Revista brasileira de iniciação científica
ISSN: 2359-232X

CONTROLANDO POPULAÇÃO INICIAL DE ALGORITMOS GENÉTICOS PARA A OTIMIZAÇÃO DE FUNÇÕES.

CONTROLLING THE INITIAL POPULATION OF GENETIC ALGORITHMS FOR FUNCTIONS OPTIMIZATION.

CONTROLANDO LA POBLACIÓN INICIAL DE ALGORITMOS GENÉTICOS PARA LA OPTIMIZACIÓN FUNCIONES.

Denis Tavares da Silva[1]
Carlos Henrique da Silva Santos[2]

Resumo: O desenvolvimento de algoritmos inspirados na natureza têm sido bastante aplicados em problemas de engenharia, destacando-se aqui os que são do tipo como enxame de partículas. Este trabalho apresenta alguns resultados de desempenho de um subtipo baseado em enxame de abelhas, pois têm sido utilizados em diferentes problemas de engenharia e também no controle da geração da população inicial de Algoritmos Genéticos. Assim, são apresentados resultados de desempenho desses algoritmos *Bee Swarm* (BS), comparando-os com os duas propostas de integração desses algoritmos, uma com o BS controlando a população inicial do GA e a outra em que o GA controla a população inicial do BS. Estando esta última versão entre os com melhores resultados nas buscas e retornando melhores resultados com menores custos computacionais. Esses resultados de desempenho são importantes para se economizar recursos computacionais quando aplicados em problemas de engenharia com métodos numéricos que requerem maior poder de processamento e armazenamento de dados.
Palavras-chave: *Algoritmo Genético, Telecomunicações. Inteligência Artificial.*

Abstract: The nature inspired algorithms development have extensively been applied in engineering problems, among them the particle swarm category based algorithms. This work presents some benchmarking results from a subtype designated by particle swarm algorithms, because it has been applied in a wide number of engineering problems and they are also applied to control the initial population generation of the Genetic algorithm. Therefore, some benchmarking results of these Bee Swarm (BS) results comparing to two integrations version of them, the first one where the BS controls the generation of the GA initial population and second where the GA controls the BS initial population. This last version ranked among the best search space results, returning best solutions with one of the lowest computational costs. These are very important performance results to save computational resources when applied in numerical methods for engineering problems that require more data processing and storage powerful.
Keywords: *Genetic Algorithm, Telecommunications, Artificial Intelligence*

[1] Licenciado em Física, pelo IFSP – Itapetininga/SP. E-mail: denis_tvrs95@gmail.com
[2] Professor do Curso de Licenciatura em Física, IFSP – Itapetininga/SP. E-mail: carlos.santos@ifsp.edu.br.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Resumen: El desarrollo de algoritmos inspirados en la naturaleza ha sido muy aplicado en problemas de ingeniería, destacándose aquí los que son del tipo como enjambre de partículas. Este trabajo presenta algunos resultados de desempeño de un subtipo basado en enjambre de abejas, pues se han aplicado en diferentes problemas de ingeniería y también en el control de la generación de la población inicial de Algoritmos Genéticos. Por lo tanto, se presentan resultados de rendimiento de estos algoritmos Bee Swarm (BS), comparándolos con las dos propuestas de integración de estos algoritmos, una con BS controlando la población inicial del GA y la otra en la que el GA controla la población inicial del BS . Estando esta última versión entre los con mejores resultados en las búsquedas y retornando mejores resultados con menores costos computacionales. Estos resultados de rendimiento son importantes para ahorrar recursos computacionales cuando se aplican a problemas de ingeniería con métodos numéricos que requieren mayor poder de procesamiento y almacenamiento de datos.
Palabras clave: *Algoritmo Genético, Telecomunicaciones, Inteligencia artificial*.

Envio 06/04/2018 **Revisão 16/04/2018** **Aceite 12/02/2019**

Revista brasileira de iniciação científica
ISSN: 2359-232X

INTRODUÇÃO

A computação natural é uma área na Inteligência Artificial e vêm sendo vastamente explorada em problemas de Engenharia (De Castro, 2006), (Da Silva; Cerqueira; Silva-Santos, 2013). Nesta, faz-se uso de comportamentos observados na natureza por diversos tipos de animais ou seres vivos, seja na parte da organização social que os mesmos trazem, seja em comportamentos específicos que são essenciais para a sobrevivência destes seres em seu habitat natural, para a construção de códigos que consigam efetivamente imitar estes comportamentos, estes são denominados algoritmos ou metaheurísticas bio-inspiradas (Santos; et al., 2010).

Ainda, nesta linha de metaheurísticas, seu uso justifica-se pela simplicidade algorítmica, flexibilidade de uso em diferentes aplicações (Silva; et al., 2014)(Silva-Santos; Rodríguez-Esquerre; Hernández-Figueroa, 2015), (Silva-Santos; et al., 2018) resoluções de problemas mono e multi-objetivo (Brianeze; et al., 2008), implementações para computação de alto desempenho (Santos; et al., 2010) e na possibilidade de integração entre diferentes técnicas para se buscar melhores resultados e, ou, redução do esforço computacional (Da Silva, Ferreira; et al., 2018).

Dentro desta categoria de algoritmos, um deles tem encontrado crescente interesse na literatura devido a sua facilidade de implementação e a seu alto desempenho quando comparado com outros algoritmos de otimização comumente estudados tais como o PSO (*Particle Swarm Optimization*) e o DE (*Differential Evolution*). Ele é conhecido como ABC (*Artificial Bee Colony*). Hancer, et al. traz uma extensa discussão acerca das vantagens e desvantagens do ABC quando aplicado em diferentes campos para diferentes propósitos, de maneira que possamos entender os porquês que permitem que o ABC continue interessante a pesquisadores das mais diversas áreas do conhecimento.

Segundo Kumar D. e Kumar B. (2013), há algumas características do ABC que os tornam interessantes sob as perspectivas computacionais e de otimização. Há de se destacar os poucos parâmetros de controle requeridos pelo algoritmo, tais quais; o tamanho da população, e o número máximo de ciclos a serem realizados; a flexibilidade e simplicidade do código base, o que possibilita a hibridização e alteração do algoritmo de maneira eficaz, e a eficiente convergência em diferentes otimizações.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Por essas características, optou-se em desenvolver variações desse algoritmo para realizar comparações de desempenho, assim como combinações (*hibridization*) com uma versão de Algoritmo Genético (AG) para se controlar a geração da população inicial. Tudo isso, possibilitou que sete variações do ABC, com ou sem o AG, fossem implementadas para a realização de testes de desempenho na convergência de otimizações de funções de referência encontradas na literatura.

Para desenvolver o código, foi escolhido o ambiente Scilab, devido à sua facilidade de entendimento e fácil "tradução" para outras linguagens computacionais, ser gratuito, possuir recursos para visualização de dados e atender as demandas computacionais para o processamento numérico desses algoritmos, tanto da perspectiva de processamento quanto da de armazenamento dos dados.

ALGORITMO ABC

Proposto por Karaboga (2005), o ABC busca simular o comportamento inteligente de um enxame de abelhas. Para a realização dos estudos propostos neste trabalho, partimos da forma base do ABC. Este algoritmo encontra-se em uma categoria denominada como *swarm algorithms* ou "algoritmos de enxame".

> Há muitos tipos de 'enxames' no mundo. Mas não é possível chamar todos eles de "inteligentes", pois o nível desta inteligência pode variar de acordo com o enxame. A auto-organização é essencial para um sistema de enxame que resulta num comportamento coletivo através de interações locais por agentes simples (Bonabeau, et al. 1999) (tradução livre realizada pelos autores).

Pode-se considerar que para um *swarm*, é necessário que a informação transmitida e gerada por e através dele tenha um grau de organização interna suficiente de forma a orientar e classificar os agentes do enxame. Ainda, segundo Bonabeau (1999), outra característica importante em um enxame é sua capacidade de realizar, simultaneamente, tarefas organizadas por um sistema de trabalho definido, ou seja, a presença de uma estrutura organizacional é essencial para este tipo de algoritmo.

O comportamento simulado pelo ABC é realizado por abelhas na natureza e é conhecido como "forrageamento", que, simplificadamente, é o que dita se uma colmeia terá

Revista brasileira de iniciação científica
ISSN: 2359-232X

fontes de comida suficientes para continuar viva ou não. Para que tal fato ocorra de maneira efetiva, as abelhas precisam se comunicar, provendo à colmeia, as informações necessárias para que sempre haja fontes de comida novas disponíveis, caso alguma explorada atualmente, venha a se esgotar.

Desta maneira, uma hierarquia é criada onde as abelhas são separadas em três categorias: As abelhas campeiras que explorarão as fontes de comida escolhidas; as abelhas observadoras que checarão quão boas são as fontes de comida existentes; e as abelhas exploradoras, que são aquelas que buscarão globalmente por novas fontes de comida (Karaboga, 2005).

Esta hierarquia criada é flexível justamente pelo fato de que uma abelha pode assumir diferentes papéis de acordo com a necessidade da situação. Uma abelha campeira que explora uma fonte de comida esgotada passa a ser, neste momento, exploradora e assim, inicia uma busca local naquela região por novas fontes.

O ABC estabelece variáveis que assumem o papel destas abelhas, de forma que com base nas fontes conhecidas (população inicial), tenta-se minimizar o valor das melhores soluções de forma que ao fim das iterações de busca, tenhamos as melhores soluções possíveis para a função estudada.

Vale a pena ressaltar que um comportamento de abandono deve ser adotado pelo código para os casos nos quais as soluções obtidas não consigam ser otimizadas além de certo limite. Tal comportamento é descrito a seguir:

> Se a rentabilidade das fontes de comida não pode ser mais aumentada e a quantidade de ciclos em que a mesma não foi alterada é maior que o valor predeterminado de testes, então chamado de “limite”, tais soluções serão abandonadas pelas abelhas campeiras. Então, novas soluções serão buscadas. (Kumar. D; Kumar. B, 2013) (tradução livre realizada pelos autores).

Para que este comportamento seja feito de maneira a não comprometer o resultado final das soluções, da convergência dos valores gerados pelo algoritmo, um padrão de comparação deve ser estabelecido, de forma que o código possa, com base nos dados das iterações anteriores, comparar a qualidade das soluções existentes com as novas encontradas pelas abelhas observadoras.

Revista brasileira de iniciação científica
ISSN: 2359-232X

> Depois que cada solução candidata é produzida e avaliada pela abelha artificial, seu desempenho é comparado com a solução antiga. Se a nova fonte de comida possuir melhor qualidade que a fonte antiga, a velha é substituída pela nova. Caso contrário, a solução antiga é mantida. (Weifeng, et al. 2012) (tradução livre realizada pelos autores)

Ao longo dos anos, algumas variações para o ABC foram propostas por diferentes autores, seja na tentativa de aplicar o algoritmo em sistemas restringidos, nos quais devido à natureza intrínseca das variáveis do sistema o algoritmo encontra dificuldades em otimizar as soluções encontradas; ou simplesmente na tentativa de melhorar o desempenho do ABC como um todo.

Pensando nisso, uma das tentativas de aprimoramento discutidas na literatura é o da inicialização da população inicial de algoritmos de otimização bio-inspirados. A importância de tal fato se dá por (Gao; Liu; Huang, 2012):

> A inicialização da população inicial é uma tarefa crucial em algoritmos evolucionários, pois ela pode afetar a velocidade de convergência e a qualidade da solução final. (Weifeng, et al. 2012) (tradução livre realizada pelos autores)

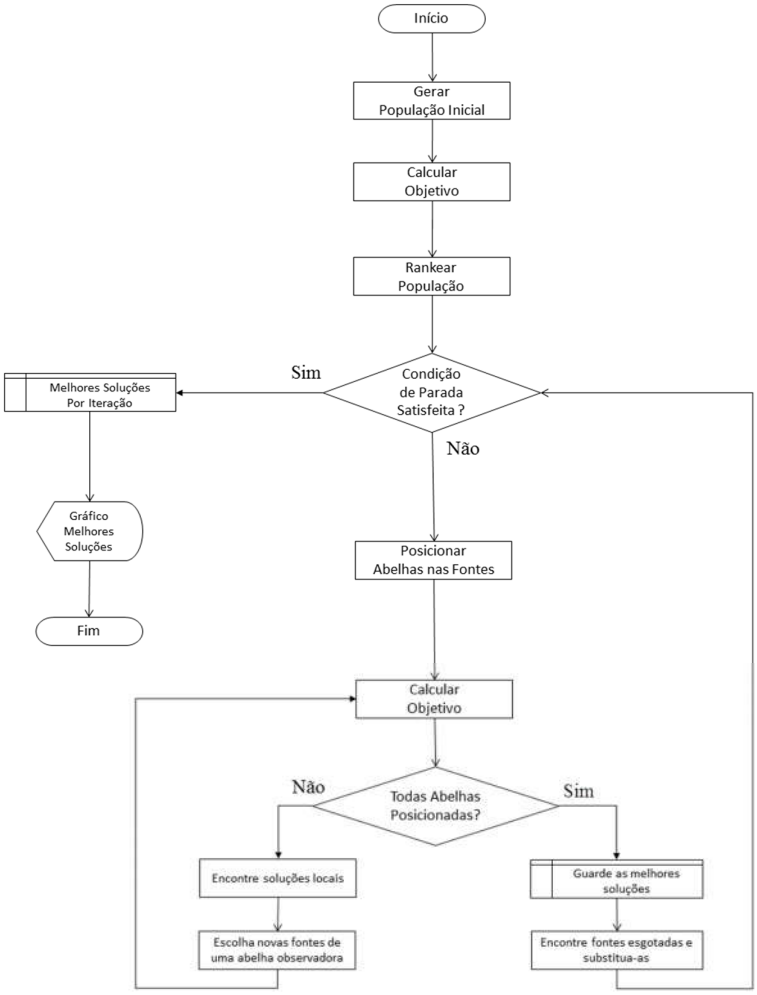
Levando em conta todos esses pontos elucidados e baseados em autores interessantes da literatura, passou a ser viável implementar diferentes variações do ABC, incluindo duas versões híbridas com Algoritmos Genéticos, valendo-se do controle da geração da populacional para analisar se a adoção dessas técnicas reduz a quantidade necessária de iterações para convergência aos resultados esperados para cada problema.

FUNCIONAMENTO DO ABC

O ABC pode ser separado em algumas etapas básicas para um melhor entendimento de seu funcionamento, iniciando com a geração da população inicial e a avaliação das soluções candidatas geradas, conforme fluxograma apresentado na Figura 1 que esquematiza esse funcionamento.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Figura 1 – Fluxograma do funcionamento do ABC, segundo a ideia base proposta por Karaboga (2005).



Ainda, conforme esquematizado na Figura 1, o funcionamento desse ABC segue os trabalhos de Karaboga (2005), que serão elucidados sequencialmente a seguir. Ainda, segundo uma análise desse código da Figura 1 e também as discussões levantadas por Karaboga (2005), pode-se estabelecer algumas relações de modo a melhor entender os processos fundamentais para o funcionamento deste algoritmo.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Inicialização da população

Nesta primeira etapa o algoritmo gera população inicial que é o ponto de partida para a inicialização da busca por melhores soluções. Para tal, uma matriz vazia (1) é preenchida com números aleatórios que variam de acordo com os valores máximos e mínimos preestabelecidos pelo problema proposto para representar cada uma das abelhas.

$$Bee = \begin{bmatrix} bee_{1,1} & \dots & bee_{1,m} \\ : & : & : \\ bee_{n,1} & \dots & bee_{n,m} \end{bmatrix} \mid \{m, n \in \quad , bee_{i,j} \in \quad \} \qquad (1)$$

Assim, *Bee* é uma matriz $m_x n$, sendo m a quantidade de atributos necessários para representar cada uma das abelhas e n a quantidade de abelhas existentes no exame (conjunto de soluções candidatas).

O enxame é modificado ao longo de certo número de iterações (gerações) que é encerrada pela satisfação de uma condição de parada. Ao longo dessas iterações são armazenados os valores das melhores soluções candidatas para um levantamento estatístico e manutenção desses dados até a melhor solução encontrada. Essa busca deve ocorrer com a menor quantidade possível de iterações para a convergência até a solução, por este motivo, técnicas de manipulação da população inicial têm sido exploradas na literatura, como um meio de aperfeiçoar de uma maneira mais efetiva os algoritmos de busca atuais sem que o custo computacional dos mesmos se torne inviável (Karaboga, 2005).

Avaliação dos valores iniciais

Aqui se estabelece um *ranking* que categoriza as soluções preferíveis ao algoritmo de acordo com uma equação que calcula o fitness de cada um destes valores. A cada uma dessas soluções é atribuído um valor probabilístico proporcional ao *fitness* calculado nas fases iniciais. Esta probabilidade associada será o fator que dirá quão rápida as fontes de alimento serão exploradas pelas abelhas, ou seja, este valor serve como um fator de seleção que garante a variabilidade das soluções exploradas.

No algoritmo básico do ABC, uma seleção com base num esquema de "roleta" é aplicada de forma a selecionar potenciais soluções de alta

Revista brasileira de iniciação científica
ISSN: 2359-232X

> qualidade, mas também, para dar chance às soluções de baixa qualidade de serem escolhidas. (Karaboga, 2016, tradução nossa).

Manipulação na hierarquia das "abelhas"

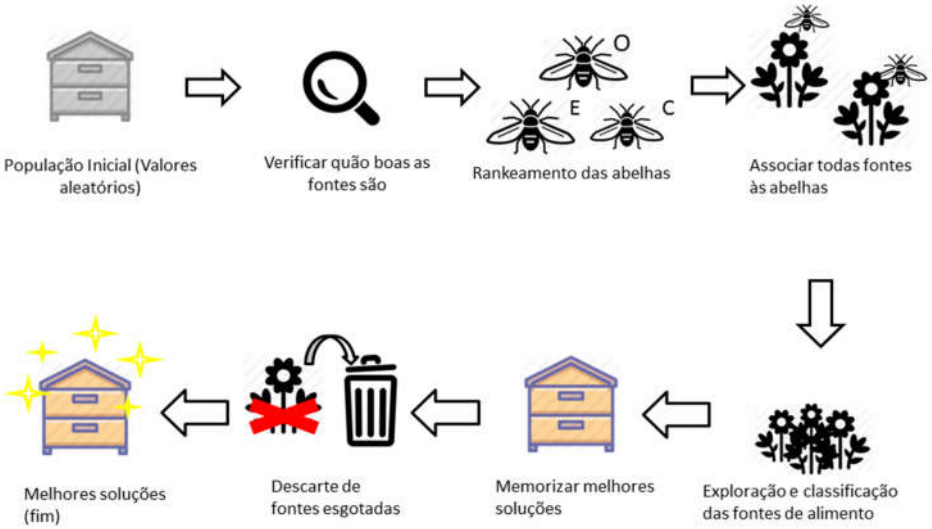
Com base na etapa de rankeamento, ou avaliação inicial das soluções disponíveis, descartam-se os piores valores no sistema, e desta maneira, as "abelhas" são rearranjadas de forma que não fiquem ociosas, ou seja, se uma abelha campeira teve a solução associada a ela, considerada inviável, a mesma muda de categoria e torna-se exploradora de modo a buscar melhores soluções.

Busca de novas soluções

Após a etapa de mudança na hierarquia, as "abelhas" procuram melhores soluções de modo a minimizar os valores encontrados. Esta novas soluções são denominadas *Best Cost* e são memorizadas pelo código de modo a sempre melhorar as melhores soluções presentes no sistema.

Estas etapas podem ser sintetizadas em um esquemático apresentado na Figura 2, que tem como proposta trazer uma visualização que permita o entendimento dos passos mais cruciais para o funcionamento do código de uma maneira mais didática.

Figura 2 – Esquemático base do funcionamento do ABC.



Revista brasileira de iniciação científica
ISSN: 2359-232X

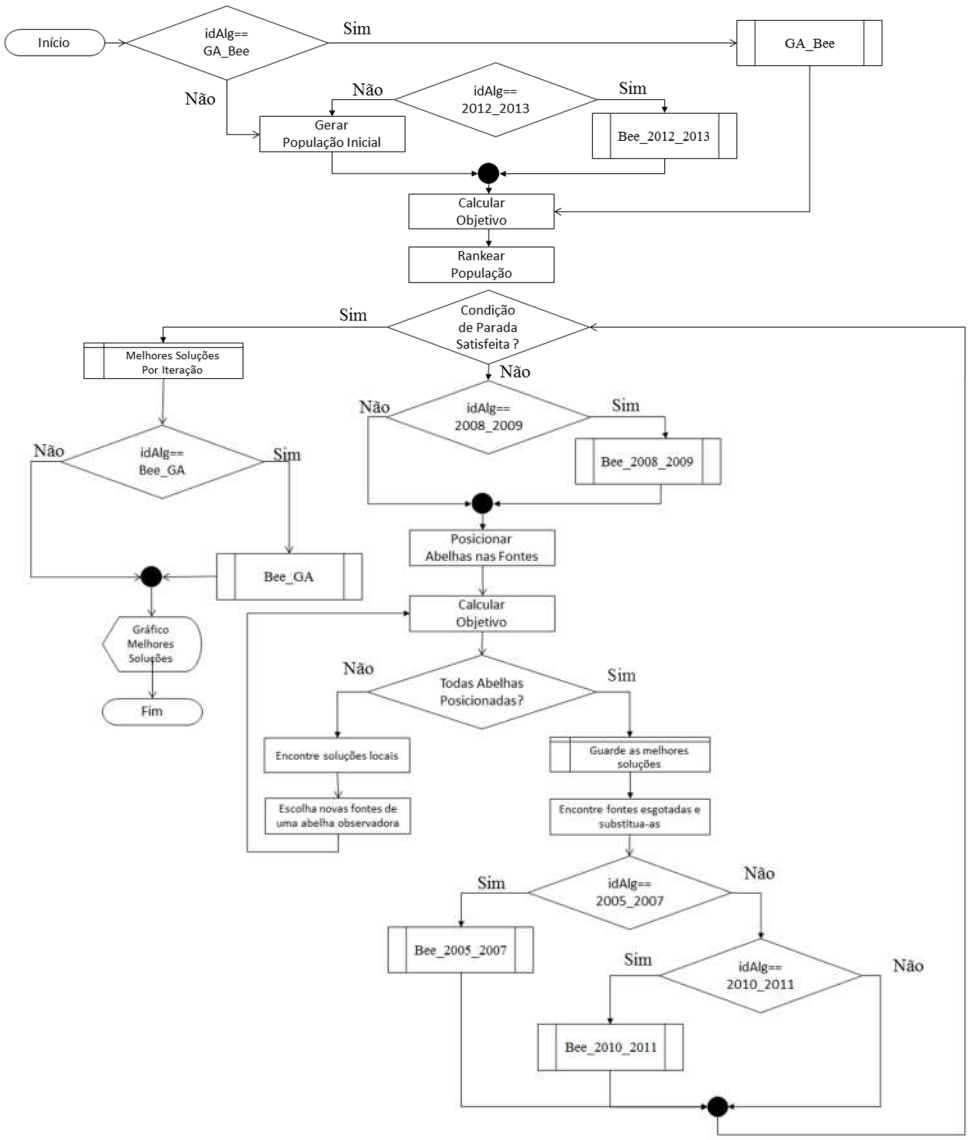
VARIAÇÕES DO ABC QUE FORAM IMPLEMENTADAS

Desde sua origem, algumas variações do ABC vêm sendo propostas para diferentes intuitos e com variadas aplicações numéricas para a realização das buscas. Tomando essa afirmação como base, este trabalho teve como objetivo implementar algumas das variações do ABC, encontradas na literatura, consideradas por nós como mais interessantes, para realização de testes de desempenho, também considerando sua integração com Algoritmos Genéticos de maneira a realizar análises do controle da geração da população inicial e sua efetividade.

As variações do ABC seguem as definições apresentadas, nos extensos trabalhos de revisão realizados por Duarte et al. (2015) e Bolaji et al. (2013), conforme apresentado no fluxograma da Figura 3.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Figura 3- Fluxograma do funcionamento base do ABC de acordo com as diferentes alterações estudadas



Cada uma das variações, compreendidas na Figura 3, estão descritas a seguir, dando ênfase em como elas funcionam matematicamente para que seja possível compreender como se dão as suas variações, assim como, quais são os interesses numéricos quanto aos espaços de busca de cada uma.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Período de 2005 a 2007 (*Bee_2005_2007*)

Essa versão foi inicialmente proposta por Karaboga e Basturk (2008), pois tinham como interesse a aplicação do ABC para a otimização de problemas com restrições. Para tal, uma alteração foi feita na equação relevante à fase de procura de soluções locais, devido ao esgotamento de fontes antigas e para a substituição destas por soluções candidatas melhores. Desta maneira a equação, que define as buscas locais, original é dada por (2) e foi substituída por (3) que considera uma condição inicial para se aplicar o cálculo adequado ao momento.

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} \quad x_{kj}) \tag{2}$$

$$v_{ij} = \begin{Bmatrix} x_{ij} + \Phi_{ij}(x_{ij} \quad x_{kj}), se\ R_j < M\ R_j \in \{1,2,3,\dots,D\} \\ x_{ij}, \qquad caso\ contrário \end{Bmatrix} \tag{3}$$

Vale destacar que nesta versão não são necessárias soluções iniciais para o código, pois estas podem ser geradas com base em um processo extremamente similar às mutações entre iterações produzidas em um algoritmo genético (Karaboga, 2016).

> Na fase da alocação das abelhas do ABCV1, uma busca local é conduzida na vizinhança da solução antigas registradas na memória das abelhas segundo a equação (2) ao invés da equação (1). A Eq. 1 faz modificações em uma das dimensões da atual solução, enquanto que a Eq. 2 faz modificações em várias dimensões, mas somente se um número aleatório uniformemente distribuído R_j é menor que a taxa de perturbação M. (Karaboga, 2016) (tradução livre realizada pelos autores).

Ainda, de acordo com uma comparação em (2), percebe-se que com o parâmetro limitante R_J e um valor M predefinido, com um valor entre zero e um pode-se estabelecer a inicialização de uma população inicial com uma diversidade maior e, desta maneira, a taxa com a qual o algoritmo atua para a otimização dos resultados gerados pelo código pode ser melhorada.

Além desta alteração, outra versão do ABC denominada ABCV2, que é contemporânea ao ABVC1, a equação (2) é substituída por um operador de busca definido por (4). Essa substituição é descrita por Karaboga (2016) como:

Revista brasileira de iniciação científica
ISSN: 2359-232X

> Esta equação utiliza um diferente vizinho (K_j) para uma das dimensões referentes à solução candidata para que assim a informação acerca de mais indivíduos possa ser compartilhada com as demais abelhas de maneira que uma taxa maior de interação possa ser obtida pela população (Karaboga, 2016, tradução livre realizada pelos autores).

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{kj}j \quad x_{ij}), \qquad R_j < MR, e\, x_{kj} \in N_i \qquad (4)$$
$$v_{ij} = v_{ij}\,, \qquad caso\ contrário$$

Período de 2008 a 2009

Algumas mudanças teóricas foram adotadas com base nas aplicações das alterações propostas anteriormente, estas são relevantes ao domínio de certas variáveis tais como Φ_{ij}. Porém, neste período Quan e Shi (2008) trouxeram a proposta de adequar a função de busca das abelhas observadoras, de forma que o número de abelhas associadas à dada fonte de alimento pudesse ser dado por (5).

$$N_i = p_i \times NW \qquad (5)$$

Sendo que p_i representa a probabilidade de seleção de uma fonte associada a cada abelha do sistema e NW o numero total de abelhas observadoras.

Segundo Quan e Shi (2008), esta alteração no algoritmo foi proposta com base em uma teoria conhecida como “espaços de Banach”, que pode ser sintetizada de grosso modo na ideia de se criar um espaço vetorial com uma métrica que permita ser possível o cálculo da distância entre vetores seguindo uma lógica ditada pela sequência de Cauchy.

Período de 2010 a 2011

Neste período, Zhu e Kwong (2010) relataram que, apesar da eficiência que gerar as fontes aleatoriamente trazia para o algoritmo, este ainda se apresentava relativamente pobre no período de busca de soluções locais.

Para corrigir esse problema levantado eles (Zhu; Kwong, 2010) propuseram a adição de uma terceira parcela na equação que define esta fase do algoritmo. Isto viria a caracterizar uma das variantes mais populares do ABC, conhecida como G-Best, e definida por (6).

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} \quad x_{kj}) + \psi_{ij}(y_j \quad x_{ij}) \qquad (6)$$

Revista brasileira de iniciação científica
ISSN: 2359-232X

Essa variação é interessante pelo fato de que um fator de correção é criado com base na melhor solução encontrada na iteração atual do código e na melhor solução global do sistema até o momento.

O fator de correção garante uma maior variabilidade nas soluções existentes e previne que o sistema fique preso na situação problemática conhecida na literatura como '*local optima*' que impede o sistema de otimizar as soluções além de uma dada atual melhor solução local.

> Nestes algoritmos de otimização, "exploração", refere-se à habilidade de investigar as várias regiões desconhecidas no espaço-solução para descobrir um valor otimal global. Enquanto, "aproveitamento" se refere à habilidade de aplicar o conhecimento de antigas boas soluções para encontrar soluções melhores [...] O algoritmo GABC (ou ABC G-best) faz uma comparação entre a nova solução candidata e a solução antiga, mantendo desta forma, a melhor entre as duas. (Zhu e Kwong, 2010) (tradução nossa).

Outras variações a partir deste conceito foram criadas, tais como o ABC *G-Best Guided* (Zhu; Kwong, 2010*)* que realiza um direcionamento na busca do parâmetro que governa o valor da terceira parcela ψ_{ij} adicionada na variação G-Best de (5).

Além desta variação Karaboga (2005) também propôs uma melhoria na busca local do algoritmo tornando a fase de exploração realizada pelas abelhas fechadas em ciclos denominados SPP (*Scout Production Period)* O SPP só ocorre após o término de um número determinado de ciclos preestabelecidos na inicialização do sistema.

Período de 2012 a 2013

Gao, Liu e Huang (2012) tomaram como a versão G-Best de Zhu e Kwong (2010) propuseram uma variação na inicialização da população do ABC utilizando um sistema caótico combinado com o método de aprendizagem baseado em oposição.

Além disto, também indicaram que o uso de uma equação diferente para obter-se a diversidade da população de soluções poderia avançar o algoritmo e a definiram por (7).

$$D = \frac{1}{SN}\sum_{i=1}^{SN}\sqrt{\frac{1}{D}\sum_{j=1}^{D}(x_{ij} \quad \underline{x_j})^2} \qquad (7)$$

Revista brasileira de iniciação científica
ISSN: 2359-232X

Nesse caso SN é o número de soluções na colônia, D o número de variáveis de projeto do sistema e $\underline{x_j}$, o valor médio de j dentro da colônia. Essas alterações foram justificadas por Gao, Liu e Huang (2012) como sendo necessárias para se obter maior diversidade populacional e melhorar o conjunto de soluções obtidas pelo código.

Ao algoritmo proposto por Gao, Liu e Huang (2012) e por Xiang e An (2013), como houve uma maior variabilidade no valor da diversidade das soluções explicitada em (7), eles estabeleceram uma mudança na equação que governa a probabilidade de escolha das fontes de alimento passa a ser representada por (8) para que seja possível agilizar a busca pelas melhores e controlar as variações probabilísticas da solução.

$$p_i = \frac{1}{\frac{fit_i}{\sum_{j=1}^{SN} \frac{1}{fit_j}}} \tag{8}$$

Onde fit_i corresponde à aptidão da solução i encontrada e fit_j a aptidão de uma solução j, enquanto que $\sum_{j=1}^{SN} \frac{1}{fit_j}$ é um somatório a ser realizado para todas as soluções disponíveis no sistema (SN). p_i, conforme explicitado em (5) representa a probabilidade associada à escolha de uma fonte por uma abelha da colmeia virtual.

Segundo Xiang e An (2013), um importante argumento para essa alteração está associado ao o fato de que normalmente a taxa de abandono associada às soluções é fixa no sistema. Porém, com base nesta proposta, seria possível associar diferentes índices de abandono para diferentes fontes de alimento de acordo com a necessidade calculada pelo sistema. Isso promove a possibilidade de uma melhor busca local de soluções candidatas.

As alterações do ABC original que foram apresentadas aqui podem ser tidas como tentativas de melhorar, ou adequar à busca de melhores soluções realizadas pelo ABC de acordo com a necessidade da situação-problema estudada. Neste sentido, um dos objetivos estipulados neste trabalho é justamente implementar variações do ABC, integrando à sua fase de inicialização da população a outro código, para que assim, possamos observar a taxa de variação da otimização entre este e as demais variações discutidas.

A justificativa dessa proposta se dá com base na ideia de que sabendo que a inicialização da população do ABC normalmente é feita com base em números aleatórios limitados pelo intervalo de busca explicitado pelo usuário, se for possível de alguma maneira

organizar essa população inicial com valores já conhecidos, em teoria, o algoritmo teria uma melhor facilidade de otimizar a função.

Ademais, neste trabalho também optou-se pela utilização do AG a ser implementado na inicialização da população do ABC devido á simplicidade do código e ser cabível de integração com o ABC. Assim, duas novas versões são propostas neste trabalho e sendo denominadas *GA_Bee* e *Bee_GA*, representando a inicialização da população do ABC com o GA e a inicialização do GA com o ABC, respectivamente.

IMPLEMENTAÇÃO, RESULTADOS E ANÁLISE DOS DADOS

Os algoritmos esquematizados na Figura 3 foram implementados para a realização de comparações de desempenho, para posterior aplicação em otimizações de telecomunicações, com foco em antenas de microfita.

Todos os algoritmos foram implementados utilizando o ambiente Scilab que é gratuito e disponível em www.scilab.org. Sua linguagem é similar à de outros ambientes matemáticos e linguagens de programação Python, PHP, Perl e outras. Esse ambiente também é interessante por ser simples a forma de instanciação das estruturas de dados matemáticas e visualização desses dados.

Na literatura há diversas funções de testes *benchmark* (Santos, 2010), sendo que, para os fins deste trabalho, optou-se pelas funções Esfera (*Sphere*), Schwefel e Rosenbrock para a realização das comparações dos algoritmos estudados por apresentarem certa variação em seus espaços de busca. A seguir, cada uma delas é apresentada e os respectivos resultados são graficamente expostos para a visualização da convergência dos algoritmos. Além disso, para facilitar a compreensão dos valores numéricos obtidos em cada problema e cada um dos sete algoritmos eles são apresentados em tabelas.

Os dados a serem apresentados foram obtidos pela execução múltipla de cada algoritmo (cinco vezes) para cada problema, de modo a se estabelecer um melhor nível de credibilidade das respostas devido à natureza estocástica que esses algoritmos possuem. Esse procedimento possibilitou verificar estatisticamente o comportamento dos algoritmos ao se calcular a variância dos melhores resultados obtidos em cada execução e apresentado nas tabelas com os valores numéricos obtidos em cada teste.

Revista brasileira de iniciação científica
ISSN: 2359-232X

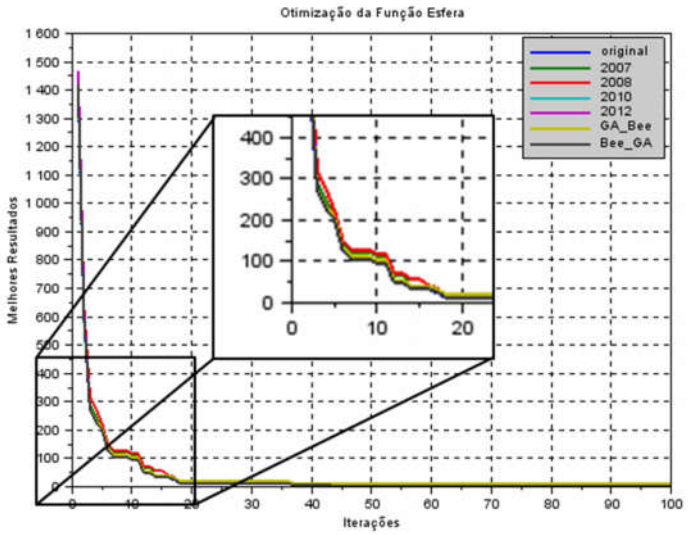
Função Esfera (*Sphere*)

A função *Sphere* é definida pelo quadrado do valor de cada atributo e que aqui é matematicamente representado por (9).

$$f(x) = \sum_{i=1}^{n} x_i^2 , \quad \{-5 \leq x_i \leq 5\} \tag{9}$$

Nas análises realizadas neste trabalho o intervalo aceitável para a geração dos valores aleatórios de x_i foram definidos entre -5 e 5. Os resultados são apresentados no gráfico da Figura 4, onde percebe-se que os algoritmos apresentaram resultados de convergência bastante próximos e que uma análise numérica apresentada na Tabela 1 deixa mais evidente os valores obtidos com cada um deles.

Figura 4 – Gráfico de *benchmark* com a função *Sphere* com os algoritmos implementados.



Como já foi discutido, optamos por alterar o código de maneira que este seja capaz de realizar vários testes consecutivamente, assim, a tabela apresentada dispõe o melhor resultado encontrado dentre os cinco testes, indicando o número representante deste, assim como a média dos melhores valores encontrados pelas diferentes versões, que neste caso por modo de padronização experimental, conforme argumentado anteriormente, são cinco.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Tabela 1- Melhores valores encontrados para a função *Sphere* em cinco testes pelas variações do ABC estudadas.

Variação do ABC	Teste	Melhor Valor Encontrado	Média dos Melhores Valores	Variância
Original	5	1,023262085	1,55291	0,285011
2005_2007	3	0,625276657	2,89133	4,217109
2008_2009	5	0,532972815	2,95452	4,913365
2010_2011	1	1,74	15,262	280,43
2012_2013	1	0,473381042	3,7931	10,042
GA_Bee	2	9	28,6	121,84
Bee_GA	5	2,060577316	3,81444	3,4111

Observando os dados, podemos tirar algumas conclusões baseadas no gráfico gerado e nos valores dispostos na tabela indicada, uma dessas é que no caso desta função o desempenho das versões estudadas foi similar, vale a pena nos atentarmos, porém, ao valor elevado para a variância das versões 2010_2011 e *GA_Bee*. A variância é uma grandeza que pode ser entendida como uma medida do quanto as soluções encontradas variam quando comparadas, logo, poderíamos aferir que maiores valores para a variância indicam uma maior variabilidade nas soluções, que por sua vez indicaria uma curva de otimização mais interessante. Porém, como podemos observar pelo gráfico, os melhores valores encontrados para a função *sphere* nessas duas variações não são os melhores dentre as sete variações estudadas. Com base em somente uma função de *Benchmark* seria difícil de julgar a eficiência destas variações, logo, ao longo dos demais testes, poderemos observar os comportamentos dessas variações de modo a tirar mais conclusões.

Função Rosenbrock

Nesta seção discutiremos os resultados obtidos para a otimização da função de Rosenbrock pelas variações do algoritmo estudadas. Esta função pode ser definida matematicamente como (10).

$$f(x) = \sum_{i=1}^{n-1} [100 \quad (x_{i+1} \quad x_i^2) + (x_i \quad 1)^2], \{ \quad \infty \leq x_i \leq \infty \; e \; 1 \leq i \leq n \} \qquad (10)$$

Revista brasileira de iniciação científica
ISSN: 2359-232X

Similarmente à função anterior, seguem a curva de otimização obtida ao longo de cinco testes para a função Rosenbrock (Figura 5), assim como na Tabela 2 que apresenta os valores associados às melhores soluções encontradas nas setes variações estudadas.

Figura 5 – Gráfico dos melhores valores encontrados para a função Rosenbrock em cinco testes pelas versões do ABC estudadas.

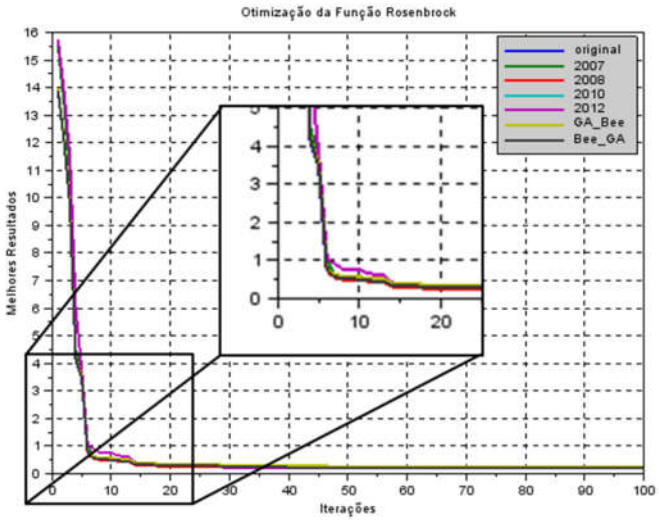


Tabela 2- Melhores valores encontrados para a função Rosenbrock em cinco testes pelas variações do ABC estudadas

Variação do ABC	Teste	Melhor Valor Encontrado	Média dos Melhores Valores	Variância
Original	5	0,007482809	0,18778	0,0127
2005_2007	3	0,051232341	0,2557	0,03194
2008_2009	5	0,0002934	0,14214	0,01671
2010_2011	4	0,15	0,29	0,019339
2012_2013	5	0,021871234	0,1952	0,027035
GA_Bee	1	0,13251014	0,61262	0,110452
Bee_GA	5	0,112409001	0,210293	0,0053735

Associando os dados gráficos da Figura 5 e numéricos da Tabela 2 encontrados por cada teste, em cada variação do código estudada, podemos perceber que nesta função, as variações que obtiveram um melhor desempenho foram a original, a 2008_2009 e a *Bee_GA*. Chegamos a essa conclusão tanto pela análise do comportamento da curva de otimização e na sobreposição das mesmas no gráfico obtido, assim como no valor encontrado para a variância

Revista brasileira de iniciação científica
ISSN: 2359-232X

dos resultados, sendo que dentre as três versões destacadas, a versão *Bee_GA* obteve a menor variância entre todas as variações estudadas.

Função Schwefel

A função de schwefel pode ser definida matematicamente como:

$$f(x) = 418{,}98229 \quad D \quad \sum_{i=1}^{D} x_i sin(|\sqrt{x_i}|) \qquad x_i \epsilon \ [-500.500] \tag{11}$$

Vale a pena relembrar que, conforme já comentado, normalmente, uma otimização trabalha com base na teoria de sempre buscar um valor menor que atenda às condições de solução para uma dada função estudada, porém no caso da função de Schwefel, valores negativos podem ser aceitos como melhores soluções, devido ao caráter da superfície descrita pela função em (11).

Com base nos resultados apresentados na Figura 6 e Tabela 3, pode-se observar as variações que apresentaram um melhor desempenho na otimização desta função foram respectivamente em ordem de qualidade de original, a *Bee_GA,* e a *GA_Bee*. Esta afirmação se justifica pela mesma lógica utilizada nas outras duas funções anteriores, ou seja, pela análise do comportamento das curvas de otimização do valor do melhor resultado encontrado ao longo de cinco testes.

Considerando-se as três funções propostas para o benchmark do algoritmo, poderíamos salientar que as versões do algoritmo que obtiveram um melhor desempenho ao longo de nossa análise foram a versão original, a *Bee_GA,* e a *GA_Bee*, ou seja, as duas propostas levantadas neste trabalho demonstraram um rendimento excelente em comparação com as demais. Vale a pena, contudo, nos atentarmos ao alto valor para a variância encontrado em algumas das variações estudadas. Este fato pode ser explicado justamente pela natureza de uma otimização, uma menor variância representaria uma uniformidade entre os dados encontrados, logo nos casos onde os valores iniciais foram ideais e/ou a convergência ao melhor valor encontrado foi rápida a variância encontrada acabou exprimindo um número menor.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Figura 6 – Gráfico dos melhores valores encontrado para a função Schwefel em cinco testes, pelas versões do ABC estudadas.

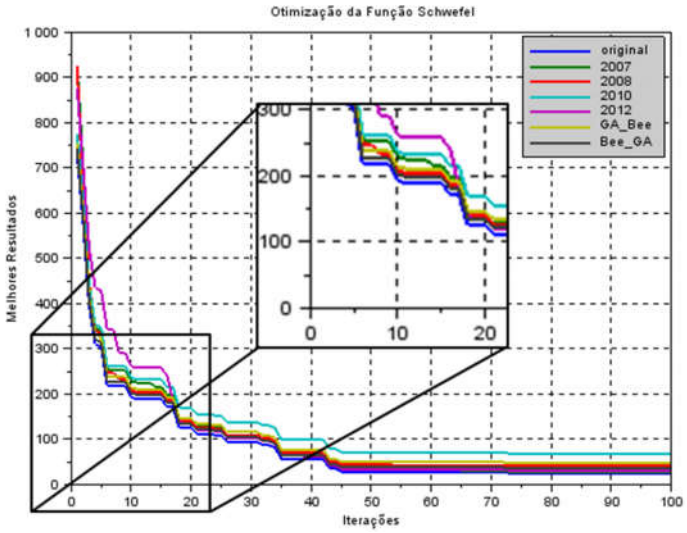


Tabela 3- Melhores valores encontrados para a função Schwefel em cinco testes pelas versões do ABC estudadas.

Variação do ABC	Teste	Melhor Valor Encontrado	Média dos Melhores Valores	Variância
Original	4	6,910860868	25,6630964	98,74668
2005_2007	3	2,96793	37,06175563	1544,409
2008_2009	1	17,47804	50,50201992	400,6287
2010_2011	3	25,52	163,03	6264,766
2012_2013	3	11,84310788	35,98	1219,88
GA_Bee	3	0,000271968	134,71	8731,36
Bee_GA	2	19,5295493	61,93	903,27

O algoritmo ABC, num geral, possui diversas fases de reavaliação das soluções encontradas pelo código, isso tudo acarreta em um custo computacional elevado em comparação com outros algoritmos cujas definições são diferentes. Nas propostas levantadas nesse trabalho, conseguimos de certa maneira burlar esse custo computacional, principalmente na versão *Bee_GA*.

Por meio da análise geral do desempenho dos algoritmos propostos, pode-se argumentar no sentido de que utilizar mecanismos para controlar a inicialização da população de um

Revista brasileira de iniciação científica
ISSN: 2359-232X

algoritmo por outro deve-se ter cuidado, pois se o primeiro algoritmo atinge a otimização do problema proposto, o segundo não conseguirá otimizar os resultados encontrados além de um certo limite.

Na proposta *Bee_GA*, foi possibilitada a inicialização da população pelo ABC com um ciclo de apenas 10 iterações, com esses dados preliminares, o GA então, realizou o ciclo padrão de 100 iterações para a otimização das funções estudadas. Conforme vimos nas funções estudadas. As propostas levantadas nesse projeto se mantiveram entre as que obtiveram os melhores desempenhos nos testes. Especificamente entre as duas, de acordo com os testes apresentados, considera-se que a versão *Bee_GA* é mais interessante que a versão original do ABC, pois seu custo computacional é muito menor.

CONCLUSÕES

Com os resultados apresentados neste trabalho verifica-se que suas aplicações nos testes apresentados obtiveram resultados interessantes, de forma que a alteração para o ABC proposta por nós se equiparou às demais já amplamente estudadas na literatura, este fato traz poder ao argumento de que talvez o *GA_Bee* seja uma estratégia interessante que pode ser tomada para a melhora da otimização de algoritmos sem que para isso tenhamos que reescrever as fases de busca intrínsecas do código.

Assim, com as variações de algoritmos e as proposições aqui apresentadas integrando GA e Bee conclui-se que eles apresentam um bom desempenho médio. O controle populacional inicial é um mecanismo interessante para acelerar a convergência às soluções esperadas, mas vale a pena ressaltar que deve-se ter cuidado com o excesso de execução do primeiro algoritmo e como essa integração dos algoritmos ocorrerá. Também sugere-se que variados casos de testes sejam executados para se garantir variabilidade de desempenho dos algoritmos.

Como trabalhos em andamento por essa equipe, constam estudos de diversidade nos valores de certas variáveis relevantes aos algoritmos, tais como taxa de mutação probabilidade de recombinação, diferentes valores para o número de iterações e tamanho das populações dos algoritmos para se verificar os desempenhos sob essas variadas condições.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Referências

BOLAJI, ASAJU LA'ARO et al. Artificial bee colony algorithm, its variants and applications: A survey. **Journal of Theoretical & Applied Information Technology**, v. 47, n. 2, 2013.

BONABEAU, Eric et al. Swarm intelligence: from natural to artificial systems. **Oxford university press**, 1999.

BRIANEZE, Juliano Rodrigues; SILVA-SANTOS, C. H.; HERNÁNDEZ-FIGUEROA, Hugo Enrique. Multiobjective evolutionary algorithms applied to microstrip antennas design algoritmos evolutivos multiobjetivo aplicados a los proyectos de antenas microstrip. **Ingeniare. Revista chilena de ingeniería,** v. 17, n. 3, p. 288-298, 2009.

DA SILVA, L. G.; CERQUEIRA, S. Arismar; DA SILVA-SANTOS, C. H. Tri-band RF filters optimization using evolutionary strategy. In: **Microwave & Optoelectronics Conference (IMOC), 2013 SBMO/IEEE MTT-S International. IEEE**, 2013. p. 1-5.

DA SILVA FERREIRA, Adriano et al. Towards an integrated evolutionary strategy and artificial neural network computational tool for designing photonic coupler devices. **Applied Soft Computing**, v. 65, p. 1-11, 2018.

DUARTE, Grasiele Regina et al. Um algoritmo inspirado em colônias de abelhas para otimização numérica com restrições. 2015.

GAO, Weifeng; LIU, Sanyang; HUANG, Lingling. A global best artificial bee colony algorithm for global optimization. **Journal of Computational and Applied Mathematics**, v. 236, n. 11, p. 2741-2753, 2012.

HANCER, Emrah; KARABOGA, Dervis. A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number. **Swarm and Evolutionary Computation**, v. 32, p. 49-67, 2017.

KARABOGA, Dervis et al. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. **Artificial Intelligence Review**, v. 42, n. 1, p. 21-57, 2014.

KARABOGA, Dervis. **An idea based on honey bee swarm for numerical optimization**. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.

KARABOGA, Dervis; AKAY, Bahriye. A modified artificial bee colony (ABC) algorithm for constrained optimization problems. **Applied soft computing**, v. 11, n. 3, p. 3021-3031, 2011.

KARABOGA, Dervis; BASTURK, Bahriye. On the performance of artificial bee colony (ABC) algorithm. **Applied soft computing**, v. 8, n. 1, p. 687-697, 2008.

KUMAR, Dharmender; KUMAR, Balwant. Optimization of benchmark functions using artificial bee colony (ABC) algorithm. **Optimization**, v. 3, n. 10, p. 9-14, 2013.

QUAN, Haiyan; SHI, Xinling. On the analysis of performance of the improved artificial-bee-colony algorithm. In: **Natural Computation, 2008. ICNC'08. Fourth International Conference on**. IEEE, 2008. p. 654-658.

SANTOS, Carlos Henrique da Silva et al. Computação paralela aplicada a problemas eletromagnéticos utilizando o método FDTD. 2005.

SANTOS, Carlos Henrique da Silva et al. Computação bio-inspirada e paralela para a análise de estruturas metamateriais em microondas e fotonica. 2010.

SILVA-SANTOS, Carlos H.; RODRÍGUEZ-ESQUERRE, Vitaly F.; HERNÁNDEZ-FIGUEROA, Hugo E. Artificial Immune Network Design of Optical Multiplexers/Demultiplexers. **Journal of Microwaves, Optoelectronics and Electromagnetic Applications**, v. 14, n. 2, p. 229-237, 2015.

SILVA-SANTOS, Carlos Henrique da; HERNÁNDEZ-FIGUEROA, Hugo Enrique. Benchmarking Parallel Natural Algorithms for Telecommunications Devices Design. **International Journal of Advanced Research in Computer Science**, v. 4, n. 3, 2013.

SILVA-SANTOS, C. H.; RODRÍGUEZ-ESQUERRE, V. F.; HERNÁNDEZ-FIGUEROA, H. E. An artificial immune system for optical fiber based directional couplers multiplexer/demultiplexers design. In: **Latin America Optics and Photonics Conference. Optical Society of America**, 2010. p. PDPTuJ5.

SILVA-SANTOS, Carlos H. et al. An artificial immune system algorithm applied to the solution of an inverse problem in unsteady inward solidification. **Advances in Engineering Software**, v. 121, p. 178-187, 2018.

SILVA, L. G.; S JR, Arismar Cerqueira; DA SILVA SANTOS, Carlos H. Development of tri-band RF filters using evolutionary strategy. **AEU-International Journal of Electronics and Communications**, v. 68, n. 12, p. 1156-1164, 2014.

XIANG, Wan-Li; AN, Mei-Qing. An efficient and robust artificial bee colony algorithm for numerical optimization. **Computers & Operations Research**, v. 40, n. 5, p. 1256-1265, 2013.

ZHU, Guopu; KWONG, Sam. Gbest-guided artificial bee colony algorithm for numerical function optimization. **Applied mathematics and computation**, v. 217, n. 7, p. 3166-3173, 2010.