1, utilizando-se as funções a seguir:

$$f_i = f_{min} + (f_{max} - f_{min})\beta, \quad (1)$$

$$v_i^{t+1} = v_i^t + (x_i^t - x_*)f_i, \quad (2)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \quad (3)$$

Onde  $\beta \in [0, 1]$  é um vetor aleatório e  $x_*$  é a melhor solução na iteração  $t$  descoberta após a realização de uma busca entre todas as soluções (YANG, 2014).

Posteriormente, pode ocorrer uma atualização na posição, caso a taxa de pulso  $r_i$  do morcego esteja menor, quando comparada a um valor aleatório. Conseqüentemente se a taxa de pulso for menor a nova solução estará distante da melhor solução global (COELHO, 2012). A função de atualização é mostrada a seguir.

$$x_{new} = x_{old} + \epsilon A^t, \quad (4)$$

Onde  $\epsilon \in [-1, 1]$  é um número aleatório e  $A^t$  é a média dos volumes dos morcegos na iteração  $t$  responsável pelo equilíbrio da exploração (AFRABANDPEY, 2014).

### 3.2.2 Volume e Taxa de Pulso

Para realização da busca por presa pelo morcego a variação do volume  $A_i \in [0, 1]$  e da taxa de pulso  $R_i \in [0, 1]$  é fundamental. O volume normalmente inicia com valor máximo 1 e diminui gradativamente até atingir o valor 0, indicando que chegou ao destino. Tanto o  $A_i$  quanto o  $R_i$  são apenas atualizados quando as novas soluções são melhores, significando que estão se movendo para posições melhores. A seguir funções para atualização:

$$A_i^{t+1} = \alpha A_i^t, \quad (5)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)], \quad (6)$$

Onde  $0 < \alpha < 1$  e  $\gamma > 0$  são valores constantes. Pode-se usar em casos mais simples  $\alpha = \gamma$  (YANG, 2014).

## 4 VARIANTES DO ALGORITMO DE MORCEGO

Após a origem do BA variantes com o objetivo de encontrar uma solução/presa da melhor forma são originadas utilizando de ideias já aplicadas com sucesso e comprovadas utilizando-as nas mais diversas áreas, como engenharias, telecomunicações e indústria. A

seguir exemplos de variantes BA que obtiveram resultados melhores comparado ao algoritmo de morcego padrão.

O CBA (*Chaotic Bat Algorithm*) proposto por Hodayun (2014) baseia-se no “*Chaos*”, sequência de números aleatórios que possuem um elemento de regularidade para inicializar melhor os parâmetros. Diferente de outras variações aleatórias, essa possui uma distribuição específica e é utilizada em aplicações como por exemplo as redes neurais e circuitos não-lineares (AFRABANDPEY, 2014).

Outra variante chamada DABA (*Directed Artificial Bat Algorithm*), proposto por Amr Rekaby (2013), mudou a forma do comportamento dos morcegos e do ambiente de busca que consistem em um conjunto de vetores e pontos visitados alterados apenas quando os morcegos mudam para outro local melhor, (TOPAL, 2014).

Estas variações quando desenvolvidas foram testadas com funções de alta complexidade e comparadas a outros algoritmos a fim de verificar sua qualidade. Uma nova variante proposta é mostrada na Seção 5.

## 5 NOVA VARIANTE BA

Diante da estrutura do algoritmo de morcego proposta por (YANG, 2010) constatou-se que as novas soluções geradas com baixa qualidade estavam sendo descartadas sem realizar nada para melhorá-la após examinar o *fitness* (etapa posterior a verificação da taxa de pulso), então uma nova variante baseada nas idealizações do criador somadas a novas etapas que obtiveram resultados melhores quando comparado a versão padrão, está sendo proposta.

Para o desenvolvimento da nova variante, regras foram adotadas:

- **Regra 1:** Novas soluções iniciar o volume em,  $V_i \in [0.5, 1.0]$  e a taxa de pulso em  $R_i \in [0, 0.5]$ ;
- **Regra 2:** Criar ao fim de cada iteração um conjunto de novas soluções não aceitas (soluções de baixa qualidade).
- **Regra 3:** Atualizar as novas soluções armazenadas.
- **Regra 4:** Realizar atualização do melhor morcego e da melhor solução a cada iteração a partir da melhor solução armazenada.

O pseudo-código da nova variante é mostrado na Figura 2 com as novas etapas destacadas:

Figura 2 - Pseudo-código da nova variante BA.

```
Função Objetivo  $f(x)$ ,  $x = (x_1, x_2, \dots, x_d)$ 
Inicializa a Frequência  $F_i$ , Velocidade  $V_i$ , Posição  $X_i$  e fitness  $FIT_i$  dos morcegos
Localiza e armazena, o melhor fitness e a melhor Posição
Enquanto (condição)
    Para todos os morcegos
        Gera uma nova solução  $S_i$  atualizando a:
            • Frequência
            • Velocidade
            • Posição
        Gerar aleatoriamente dentro dos limites a taxa de pulso  $R_i$  e o volume  $A_i$  da nova solução
        Se ( $R_i < RAND$ )
            Atualiza a nova Solução
        Fim
        Calcula o fitness da nova solução
        Se ( $(FIT_{new} \leq FIT_i) \ \& \ (A_i > RAND)$ )
            Atualiza o morcego  $i$ 
        Senão
        Armazena a nova Solução
        Fim
        Atualiza o melhor fitness e a melhor Solução
    Fim
Executa a nova variante BA nas novas Soluções armazenadas
Atualiza o melhor morcego baseado no melhor morcego Temporário
Atualiza a melhor Solução baseado no melhor morcego Temporário
Fim
```

Fonte: Elaborada pelo autor.

A nova estrutura apresentada na Figura 2, composta de novas etapas ocasionaram novas soluções: volumes mais próximos de 0, taxas de pulso mais próximo de 1 e a atualização das não aceitas (soluções de baixa qualidade), culminando, assim em resultados melhores comprovados na seção 8.

## 6 METODOLOGIA

Inicialmente realizou-se uma revisão bibliográfica com o objetivo de analisar mais profundamente o tema proposto e poder abordar com mais eficácia a produção do referencial teórico e as devidas implicações.

O BA básico possui diversas vantagens, mas a taxa de convergência lenta e de baixa precisão são características negativas do mesmo.

O método de pesquisa experimental foi utilizado, para constatar que o BA mesmo que tenha vantagens visíveis ainda pode ser melhorado. Realizaram-se alterações na quantidade de iterações das novas soluções não aceitas na nova variante BA. Desta forma, possibilitou a realização de novos testes obtendo-se então, uma melhor comparação entre os dois BA's abordados.

## 7 EXPERIMENTOS

Com o objetivo de analisar o desempenho da nova variante proposta, testes comparando a mesma com o BA foram realizados utilizando o software MATLAB e um computador pessoal com a configuração: processador Intel Core i5-2410M de 2.30 GHz e 8 GB de RAM. Nos testes foram utilizadas 1 (uma) função unimodal e 4 (quatro) funções multimodais de *benchmark* para minimizar e encontrar a melhor solução.

A realização de testes utilizando funções de alta complexidade é fundamental nesse trabalho, porque é a partir dos resultados como média e desvio padrão dos resultados finais que conseguiu definir com exatidão a qualidade dos algoritmos. A seguir, na Tabela 1 tem-se as funções com seus respectivos parâmetros.

Tabela 1 – Funções *Benchmark*.

Função	Fórmula	Limite	F(x)
f(x1): Griewank (multimodal)	$f(x) = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600, 600]	0
f(x2): Schwefel (multimodal)	$f(x) = 418.9829D - \sum_{i=1}^D x_i \sin\sqrt{ x_i }$	[-500, 500]	0
f(x3): Sphere (unimodal)	$f(x) = \sum_{i=1}^d x_i^2$	[-100, 100]	0

f(x4): Three-Hump Camel (multimodal)	$f(x) = 2x^2 - 1.05x^4 + \frac{x^6}{6} xy + y^2$	[-5, 5]	0
f(x5): Levy (multimodal)	$f(x) = \sin^2(3\pi x) + (x - 1)^2(1 + \sin^2(3\pi y)) + (y - 1)^2(1 + \sin^2(2\pi y))$	[-10, 10]	0

Fonte: Dados organizados pelo autor.

Alguns parâmetros utilizados nos algoritmos como quantidade de iterações nas soluções não aceitas na nova variante, quantidade de iterações por execução, dimensão das soluções, quantidade de morcegos, intervalo da frequência de pulso e a quantidade de execuções em cada função para cada algoritmo suficiente para realizar uma análise de qualidade em ambos algoritmos são mostrados a seguir.

Tabela 2 – Configurações usadas nos testes.

Parâmetro	Valor
Quant. de iterações nas soluções não aceitas na nova variante	[1, 10]
Quant. de iterações por execução	2000
Dimensão das soluções	50
Quant. de morcegos	40
Intervalo da frequência de pulso	[0, 2]
Quant. de execuções em cada função para cada algoritmo	100

Fonte: Dados organizados pelo autor.

A variação das iterações nas novas soluções não aceitas na nova variante, foi realizada para verificar o impacto que essa nova etapa ocasiona e poder definir uma quantidade de iterações padrão. Já o parâmetro com a quantidade de morcegos foi definido com um valor que pertence ao intervalo [15, 50] suficiente para execução da maioria dos problemas (YANG, 2010). A seguir, apresentação dos resultados e a análise dos mesmos.

## 8 RESULTADOS E ANÁLISE

Comparação das execuções alterando a quantidade de iterações nas novas soluções não aceitas na nova variante é mostrada na Tabela 3, onde Q = quantidade de iterações nas soluções não aceitas, F = função *Benchmark*, M = média das soluções, D = desvio padrão das soluções e E = quantidade de execução do F.

Tabela 3 – Resultados da nova Variante.

Q / F		f(x1)	f(x2)	f(x3)	f(x4)	f(x5)
1	M	426,6511	13078,201	8293,5252	0,023891	40,2667
	D	102,9484	1115,5646	5609,5305	0,081427	13,1729
	E	57481,65	57569,53	57507,45	79619,4	76438,55
2	M	422,9627	13574,0878	11577,5151	0,026877	39,4292
	D	111,0658	903,6409	7370,5039	0,085895	11,6289
	E	50010,16	50091,08	50034	79292,82	74393
3	M	414,3811	13713,9465	13497,5086	0,017918	37,5723
	D	112,8135	1022,9461	6884,3826	0,07128	10,3755
	E	46264,4	46249,97	46234,04	79027,64	73081,97
4	M	428,0341	13735,3027	12485,7005	0,03285	39,1883
	D	107,201	926,5128	7297,1806	0,093912	11,4929
	E	43986,52	44032,24	44000,88	78755,96	71629,04
5	M	417,3177	13954,0337	14263,4764	0,020905	37,3586
	D	99,2452	908,1314	7227,4771	0,076581	10,3538
	E	42484,05	42524,65	42544,4	78525,3	70700,2
6	M	401,7184	13972,7255	14347,9666	0,020905	37,794
	D	126,0862	921,6451	6792,1644	0,076581	12,2583
	E	41547,44	41577,74	41522,54	78314,36	69761,84
7	M	394,9997	14049,596	13399,6257	0,014932	37,8921
	D	99,8128	878,4321	6177,863	0,065415	11,8528
	E	40595,32	40632,35	40574,6	78078,43	68978,5
8	M	417,3149	13887,2767	12668,8676	0,035837	38,2743
	D	102,5532	883,5041	6582,6034	0,097535	11,8926
	E	40051,2	39995,84	39985,6	77838,32	68637,44
9	M	406,1496	14210,977	14120,3855	0,038823	38,4615
	D	108,187	929,1719	6664,0153	0,10094	11,7673
	E	39526,19	39422,15	39527,99	77716,25	67487,39
10	M	408,2776	14103,9632	13775,7923	0,017782	<b>36,0005</b>
	D	<b>98,8234</b>	<b>850,6166</b>	7003,2717	<b>0,06003</b>	11,8886
	E	<b>39227,1</b>	<b>39268,7</b>	<b>39302,7</b>	<b>77510,8</b>	<b>66865,4</b>

Fonte: Dados organizados pelo autor.



Tabela 4 – Resultados do BA.

Função	Média	Desvio Padrão	Quant. de Execuções da Função
$f(x1)$	650,5364	169,3728	80000
$f(x2)$	13317,0306	1034,4073	80000
$f(x3)$	18545,6014	10403,362	80000
$f(x4)$	0,092578	0,13881	80000
$f(x5)$	56,0962	18,0124	80000

Fonte: Dados organizados pelo autor.

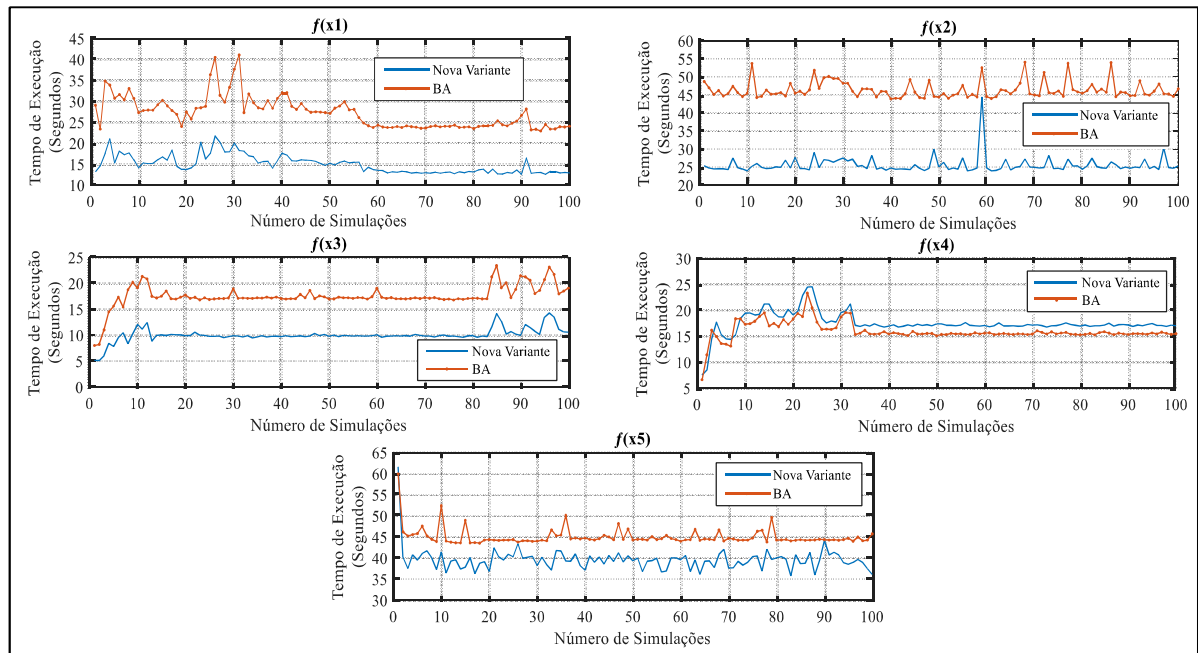
A partir da Tabela 3 podemos verificar as simulações, quando o número de iterações nas soluções não aceitas é aumentada a quantidade de execução da função *Benchmark* é reduzida, em especial as simulações com as funções  $f(x1)$ ,  $f(x2)$  e  $f(x3)$  ocorrem uma maior redução chegando a menos de 50% do total de execuções que o BA padrão executa como podemos ver na Tabela 4, já as simulações com as outras funções a redução é gradativa.

Tratando do  $Q = 10$  verifica-se nas simulações com as funções  $f(x1)$ ,  $f(x2)$  e  $f(x4)$  os melhores desvios padrões resultados que mede a variação das soluções, agora a simulação com a função  $f(x5)$  obteve melhor média das soluções, ficando apenas a simulação com a função  $f(x3)$  com melhoramento só na quantidade de execuções da função e consequentemente melhorias no custo computacional mostrado a seguir.

### 8.1 Avaliação do Desempenho

A nova variante não só ocasionou melhorias na média, no desvio padrão e na quantidade de execução da função, como melhorou também o custo computacional. A Figura 3 mostra o tempo de execução de todas as simulações utilizando  $Q = 10$ , quantidade de iterações que obteve os melhores resultados.

Figura 3 – Tempo de Execução.



Fonte: Elaborada pelo autor.

Pode-se notar que todas as simulações exceto as simulações com a função  $f(x4)$  onde a média do custo computacional: Nova Variante = 17,5465s e BA = 16,0362s com diferença de 1,5103 segundos, obtiveram melhores custos computacionais utilizando  $Q = 10$ . Por essa e pelas outras melhorias mostradas anteriormente que definisse o valor 10 como padrão para a quantidade de iterações nas novas soluções não aceitas na Nova Variante.

## 9 CONCLUSÃO

Este trabalho teve o objetivo de analisar o impacto das novas etapas propostas e inseridas no algoritmo BA, por meio de testes de validação com 5 funções de alta complexidade, além disso, também definir um valor padrão de iterações nas novas soluções não aceitas. Com a análise da média e desvio padrão das soluções e no tempo computacional, verifica-se a qualidade da variante que ocasiona melhores resultados em especial quando realiza 10 iterações nas novas soluções não aceitas definida agora como a quantidade padrão. Assim, conclui-se que a nova variante, agora aprimorada, pode ser aplicada aos problemas do mundo real, como por exemplo, nos problemas de Roteamento de Veículos, Logística, Gestão de Produção e Alocação de Recursos e conseqüentemente, poderá ocasionar melhorias nas resoluções comparadas as então produzidas pelo BA canônico, trabalhos de aplicação considerados futuros.

PROPOSAL FOR A VARIANT BA: ANALYSIS OF SENSITIVITY  
OF THE ADDED STAGES AND PERFORMANCE ASSESSMENT

**ABSTRACT**

This article presents an improved analysis of the proposed new steps and inserted in the bioinspired BA algorithm applied in several works over the years, through the use of 5 benchmark optimization functions extensively used in the validation of algorithms, with the aim of improving it That in the future is applied to real world problems. Besides the variation of the parameter that stores the number of iterations in the new solutions not accepted where it caused the improvement of the new steps and the meeting of a now standard quantity that obtains better resolutions. The evaluations consisted of the mean and standard deviation of the solutions and the computational cost that largely confirm the improvements caused in canonical BA.

**Keywords:** Bat Algorithm. Evolutionary Algorithm. Bio-inspired Algorithm. Optimization Issues.

**REFERÊNCIAS**

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

AL-MURAEB, Ahmed; ABDEL-ATY-ZOHDY, Hoda. Optimal Design of Short Fiber Bragg Grating Using Bat Algorithm With Adaptive Position Update. **IEEE Photonics Journal**, v. 8, n. 1, p. 1-11, 2016.

COELHO, Francisco Carlos Rodrigues et al. Metaheurística inspirada na Ecolocalização de Morcegos: Aperfeiçoamento e Estudo de Casos. In: **Simpósio Brasileiro de Pesquisa Operacional (SBPO)**. 2012.

PRAJNA, K. et al. Application of Bat Algorithm in dual channel speech enhancement. In: **Communications and Signal Processing (ICCSP), 2014 International Conference on**. IEEE, 2014. p. 1457-1461.

RAGHAVAN, S. et al. Bat algorithm for scheduling workflow applications in cloud. In: **Electronic Design, Computer Networks & Automated Verification (EDCAV), 2015 International Conference on**. IEEE, 2015. p. 139-144.

TOPAL, Ali Osman; ALTUN, Oguz. Dynamic virtual bats algorithm (dvba) for global numerical optimization. In: **Intelligent Networking and Collaborative Systems (INCoS), 2014 International Conference on**. IEEE, 2014. p. 320-327.

VILAR, Allan; CANDEIA, Wellington; VALE, Lucas. Aperfeiçoamento do BAT: Inserção de Novas Etapas e Análise de Desempenho. In: **Revista Científica Semana Acadêmica**. 2016.

XIE, Jian; ZHOU, Yongquan; CHEN, Huan. A novel bat algorithm based on differential operator and Lévy flights trajectory. **Computational intelligence and neuroscience**, v. 2013, 2013.

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. **Nature-inspired metaheuristic algorithms**. Luniver press, 2010.

YANG, Xin-She. **Nature-inspired optimization algorithms**. Elsevier, 2014.