

UM BREVE ESTUDO SOBRE OS ALGORITMOS DO VAGA-LUME E DO MORCEGO PARA OTIMIZAÇÃO DE FUNÇÕES DE REFERÊNCIA

F. A. P. Paiva*, I. V. O. Leite, M. H. F. Marccone

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, Campus Parnamirim
fabio.procopio@ifrn.edu.br *

Submetido 23/12/2016 - Aceito 04/04/2018

DOI: 10.15628/holos.2018.5487

RESUMO

Uma alternativa para resolver problemas complexos de otimização é usando algoritmos meta-heurísticos. As meta-heurísticas são métodos aproximados que oferecem soluções aceitáveis para problemas de otimização. Nos últimos anos, algumas meta-heurísticas inspiradas no comportamento de animais da natureza, em especial aqueles que vivem em grupos, vêm sendo bastante utilizadas. Este trabalho tem como objetivo apresentar um breve estudo de duas meta-heurísticas conhecidas como Algoritmo do Vaga-lume e Algoritmo

do Morcego. A comparação é realizada por meio de vários experimentos computacionais para otimizar funções de referência clássicas da literatura. Para isso, foram realizadas 30 execuções independentes, considerando diferentes dimensões e quantidade de avaliações da função objetivo. Após os experimentos, resultados numéricos são apresentados e eles evidenciam a superioridade do Algoritmo do Vaga-lume quando comparado ao Algoritmo do Morcego.

PALAVRAS-CHAVE: Otimização, Meta-Heurística, Inteligência de Enxames, Bioluminescência, Ecolocalização.

A BRIEF STUDY ABOUT FIREFLY AND BAT ALGORITHMS FOR OPTIMIZATION OF BENCHMARK FUNCTIONS

ABSTRACT

An alternative to solve complex problems of optimization is using metaheuristic algorithms. Metaheuristics are approximate methods that provide acceptable solutions for optimization problems. In recent years, some metaheuristics inspired by the behavior of nature animals, especially those that live in groups, have been widely used. This paper aims present a brief study about two metaheuristics known as Firefly Algorithm and Bat Algorithm. The comparison is

performed using several computational experiments to optimize benchmark functions that are classical in the literature. For this, 30 independent runs were performed. They consider different dimensions and quantity of evaluations of the fitness function. After the experiments, numeric results are presented and they show the superiority of the Firefly Algorithm compared to Bat Algorithm.

KEYWORDS: Optimization, Metaheuristic, Swarm Intelligence, Bioluminescence, Echolocation.

1 INTRODUÇÃO

Otimizar consiste no processo de encontrar a melhor solução para um determinado problema que pode ser definido sob um conjunto de restrições. Em geral, as técnicas de otimização são usadas quando não existem alternativas simples para se resolver um problema (Viali, 2011). Quando são usadas em problemas não-lineares e multimodais, muitas técnicas convencionais de otimização não apresentam bons resultados e, por isso, a tendência atual vem sendo a utilização de algoritmos meta-heurísticos inspirados na natureza. Eles têm apresentado resultados bastante promissores.

Para Arroyo (2002), as meta-heurísticas são métodos inteligentes e flexíveis. A flexibilidade está relacionada a uma estrutura formada por componentes genéricos que são adaptados ao problema que se deseja solucionar. Glover e Kochenberger (2006) definem as meta-heurísticas como métodos de solução que coordenam procedimentos de busca locais, com estratégias de alto nível, criando um processo capaz de escapar de mínimos locais e realizar buscas robustas no espaço de soluções de um dado problema.

Os problemas de otimização são comuns em diversas áreas de aplicação do mundo real. Nesse contexto, várias técnicas bioinspiradas baseadas na inteligência de enxames têm sido aplicadas em diversas áreas como sistemas de potência (Yoshida *et al.*, 2000; Duque, L. Oliveira & E. Oliveira, 2012), telecomunicações (Martins, Lins & Silva, 2012), roteamento de veículos (Bell & McMullen, 2004; Bianchi *et al.*, 2006), planejamento de robôs (Ayala, Molin & Coelho, 2009; T. Arora, Gigras & V. Arora, 2014) e em outras áreas.

Um dos conceitos que tem sido bastante difundido está relacionado à Inteligência de Enxames. Esse conceito envolve um conjunto de meta-heurísticas cujos comportamentos emergentes podem resultar em uma capacidade de resolver problemas complexos de engenharia (Silva & Bastos Filho, 2015). Os sistemas baseados na inteligência de enxames são tipicamente constituídos por uma população de agentes simples que interagem, localmente, 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 formigas, vespas, abelhas, grilos, pássaros, peixes, vaga-lumes, morcegos e outros.

Dentre os diversos algoritmos implementados com base na inteligência de enxames, o Algoritmo do Vaga-lume (*Firefly Algorithm*) e o Algoritmo do Morcego (*Bat Algorithm*) foram, recentemente, propostos por Yang (2009) e Yang (2010), respectivamente. O *Firefly Algorithm* (FA) é baseado na característica de bioluminescência dos vaga-lumes e vem sendo usado para resolver problemas na área de controle de tráfego aéreo (Wang *et al.*, 2012), de telecomunicações (Sarma & Gopi, 2014), de sistemas de potência (Balachennaiah, Suryakalavathi & Nagendra, 2015) e de muitos outros. Já o *Bat Algorithm* (BA) é inspirado na ecolocalização dos morcegos a qual é desempenhada durante o voo. Assim como o FA, várias aplicações do algoritmo BA podem ser encontradas e, dentre elas, estão: processamento de imagem (Zhang & Wang, 2012), classificação de dados (S. Mishra, Shaw & D. Mishra, 2012), automação (Reddy & Manoj, 2012) e outras.

Este trabalho apresenta um estudo sobre os algoritmos FA e BA com o objetivo de avaliar os seus desempenhos ao serem usados para otimizar funções de referência, bastante utilizadas por técnicas de otimização. Durante a avaliação dos algoritmos, diversos experimentos computacionais foram realizados. Eles consideram uma variedade de configurações tais como tamanho do enxame, número de avaliações da função objetivo e dimensionalidade do problema. Ao fim dos experimentos, no que diz respeito à qualidade e à estabilidade das soluções encontradas, foi observada a superioridade do algoritmo FA em relação ao BA.

O trabalho está organizado como segue. Na Seção 2, é apresentada uma breve introdução sobre os principais conceitos de Inteligência de Enxames. Essa seção também apresenta os algoritmos do vaga-lume e do morcego e, para finalizar, são apontadas algumas aplicações no mundo real. Na Seção 3, é detalhada a metodologia utilizada no desenvolvimento deste trabalho. Na Seção 4, os resultados e as discussões dos experimentos computacionais realizados são apresentados detalhadamente. E, por fim, na Seção 5, são discutidos os resultados e apresentado o direcionamento de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, inicialmente é apresentado o conceito de Inteligência de Enxames no contexto dos algoritmos de otimização. Nas subseções seguintes, são apresentadas duas meta-heurísticas propostas recentemente. Elas são conhecidas como Algoritmo do Vaga-lume e Algoritmo do Morcego. Por último, são relacionadas algumas aplicações do mundo real que utilizam essas duas meta-heurísticas.

2.1 Inteligência de Enxames

Nas últimas décadas, em especial na área de otimização meta-heurística, surgiram vários algoritmos inspirados na natureza (Yang, 2010). Os algoritmos de otimização bioinspirados são baseados em populações, dentre os quais se destacam os algoritmos evolutivos e os algoritmos de enxames. Os primeiros utilizam técnicas computacionais baseadas nos princípios biológicos evolutivos encontrados na natureza, tais como seleção natural, herança genética, mutação e comportamentos coletivos para intercâmbio de informação (Eiben & Smith, 2003). Já os algoritmos de enxames consideram um conjunto de técnicas baseadas na inteligência de enxames.

O termo Inteligência de Enxames (*Swarm Intelligence*) foi utilizado inicialmente para resolver problemas de sistemas auto-organizados, utilizando células de trabalho robotizadas, visando desenvolver máquinas com inteligência distribuída com capacidades adaptativas e de auto-organização (Hackwood & Wang, 1988; Hackwood & Beni, 1992). Posteriormente, a definição do termo inteligência de enxames foi estendida no intuito de abordar não apenas os sistemas robóticos, mas também qualquer algoritmo ou método de solução de problemas inspirado no comportamento social de espécies animais (Bonabeau, Dorigo & Theraulaz, 1999).

Um enxame pode ser visto como um grupo de agentes cooperando para alcançar algum comportamento proposital e atingir um objetivo (Abraham, Grosan & Ramos, 2007). Sistemas de inteligência de enxames são normalmente constituídos por uma população de agentes simples

capazes de realizar/executar determinadas operações interagindo localmente uns com os outros e com o meio ambiente. Embora, normalmente, não há nenhuma estrutura de controle centralizado ditando como os agentes individuais devem se comportar. As interações locais entre tais agentes, muitas vezes, levam ao surgimento de um comportamento global (Das, Abraham & Konar, 2008).

O exemplo clássico de enxame é o de abelhas, mas a metáfora de um enxame pode ser estendida para outros sistemas com arquitetura similar. Uma colônia de formigas, por exemplo, pode ser entendida como um enxame cujos agentes individuais são formigas. Já um bando de pássaros forma um enxame cujos agentes são os pássaros (Voit, 2010).

Segundo Millonas (1992), para que um algoritmo de enxame apresente um comportamento inteligente, é importante que os seguintes princípios sejam considerados:

1. Proximidade – o enxame deve ser capaz de realizar cálculos simples de espaço e de tempo;
2. Qualidade – o enxame deve ser capaz de responder às características de qualidade do seu ambiente (por exemplo, qualidade da comida);
3. Diversidade – o enxame não deve comprometer a sua atividade ao longo dos canais excessivamente estreitos;
4. Estabilidade – o enxame não deve mudar o seu comportamento por causa de toda e qualquer variação ambiental e;
5. Adaptabilidade – o enxame deve ser capaz de se adequar, quando for necessário, a variações ambientais.

Sendo assim, os algoritmos baseados na inteligência de enxames permitem uma troca eficiente de informações entre os indivíduos de uma colônia. Este tipo de algoritmo fornece características favoráveis para aplicações em sistemas auto-organizados, flexíveis e dinâmicos.

Os algoritmos de inteligência de enxames têm sido aplicados com sucesso em vários domínios. Podem ser citados como exemplos a) roteamento de veículos (Gambardella, Taillard & Agazzi, 1999), b) composição musical (Blackwell & Bentley, 2002), c) análise de *clustering* (Chen & Ye, 2004), d) aplicações de eletromagnetismo (Mikki & Kishk, 2005), e) bioinformática (Correa, Freitas & Johnson, 2006), f) problema do caixeiro-viajante (Souza, 2013), g) treinamento de redes neurais (Zhou & Lin, 2014), h) processamento paralelo (Silva & Bastos Filho, 2015), i) otimização de funções (Paiva, Costa & Silva, 2016; Paiva, Costa & Silva, 2017), entre outros. Alguns modelos de inteligência de enxames que já foram implementados computacionalmente são: Otimização por Enxame de Partículas (Kennedy & Eberhart, 1995), Colônia Artificial de Abelhas (Karaboga, 2005), Otimização por Colônia de Formigas (Dorigo, Birattari & Stützle, 2006) e outros.

2.2 Algoritmo do Vaga-lume

O Algoritmo do Vaga-lume (*Firefly Algorithm – FA*) foi proposto por Yang (2009) e baseia-se na característica de bioluminescência dos vaga-lumes. De acordo com Day (2010), existem cerca de mil espécies de vaga-lumes e a maioria deles produz *flashes* curtos e rítmicos. Eles são capazes

de produzir uma luz fria graças a órgãos fotogênicos situados muito perto da superfície do corpo, atrás de uma carapaça de queratina do exoesqueleto translúcido.

De acordo com Yang (2009), embora a biologia não conheça ainda completamente a utilidade das emissões luminosas dos vaga-lumes, três funções já foram identificadas: a) ferramenta de comunicação e atração para acasalamento; b) isca para atração de eventuais presas e; c) mecanismo de alerta para potenciais predadores, visto que os vaga-lumes possuem um “sabor amargo”.

Para desenvolver o algoritmo inspirado nos vaga-lumes, algumas simplificações são admitidas. Elas podem ser resumidas como: a) presume-se que todos os vaga-lumes têm um único sexo e todos eles serão atraídos um pelo outro; b) capacidade de atração é proporcional ao seu brilho e diminui com o aumento da distância entre eles e; c) caso não exista nenhum vaga-lume mais brilhante que os demais, eles se moverão aleatoriamente. O brilho de um vaga-lume é diretamente afetado por sua aptidão (valor da função objetivo). Deste modo, a intensidade de emissão de luz de um vaga-lume $I(x_i)$ é proporcional à função objetivo $J(x_i)$. Porém, a percepção da intensidade luminosa por um vaga-lume decai em função da distância entre eles, devido à absorção da luz pelo meio. A intensidade percebida por um vaga-lume é dada por (Gomes & Rodrigues, 2011):

$$I(r) = I_0 e^{-\gamma r^2}, \quad (1)$$

onde I_0 é a intensidade da luz emitida; r é a distância Euclidiana entre os vaga-lumes i e j , sendo i o vaga-lume mais brilhante e j o vaga-lume menos brilhante; e γ é o parâmetro de absorção da luz pelo meio (Gomes & Rodrigues, 2011).

A atratividade de um vaga-lume é expressa pela seguinte função (Yang, 2009):

$$\beta(r) = \beta_0 e^{-\gamma r^m}, \text{ com } m \geq 1 \quad (2)$$

onde r é a distância entre dois vaga-lumes, β_0 é a atratividade inicial em $r = 0$ e γ é o coeficiente de absorção que controla o decaimento da intensidade de luz. A distância entre dois vaga-lumes i e j , nas posições x_i e x_j , respectivamente, é dada por (Yang, 2009):

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}, \quad (3)$$

onde $x_{i,k}$ é o k -ésimo componente da coordenada espacial x_i do i -ésimo vaga-lume e d é o número de dimensões. Já a movimentação de um vaga-lume i em direção a outro vaga-lume j mais brilhante é definida por:

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \times \left(\text{rand}() - \frac{1}{2} \right), \quad (4)$$

onde o primeiro termo é a posição atual de um vaga-lume, o segundo termo é usado para representar a atratividade de um vaga-lume e a intensidade de luz vista por vaga-lumes adjacentes, e o terceiro termo é usado para o movimento aleatório de um vaga-lume no caso de não existir nenhum outro vaga-lume mais brilhante; α é um fator de aleatoriedade e $\text{rand}()$ é um valor aleatório gerado a partir de uma distribuição normal (Serapião & Rocha, 2012).

No segundo termo, o parâmetro γ caracteriza a variação da atratividade na execução do algoritmo. Seu valor é fundamental, pois determina a velocidade de convergência do algoritmo. Em teoria $\gamma \in [0, \infty]$, mas para a maioria dos projetos se utiliza $[0.01, 100]$ (Araújo, 2014).

Quando $\gamma \rightarrow 0$, observa-se que a luminosidade não diminui com a distância. Assim, localiza-se facilmente um mínimo local ótimo. Os vaga-lumes se encontram em um ambiente sem ruídos, sem dispersão de luz, onde cada vaga-lume enxerga o outro. No entanto, quando $\gamma \rightarrow \infty$, os vaga-lumes se encontram em um ambiente muito ruidoso, como uma neblina, por exemplo, com extrema dispersão de luz e a atratividade entre dois vaga-lumes tende a zero. Os vaga-lumes não se veem e a técnica torna-se completamente aleatória (Araújo, 2014). O Algoritmo 01 apresenta o pseudocódigo do Algoritmo do Vaga-lume.

```

01: Gera população inicial de vaga-lumes  $x_i$  ( $i = 1, 2, \dots, n$ )
02: Determina a intensidade de luz  $I_i$  em  $x_i$  usando  $f(x_i)$ 
03: Defina o coeficiente de absorção  $\gamma$ 
04: enquanto critério de convergência não for atingido faca
05:   para  $i$  de 1 ate  $n$  faca
06:     para  $j$  de 1 ate  $n$  faca
07:       se ( $I_i < I_j$ ) entao
08:         Mova o vaga-lume  $i$  em direção ao vaga-lume  $j$ 
09:       fimse
10:     Calcule  $\beta(r) = \beta_0 e^{-\gamma r^2}$ 
11:     Avalia novas soluções e atualiza as respectivas intensidades de luz
12:   fimpara
13: fimpara
14: Classifique os vaga-lumes e encontre o melhor
15: fimenquanto

```

Algoritmo 01: Pseudocódigo do algoritmo do vaga-lume.

Inicialmente, a população de vaga-lumes é gerada de forma aleatória (linha 01). Em seguida, a intensidade da luz, que corresponde ao valor da função objetivo, é calculada (linha 02). Na linha 03, o coeficiente de absorção do meio é definido. Caso a intensidade de luz do vaga-lume i for menor que a do vaga-lume j , i é atraído por j (linhas 07 – 09). Na linha 10, o fator de atratividade de um vaga-lume é calculado pela equação $\beta(r) = \beta_0 e^{-\gamma r^2}$, em que r é a distância euclidiana entre os vaga-lumes i e j , β_0 é a atratividade inicial em $r = 0$ e γ é o coeficiente de absorção do meio. Na linha 11, as novas soluções encontradas são avaliadas e as suas respectivas intensidades de luz são calculadas. Por fim, cada vaga-lume é ordenado de acordo com o seu brilho e o que apresentar a maior intensidade de luz é escolhido como a melhor solução.

2.3 Algoritmo do Morcego

O algoritmo proposto por Yang (2010) fundamenta-se no processo de ecolocalização desempenhada pelo bando de morcegos durante o seu voo para fins de detecção de presas e para evitar obstáculos. A ecolocalização é uma sofisticada capacidade biológica utilizada para determinar a distância e a posição de obstáculos em um ambiente através da emissão de ondas

ultrassônicas (Eiben & Smith, 2003). Após a emissão de uma onda ultrassônica, o morcego calcula o tempo de retorno do obstáculo até ele. A partir da diferença entre o tempo de emissão e a reflexão da onda e a variação da amplitude do eco, o morcego se localiza naquele instante de tempo (Altringham, 2011).

Os morcegos possuem este “sentido extra” bem desenvolvido, o que lhes é conveniente, pois são animais de hábitos noturnos e não contam com uma boa visão. Muitas vezes, as presas são pequenos insetos que refletem as ondas emitidas por estes mamíferos. Portanto, é necessário que o comprimento de onda seja da ordem de milímetros, o que implica em altas frequências, na ordem de 20 – 215 kHz.

O algoritmo do morcego possui um número fixo n de morcegos virtuais que se deslocam ao longo das iterações t , sobre a superfície da função objetivo especificada. Cada morcego se localiza em uma posição x_i ($i = 1, 2, 3, \dots, n$) que representa uma possível solução para a função objetivo que se deseja encontrar o ponto ótimo, sendo x_* a melhor solução na iteração t (Coelho, 2013).

De acordo com Yang (2010), algumas premissas são estabelecidas referentes às características da ecolocalização dos morcegos. São elas:

1. Todos os morcegos fazem uso da ecolocalização para perceber e calcular a distância, além de conhecer a diferença entre a presa e as condições limite no espaço em que ele se encontra;
2. Todos os morcegos percorrem o espaço de busca com velocidade v_i , na posição x_i , com uma frequência mínima f_{\min} (ou comprimento de onda λ), variando o comprimento de onda λ (ou frequência f) e amplitude do pulso emitido A_0 ao buscar a presa. Eles podem ajustar automaticamente o comprimento de onda de seus pulsos emitidos e ajustar a taxa de emissão de pulso $r \in [0, 1]$, dependendo da proximidade do alvo;
3. Embora a amplitude possa variar de várias maneiras, assume-se normalmente que a amplitude varia de um valor elevado A_0 a um mínimo valor A_{\min} .

O Algoritmo 02 apresenta o pseudocódigo do algoritmo do morcego. A execução é iniciada com a geração aleatória de morcegos (linha 01), em que cada morcego tem sua própria taxa de emissão de pulso r_i , volume do pulso emitido A_i , frequência f_i e velocidade v_i . Estas soluções e parâmetros são atribuídos aleatoriamente, porém dentro de limites pré-estabelecidos. Na linha 02, a frequência inicial f_i é determinada para a posição x_i . A taxa de pulso r_i e a amplitude A_i são inicializadas (linha 03). As linhas 04 – 16 representam a evolução dos morcegos ao longo do tempo, em que novas soluções são estabelecidas e frequências, velocidades e posições são atualizadas (linha 05). As soluções, que também podem ser entendidas como posições dos morcegos, são avaliadas e classificadas por meio da função objetivo, onde uma lista é criada com as aptidões de cada morcego, de forma que se atribui a posição x_* para o melhor indivíduo.

```

01: Gere a população inicial de morcegos  $x_i$  ( $i = 1, 2, \dots, n$ ) e  $v_i$ 
02: Determine a frequência de pulsos  $f_i$  em  $x_i$ 
03: Inicialize a taxa de pulso  $r_i$  e a amplitude  $A_i$ 
04: enquanto critério de convergência não for atingido faca
05:   Gere novas soluções ajustando a frequência e atualize as velocidades e posições
06:   se ( $rand > r_i$ ) entao
07:     Selecione uma solução dentre as melhores soluções
08:     Gere uma solução local próxima da melhor solução selecionada
09:   fimse
10:   Gere uma nova solução aleatoriamente
11:   se ( $rand < A_i$ ) e ( $f(x_i) < f(x^*)$ ) entao
12:     Aceite as novas soluções
13:     Aumente  $r_i$  e diminua  $A_i$ 
14:   fimse
15:   Classifique os morcegos e encontre o melhor deles
16: fimenquanto

```

Algoritmo 02: Pseudocódigo do algoritmo do morcego.

A cada iteração t , a nova frequência f_i , velocidade v_i^t e posição x_i^t , para cada indivíduo i , são atualizadas de acordo com as equações:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (5)$$

$$v_i^{t+1} = v_i^t + (x_i^t - x_*) \quad (6)$$

$$x_i^{t+1} = x_i^t + v_i^t \quad (7)$$

A frequência de cada morcego é definida no intervalo f_{min} e f_{max} . $\beta \in [0,1]$ é um valor gerado aleatoriamente a partir de uma distribuição normal. Após a atualização dos parâmetros frequência e velocidade, as taxas de emissão de pulso emitidas por cada morcego são verificadas. Na linha 06, a taxa de emissão de pulso r_i é comparada com um valor gerado aleatoriamente a partir de uma distribuição normal. Se r_i for menor que o valor gerado, significa que, provavelmente, o morcego i está distante de sua presa x_i . Assim, executa-se uma busca local sorteando uma solução entre as melhores e, por meio de um pequeno deslocamento, gera-se uma nova solução para o morcego i (linha 08). Na linha 10, é criada uma solução aleatória e, na linha seguinte, algumas condições são avaliadas para que essa solução seja admitida. Em caso positivo, as novas soluções são aceitas (linha 12), a taxa r_i é aumentada e a amplitude A_i diminuída (linha 13). Por fim, os morcegos são classificados e, de acordo com essa classificação, o melhor morcego é selecionado (linha 15).

2.4 Aplicações práticas dos algoritmos do Vaga-lume e do Morcego

Wang *et al.* (2012) propôs uma variante do FA que consiste na troca de informações entre os melhores vaga-lumes durante o processo de atualização da intensidade de luz. A nova abordagem foi aplicada na área de planejamento de veículos aéreos não tripulados e apresentou resultados melhores que outras meta-heurísticas clássicas como *Particle Swarm Optimization*

(PSO), *Differential Evolution* (DE), *Genetic Algorithm* (GA) e *Ant Colony Optimization* (ACO). Com essa modificação, o veículo é capaz de encontrar caminhos seguros para voar, evitando ameaçadas áreas e minimizando o consumo de combustível.

Sarma e Gopi (2014) usaram o FA para resolver o problema de gasto de energia em sensores de rede sem fio. A meta-heurística foi usada para implementar um *cluster* de energia eficiente para redes de sensores sem fio. Para isso, uma função objetivo foi definida a fim de minimizar a distância entre os *clusters*.

Já Balachennaiah *et al.* (2015) aplicaram o FA com o objetivo de minimizar a perda de potência real de uma rede de transmissão de energia. O algoritmo tem a função de otimizar variáveis como, por exemplo, a magnitude de tensão. Quando o desempenho do FA foi comparado ao do Algoritmo Genético, o algoritmo do vaga-lume mostrou-se superior.

Um novo algoritmo modificado dos morcegos foi aplicado por Zhang e Wang (2012) na área de processamento de imagens. A modificação consiste em uma alteração para aplicar mutação nos morcegos durante o processo da atualização de novas soluções.

Mishra *et al.* (2012) também usaram o BA em uma outra área de estudo conhecida como classificação de dados. O algoritmo foi utilizado para atualizar os pesos de um classificador funcional para ligação de redes neurais artificiais e apresentou resultados superiores a outros métodos como o *Functional Link Artificial Neural Network* (FLANN) e o PSO-FLANN.

Reddy e Manoj (2012) usaram a lógica *fuzzy* combinada ao BA na área de automação. Lógica *fuzzy* foi utilizada para identificar as localizações ótimas dos capacitores e o BA foi implementado para encontrar o tamanho ótimo do capacitor a fim de reduzir a perda máxima.

3 METODOLOGIA

Os experimentos computacionais foram executados em um computador que utiliza processador Intel Core i7 com 2,5 GHz de frequência, 8 GB de memória RAM e sistema operacional Windows 10 Pro, 64 bits. Os algoritmos FA e BA foram implementados na linguagem de programação Matlab R2013a e não foram utilizadas técnicas de processamento paralelo.

A fim de avaliar o desempenho dos algoritmos, foram escolhidas seis funções de referência bastante utilizadas na literatura de algoritmos meta-heurísticos. São elas: Esfera, Rosenbrock, Griewank, Rastrigin, Csendes e Schumer Steiglitz. Todas são aplicadas a problemas de minimização. As funções Esfera, Rosenbrock e Schumer Steiglitz são unimodais, ao passo que Griewank, Rastrigin e Csendes são multimodais com vários mínimos locais. Essas funções têm sido aplicadas em vários estudos (Deb, 2011; Amouzgar, 2012; Paiva, Costa & Silva, 2015). As funções usadas nos experimentos são descritas abaixo:

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

$$f_1(x) = \sum_{i=1}^d x_i^2 \quad (8)$$

- Função Rosenbrock – possui o mínimo global em um vale parabólico:

$$f_2(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (9)$$

- Função Griewank – possui vários mínimos locais regularmente distribuídos:

$$f_3(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (10)$$

- Função Rastrigin – é uma função altamente multimodal, no entanto as localizações dos vários mínimos locais são regularmente distribuídas:

$$f_4(x) = \sum_{i=1}^d (x_i^2 - 10\cos(2\pi x_i) + 10) \quad (11)$$

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

$$f_5(x) = \sum_{i=1}^d x_i^6 \left[2 + \text{sen}\left(\frac{1}{x_i}\right) \right] \quad (12)$$

- Função Schumer Steiglitz – é uma função que não possui mínimos locais, apenas o mínimo global:

$$f_6(x) = \sum_{i=1}^d x_i^4 \quad (13)$$

Para cada uma das funções listadas anteriormente, são apresentadas as fronteiras do espaço de busca, a região de inicialização do algoritmo, a solução global e a solução ótima, conforme Tabela 1.

Tabela 1: Características das funções de referência.

Função	Espaço de busca	Faixa de inicialização	Solução global	Solução ótima
f_1 – Esfera	$-5.12 \leq x_i \leq 5.12$	$2.56 \leq x_i \leq 5.12$	$f(x^*) = 0$	$x^* = (0, \dots, 0)$
f_2 – Rosenbrock	$-30 \leq x_i \leq 30$	$15 \leq x_i \leq 30$	$f(x^*) = 0$	$x^* = (1, \dots, 1)$
f_3 – Griewank	$-600 \leq x_i \leq 600$	$300 \leq x_i \leq 600$	$f(x^*) = 0$	$x^* = (0, \dots, 0)$
f_4 – Rastrigin	$-5.12 \leq x_i \leq 5.12$	$2.56 \leq x_i \leq 5.12$	$f(x^*) = 0$	$x^* = (0, \dots, 0)$
f_5 – Csendes	$-1 \leq x_i \leq 1$	$0.5 \leq x_i \leq 1$	$f(x^*) = 0$	$x^* = (0, \dots, 0)$
f_6 – S. Steiglitz	$-100 \leq x_i \leq 100$	$50 \leq x_i \leq 100$	$f(x^*) = 0$	$x^* = (0, \dots, 0)$

Os valores definidos para os parâmetros utilizados nos experimentos foram os mesmos valores padrões usados por Yang (2009) e Yang (2010). No FA, os valores atribuídos aos parâmetros foram: aleatoriedade $\alpha = 0.2$, atratividade $\beta_0 = 1.0$ e coeficiente de absorção de luz no meio $\gamma = 1.0$. Já no BA, os valores atribuídos foram: amplitude $A \in [1, 2]$, taxa de pulso r_i definida no intervalo $[0, 1]$, frequência mínima $f_{min} = 0$ e frequência máxima $f_{max} = 2.0$.

Para cada uma das funções de referência, são definidas 3 configurações que consistem na variação de sua dimensionalidade e do número máximo de iterações. Apenas o tamanho da população é fixado em 20. Quanto à dimensionalidade do problema, ela é definida em 10 dimensões, 20 dimensões e 30 dimensões. Para 10 dimensões, o número de iterações é 1.000; para 20 dimensões, são 2.000 iterações e; para 30 dimensões, o número de iterações é 3.000. Após os experimentos, a melhor solução, a pior solução, a média do valor de *fitness* e o desvio padrão obtidos são registrados. Para cada algoritmo, são realizadas 30 execuções independentes.

4 RESULTADOS E DISCUSSÃO

As Figuras 1a, 1b, 2a, 2b, 3a e 3b apresentam o comportamento dos algoritmos durante o processo de otimização das seis funções de referência escolhidas. Na otimização de f_1 (Figura 1a), a velocidade de convergência do algoritmo BA varia lentamente e não se observam melhorias significativas ao longo das 3.000 iterações. Por outro lado, observa-se que o comportamento de convergência do algoritmo FA ocorre de forma ativa e sem estagnação do enxame, embora a solução ótima não tenha sido encontrada.

Na otimização da função f_2 (Figura 1b), o BA apresenta novamente um comportamento de convergência muito lento e as soluções encontradas são consideradas insatisfatórias para o processo de minimização. Por outro lado, o FA apresenta um bom comportamento de convergência até iteração 1.800, aproximadamente. No entanto, nas iterações seguintes, o algoritmo aparenta um comportamento de estagnação.

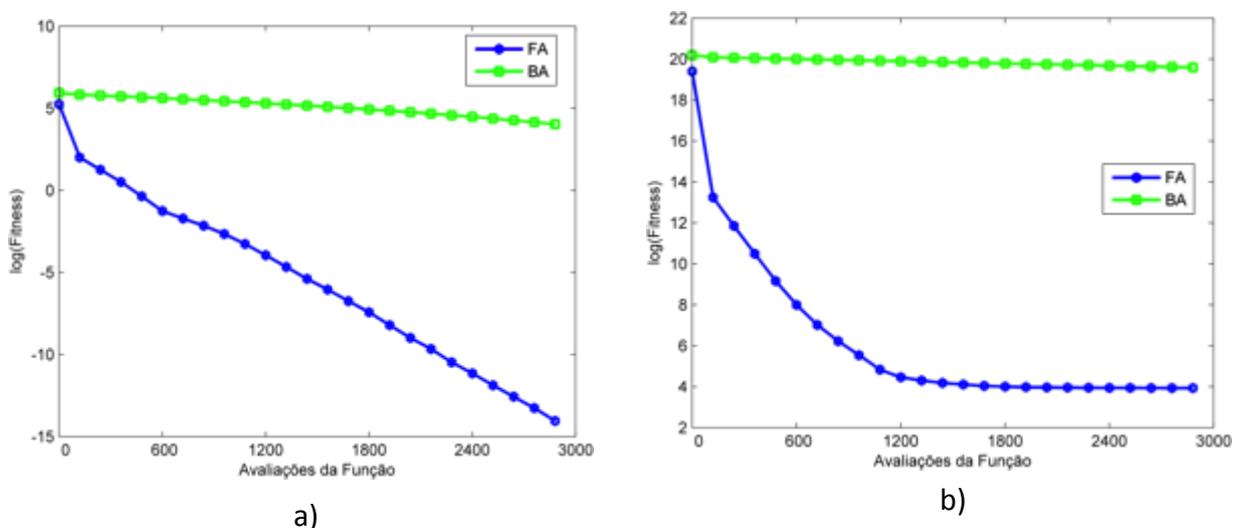


Figura 1: Comportamento de convergência do FA e do BA, 3000 iterações, 30D: a) Função f_1 ; b) Função f_2 .

Quando a função f_3 é otimizada (Figura 2a), mais uma vez, o desempenho do algoritmo BA é inferior quando comparado ao do FA. Vê-se que ele se mantém praticamente estagnado durante todas as iterações, enquanto o FA apresenta um comportamento de convergência ativo e sem estagnação. Já na otimização da função f_4 (Figura 2b), os dois algoritmos apresentaram um comportamento de estagnação, porém o FA obteve resultados melhores que o BA. O algoritmo do morcego estagna, aproximadamente, na iteração de número 500, ao passo que o comportamento de estagnação do algoritmo do vaga-lume inicia-se próximo da iteração 1.800.

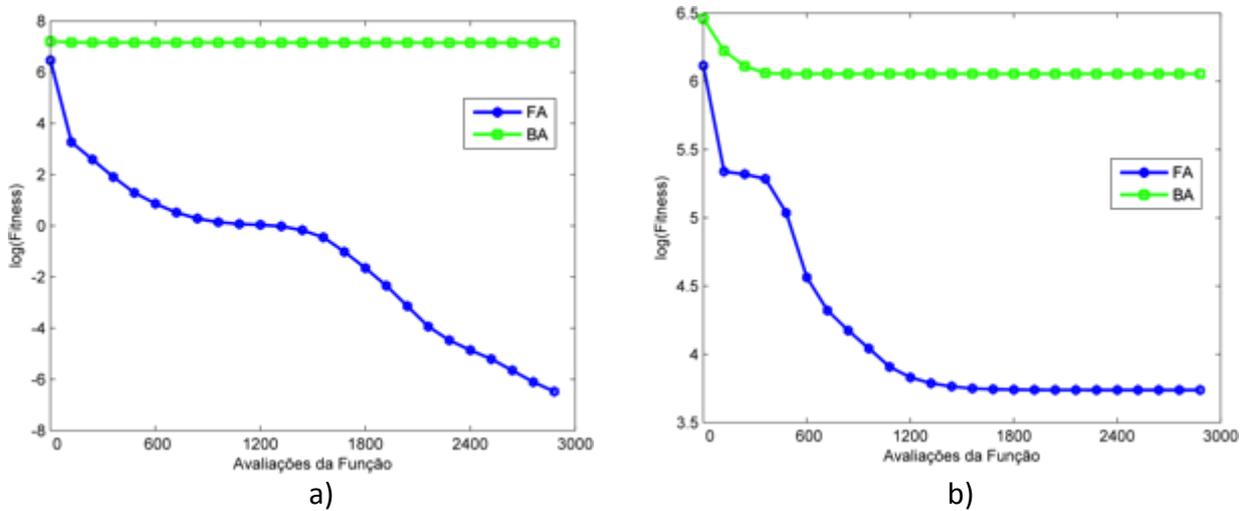


Figura 2: Comportamento de convergência do FA e do BA, 3000 iterações, 30D: a) Função f_3 ; b) Função f_4 .

Para otimizar a função f_5 (Figura 3a), o algoritmo do morcego apresenta uma melhoria nos resultados até a iteração 600, aproximadamente. Em seguida, o algoritmo fica estagnado até o fim das 3.000 iterações. Já o FA tem um comportamento ativo durante todas as iterações e apresentou resultados muito superiores aos do BA. Na Figura 3b, função f_6 , o BA permanece estagnado durante praticamente todas as iterações. Por outro lado, o algoritmo do vaga-lume, mais uma vez, apresenta superioridade quando comparado ao BA.

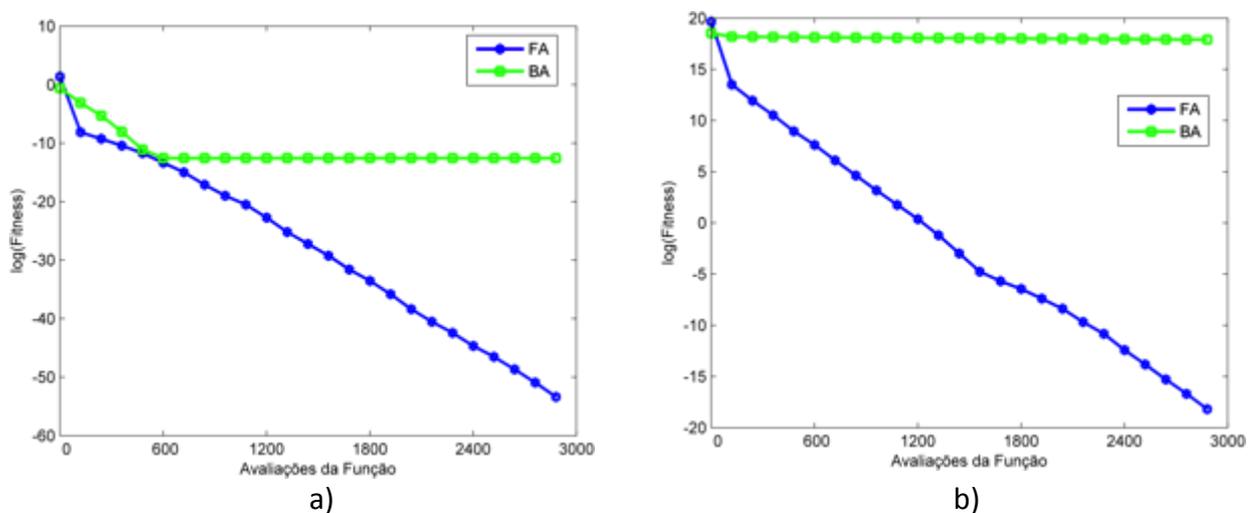


Figura 3: Comportamento de convergência do FA e do BA, 3000 iterações, 30D: a) Função f_5 ; b) Função f_6 .

A Tabela 2 mostra os resultados dos experimentos realizados. Em todas as funções avaliadas, os valores encontrados pelo algoritmo FA (média, desvio padrão, melhor solução e pior

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

Função	Dim	Iter	Algoritmo dos Vaga-lumes			Algoritmo dos Morcegos		
			Média (Desvio)	Melhor	Pior	Média (Desvio)	Melhor	Pior
f_1	10	1000	3,95E-08 (1,18E-08)	1,27E-08	5,89E-08	2,88E+01 (1,30E-01)	7,75E+00	5,21E+01
	20	2000	1,79E-07 (4,07E-08)	1,02E-07	2,63E-07	4,63E+01 (1,53E+01)	1,41E+01	7,66E+01
	30	3000	3,99E-07 (8,99E-08)	2,51E-07	5,94E-07	4,88E+01 (1,55E+01)	1,39E+01	7,44E+01
f_2	10	1000	1,64E+01 (3,68E+01)	3,87E+00	1,89E+02	6,08+E07 (3,65E+07)	7,05+E06	1,52E+08
	20	2000	4,59E+01 (8,08E+01)	1,46E+01	4,34E+02	1,92E+08 (5,19E+07)	3,11E+07	2,72E+08
	30	3000	5,07E+01 (7,54E+01)	2,52E+01	4,20E+02	3,14E+08 (5,26E+07)	2,29E+08	3,92E+08
f_3	10	1000	4,21E-02 (5,39E-02)	3,47E-05	2,21E-01	2,98E+02 (5,88E+01)	1,89E+02	4,37E+02
	20	2000	3,98E-03 (1,05E-02)	1,38E-04	5,40E-02	7,44E+02 (8,37E+01)	5,36E+02	9,20E+02
	30	3000	1,14E-03 (2,82E-03)	2,31E-04	1,26E-02	1,27E+03 (8,18E+01)	1,08E+03	1,41E+03
f_4	10	1000	7,82E+00 (2,78E+00)	2,98E+00	1,19E+01	1,11E+02 (3,09E+01)	5,57E+01	1,56E+02
	20	2000	2,33E+01 (9,01E+00)	8,95E+00	4,17E+01	2,46E+02 (4,54E+01)	1,08E+02	3,05E+02
	30	3000	4,20E+01 (1,15E+01)	2,18E+01	6,66E+01	4,25E+02 (3,52E+01)	3,46E+02	5,16E+02
f_5	10	1000	1,89E-27 (4,74E-27)	2,79E-29	2,47E-26	1,79E-06 (6,26E-07)	2,43E-07	3,54E-06
	20	2000	6,77E-26 (9,25E-26)	1,77E-27	4,08E-25	2,78E-06 (8,33E-07)	8,64E-07	5,06E-06
	30	3000	7,57E-25 (1,19E-24)	6,62E-26	5,81E-24	3,44E-06 (1,05E-06)	1,61E-06	5,71E-06
f_6	10	1000	6,49E-11 (4,98E-11)	1,05E-11	2,28E-10	1,36E+07 (1,09E+07)	3,18E+05	3,73E+07
	20	2000	8,06E-10 (6,35E-10)	2,53E-10	2,65E-09	2,55E+07 (2,64E+07)	4,39E+06	1,38E+08
	30	3000	3,18E-09 (1,70E-09)	1,12E-09	7,21E-09	5,82E+07 (5,16E+07)	1,95E+06	7,21E+08

solução) são melhores que os do algoritmo BA. Os valores do desvio padrão do FA são menores que os do BA, isso mostra que as soluções encontradas pelo algoritmo do vaga-lume são mais próximas do valor médio obtido em cada execução independente. A superioridade do FA também

pode ser evidenciada com a análise dos piores e melhores resultados, em que os piores valores do FA são menores que os dos melhores do BA.

5 CONCLUSÃO

Este trabalho apresentou um breve estudo comparando o desempenho de dois algoritmos, relativamente recentes, que vêm sendo bastante utilizados para resolver problemas de otimização. Eles são inspirados no comportamento de bioluminescência dos vaga-lumes e na ecolocalização dos morcegos e, portanto, são conhecidos como Algoritmo do Vaga-lume e Algoritmo do Morcego.

Para avaliar o desempenho dos algoritmos, vários experimentos computacionais foram realizados utilizando seis funções de referência clássicas da literatura: Esfera, Rosenbrock, Griewank, Rastrigin, Csendes e Schumer Steiglitz. Após os diversos experimentos, observou-se que o Algoritmo do Vaga-lume apresentou melhor desempenho em relação ao do Morcego.

O algoritmo do vaga-lume se destaca por apresentar melhores resultados numéricos e maior velocidade de convergência quando comparado ao algoritmo do morcego. Em todas as funções de referência avaliadas, os resultados apresentados evidenciam a superioridade do algoritmo do vaga-lume, uma vez que as suas piores soluções são mais satisfatórias que as melhores soluções do BA. Além disso, em todos os experimentos, o BA apresentou um comportamento de estagnação antes do FA, embora este último, em alguns experimentos, não conseguiu escapar de ótimos locais.

Como trabalhos futuros, pretende-se criar uma variante do FA a partir da utilização de mecanismos capazes de aumentar a geração de diversidade durante o processo de busca e, conseqüentemente, evitar que o algoritmo fique “preso” em ótimos locais. Uma alternativa que pode ser utilizada com esse objetivo é a combinação do operador de Mutação de Cauchy com uma técnica conhecida como Aprendizagem Baseada em Oposição.

6 REFERÊNCIAS

- Abraham, A., Grosan, C., & Ramos, V. (Eds.). (2007). *Swarm intelligence in data mining* (Vol. 34). Springer.
- Altringham, J. D. (2011). Echolocation and other senses. _____. *Bats: from ecology to conservation*. Reino Unido: Oxford University Press, 2, 61-96.
- Amouzgar, K. (2012). Multi-objective optimization using Genetic Algorithms.
- Araújo, W. C. D. (2014). Síntese de superfícies seletivas de frequência através de técnicas de computação natural.
- Arora, T., Gigras, Y., & Arora, V. (2014). Robotic Path Planning using Genetic Algorithm in Dynamic Environment. *international Journal of Computer Application* (0975-8887), 89(11).
- Arroyo, J. E. C. (2002). Heurísticas e metaheurísticas para otimização combinatória multiobjetivo.

- Ayala, H. V. H., Molin, V., & Coelho, L. (2009). Otimização multiobjetivo usando enxame de partículas de controladores multivariáveis para uma aplicação em robótica. *IX SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE (SBAI)*, [sn], 1-6.
- Balachennaiah, P., Suryakalavathi, M., & Nagendra, P. (2015). Firefly algorithm based solution to minimize the real power loss in a power system. *Ain Shams Engineering Journal*.
- Bell, J. E., & McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18(1), 41-48.
- Bianchi, L., Birattari, M., Chiarandini, M., Manfrin, M., Mastrolilli, M., Paquete, L., Rossi-Doria, O., & Schiavinotto, T. (2006). Hybrid metaheuristics for the vehicle routing problem with stochastic demands. *Journal of Mathematical Modelling and Algorithms*, 5(1), 91-110.
- Blackwell, T. M., & Bentley, P. (2002, May). Improvised music with swarms. In *Proceedings of IEEE Congress on Evolutionary Computation* (Vol. 2, pp. 1462-1467).
- Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems* (No. 1). Oxford university press.
- Chen, C. Y., & Ye, F. (2004). Particle swarm optimization algorithm and its application to clustering analysis. In *Networking, Sensing and Control, 2004 IEEE International Conference on* (Vol. 2, pp. 789-794). IEEE.
- Coelho, F. C. R. (2013). Alocação de geração distribuída em sistemas de distribuição de energia elétrica via otimização bioinspirada na ecolocalização de morcegos.
- Correa, E. S., Freitas, A. A., & Johnson, C. G. (2006, July). A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation* (pp. 35-42). ACM.
- Das, S., Abraham, A., & Konar, A. (2008). Swarm intelligence algorithms in bioinformatics. In *Computational Intelligence in Bioinformatics* (pp. 113-147). Springer Berlin Heidelberg.
- Day, J. C. (2010). An Introduction to Fireflies and Glowworms.
- Deb, K. (2011). Multi-objective optimisation using evolutionary algorithms: an introduction. In *Multi-objective evolutionary optimisation for product design and manufacturing* (pp. 3-34). Springer London.
- Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., & Winfield, A. (Eds.). (2008). *Ant Colony Optimization and Swarm Intelligence: 6th International Conference, ANTS 2008, Brussels, Belgium, September 22-24, 2008, Proceedings* (Vol. 5217). Springer.
- Duque, F. G., Oliveira, L. W., & Oliveira, E. J. (2012). Alocação de bancos de capacitores em sistemas de distribuição utilizando técnica de otimização bio-inspirada. In *XIX Congresso Brasileiro de Automática (CBA)*.
- Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing* (Vol. 53). Heidelberg: springer.
- Gambardella, L. M., Taillard, É., & Agazzi, G. (1999). MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows.
- Glover, F. W., & Kochenberger, G. A. (Eds.). (2006). *Handbook of metaheuristics* (Vol. 57). Springer Science & Business Media.

- Gomes, Rodrigues. Algoritmo Meta-Heurístico de Vaga-Lume para a Otimização Estrutural em Tamanho e Forma com Restrições de Frequências Naturais. 2011. Curso de Engenharia Mecânica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011.
- Hackwood, S., & Beni, G. (1992, May). Self-organization of sensors for swarm intelligence. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on* (pp. 819-829). IEEE.
- Hackwood, S., & Wang, J. (1988, August). The engineering of cellular robotic systems. In *Intelligent Control, 1988. Proceedings., IEEE International Symposium on* (pp. 70-75). IEEE.
- Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization* (Vol. 200). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
- Kennedy, J.; Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*.
- Martins, S. R., Lins, H. W., & Silva, C. R. (2012, August). A self-organizing genetic algorithm for UWB microstrip antenna optimization using a machine learning technique. In *International Conference on Intelligent Data Engineering and Automated Learning* (pp. 642-649). Springer Berlin Heidelberg.
- Mikki, S. M., & Kishk, A. A. (2005, July). Investigation of the quantum particle swarm optimization technique for electromagnetic applications. In *IEEE antennas and propagation society international symposium* (Vol. 2, p. 45). IEEE; 1999.
- Millonas, M. M. (1992). *Swarms, phase transitions, and collective intelligence* (No. LA-UR-92-3980; CONF-9206329--1). Los Alamos National Lab., NM (United States).
- Mishra, S., Shaw, K., & Mishra, D. (2012). A new meta-heuristic bat inspired classification approach for microarray data. *Procedia Technology*, 4, 802-806.
- Paiva, F. A. P., Costa, J. A. F., & Silva, C. R. M. (2015). Uma meta-heurística alternativa de inteligência de enxames baseada em serendipidade guiada. Em '2nd LA-CCI (Latin American) and 12th CBIC Brazilian Congress on Computational Intelligence'.
- Paiva, F. A. P., Costa, J. A. F., & Silva, C. R. M. (2016). A serendipity-based pso approach to delay premature convergence using scout particle. *International Journal of Innovative Computing, Information and Control*, 12(4). 1349-4198.
- Paiva, F. A. P., Costa, J. A. F., & Silva, C. R. M. (2017). A Serendipity-Based Approach to Enhance Particle Swarm Optimization Using Scout Particles. *IEEE Latin America Transactions*, 15(6), 1619-1624.
- Reddy, V. U., & Manoj, A. (2012). Optimal capacitor placement for loss reduction in distribution systems using bat algorithm. *IOSR journal of Engineering*, 2(10), 23-27.
- Sarma, N. V. S. N., & Gopi, M. (2014). Implementation of Energy Efficient Clustering Using Firefly Algorithm in Wireless Sensor Networks. *International Proceedings of Computer Science and Information Technology*, 59, 1.
- Serapião, A. B., & Rocha, R. K (2012). Algoritmos de otimização bioinspirados baseados em populações para o problema de despacho econômico de carga. Anais do XIX Congresso Brasileiro de Automática.

- Silva, E. H., & Bastos Filho, C. J. (2015). PSO Efficient Implementation on GPUs Using Low Latency Memory. *IEEE Latin America Transactions*, 13(5), 1619-1624.
- Souza, G. R. D. (2013). Arquitetura multiagente baseada em nuvem de partículas para hibridização de metaheurísticas.
- Viali, L. (2011). *Métodos quantitativos: estatística e matemática aplicadas*.
- Voit, J. (2010). *Otimização por enxame de partículas com congregação passiva seletiva* (Doctoral dissertation, Tese (Doutorado) —Universidade Federal do Rio de Janeiro, Brasil).
- Wang, G., Guo, L., Duan, H., Liu, L., & Wang, H. (2012). A modified firefly algorithm for UCAV path planning. *International Journal of Hybrid Information Technology*, 5(3), 123-144.
- Yang, X. S. (2009, October). Firefly algorithms for multimodal optimization. In *International Symposium on Stochastic Algorithms* (pp. 169-178). Springer Berlin Heidelberg.
- Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74). Springer Berlin Heidelberg.
- Yoshida, H., Kawata, K., Fukuyama, Y., Takayama, S., & Nakanishi, Y. (2000). A particle swarm optimization for reactive power and voltage control considering voltage security assessment. *IEEE Transactions on power systems*, 15(4), 1232-1239.
- Zhang, J. W., & Wang, G. G. (2012). Image matching using a bat algorithm with mutation. In *Applied Mechanics and Materials* (Vol. 203, pp. 88-93). Trans Tech Publications.
- Zhou, F., & Lin, X. (2014, August). Research of Training Feedforward Neural Networks Based on Hybrid Chaos Particle Swarm Optimization-Back-Propagation. In *International Conference on Intelligent Computing* (pp. 41-47). Springer International Publishing.