

# Colônia Artificial de Abelhas na Otimização de Problemas com Restrições

Danrley Alves dos Santos  
Programa de Pós-Graduação em  
Engenharia da Computação e  
Sistemas (PECS)  
Universidade Estadual do Maranhão  
(UEMA)  
Teresina, Piauí, Brazil

Elton de Sousa e Silva  
Programa de Pós-Graduação em  
Engenharia da Computação e  
Sistemas (PECS)  
Universidade Estadual do Maranhão  
(UEMA)  
São Luís, Maranhão, Brazil

Omar Andres Carmona Cortes  
Departamento de Computação  
(DComp)  
Instituto Federal do Maranhão (IFMA)  
São Luís, Maranhão, Brazil  
omar@imfa.edu.br

## ABSTRACT

Many insects such as bees, ants, and termites have a low level of intelligence; however, when they work together, they are able to execute complex tasks in an organized way. With great inspiration in these insects' collective intelligence aspect, this work aims to present results and comparisons using ANOVA and Tukey test using three optimization metaheuristics: Artificial Bee Colony, Particle Swarm Optimization, and Differential Evolution. All the algorithms were implemented in Python and later compared using three constrained benchmark functions called G06, G0, and G11. Results have shown that ABC tends to present better results.

## KEYWORDS

Inteligência de Enxames, Algoritmo Colônia de Abelhas Artificiais, Otimização, Métodos de Amostragem, Meta-Heurísticas.

## 1 INTRODUÇÃO

A busca por resoluções de problemas intratáveis (NP-completos) possuem métodos e modelos heurísticos com base na natureza e conceitos sociais [16]. Utilizando-se de informações anteriores com o objetivo de chegar a uma boa solução, muitos algoritmos foram desenvolvidos porque as resoluções de problemas de ordem exponencial por métodos determinísticos levariam um tempo absurdo para serem solucionados [15] quando passam dos limites aceitável para obter resultados.

Para solucionar problemas de otimização numérica como alternativas as meta-heurísticas tem se mostrado bastante eficientes na solução de muitos problemas atingindo diversas áreas como, por exemplo, na engenharia [14], ciências sociais [6], medicina [19] e outras. Devido ao desempenho e a facilidade de implementação em busca de melhores soluções o número de aplicações justificam-se por conta dos resultados de melhores resoluções e pelo fato de não serem fundamentadas limitadas por espaço de busca [7].

Com objetivos de avaliar a eficiência dos algoritmos, as funções de benchmark são funções onde seus valores máximos e mínimo são conhecidos [8]. As mesmas possibilitam que os algoritmos sejam colocados em teste e avaliados posteriormente. Sendo assim, neste trabalho serão utilizadas as seguintes meta-heurísticas: Otimização

por Enxame de Partículas (PSO) [12], Evolução Diferencial (ED) [18] e Colônia Artificial de Abelhas (ABC) [3]. Os mesmos serão utilizados na resolução de três funções Benchmark apresentados na tabela 1 e comparando os melhores resultados obtidos entre eles.

Todas as variáveis envolvidas na definição da função objetivo deverão representar os recursos utilizados no projeto, sempre observando o que os mesmos serão limitados. Constantemente a solução do problema de otimização será passível a restrições, isto significa que alguns requisitos deverão ser atendidos para que a mesma seja considerada viável.

O algoritmo Colônia de Abelhas Artificiais (ABC - do inglês *Artificial Bee Colony*), é um algoritmo bio-inspirado proposto em 2005 para a resolução de problemas de otimização multimodais e multidimensionais [11]. Foi inspirado no comportamento social observado em colônias de abelhas reais, em particular no comportamento social observado em colônia de abelhas reais, em particular durante o forrageamento<sup>1</sup>. Quando descobrem novas fontes de alimento ou mesmo já explorando uma, ao retornarem para o ninho cada abelha divide com as outras, informações sobre a fonte conhecida. Existe alguma tendência de que as fontes de melhor qualidade sejam exploradas por um número maior de abelhas. Ao avistarem as fontes conhecidas, caso na vizinhança seja encontrada alguma de qualidade superior, as abelhas poderão memorizar a localização dessa nova fonte e dividir a informação ao retornarem para o ninho. Já no forrageamento de colônias de abelhas, três tipos de abelhas estão presentes: as abelhas campeiras que exploram as fontes conhecidas e dividem informações ao retornarem para o ninho, as abelhas observadoras que aguardam no ninho e escolhem uma fonte para explorarem de acordo com as informações recebidas e as abelhas exploradoras que buscam novas fontes para substituírem aquelas esgotadas.

Dentro da busca pela solução do problema, o algoritmo ABC simula a atividade de forrageamento de uma colônia de abelhas. Para tanto, de forma artificial são presentes as fontes de alimento e os três tipos de abelhas envolvidas. As abelhas observadoras escolhem as soluções conhecidas pelas abelhas campeiras de acordo com a sua qualidade para resolver o problema. As abelhas campeiras e observadoras realizam ainda uma busca local em torno de fonte conhecida e sendo encontrada uma solução melhor, a fonte memorizada será substituída. As fontes conhecidas por abelhas campeiras que não melhoram durante o processo de busca são substituídas

Authors' addresses: Danrley Alves dos Santos, Programa de Pós-Graduação em Engenharia da Computação e Sistemas (PECS), Universidade Estadual do Maranhão (UEMA), Teresina, Piauí, Brazil; Elton de Sousa e Silva, Programa de Pós-Graduação em Engenharia da Computação e Sistemas (PECS), Q.Z.C.03 - São Sebastião, Universidade Estadual do Maranhão (UEMA), São Luís, Maranhão, Brazil; Omar Andres Carmona Cortes, Departamento de Computação (DComp), Instituto Federal do Maranhão (IFMA), Rono-Hills, São Luís, Maranhão, Brazil, omar@imfa.edu.br.

<sup>1</sup>Forrageamento é a busca e a exploração de recursos alimentares. É uma habilidade particularmente importante pois afeta a aptidão do animal, influenciando diretamente a sobrevivência e a reprodução do organismo [4]

por outras geradas aleatoriamente, simulando assim a atividade das abelhas exploradoras. O algoritmo ABC é de fácil implementação para sua execução representado na figura 1 onde requer a definição de poucos parâmetros.

Neste contexto, este trabalho tem como objetivo realizar uma análise e estudo comparativo do algoritmo ABC em relação aos algoritmos de Otimização por Enxame de Partículas (PSO) e Evolução Diferencial (DE), aplicando funções de otimização com restrição de Benchmarks.

Este artigo está dividido em 6 seções da seguinte forma: A seção 2 abordar os tópicos relacionados aos principais conceitos de otimização e descrição das funções utilizadas para a implementação do algoritmo ABC bem como as funções de Benchmark. Em seguida, na seção 3 descreve o comportamento das abelhas na natureza e a inspiração que levou o desenvolvimento da meta-heurística ABC. Na seção 4 foram apresentados as configurações da máquina e parâmetros utilizados para implementação dos algoritmos e testes realizados. A seção 5 apresenta os experimentos e resultados comparativos realizados entre os Algoritmo ABC, PSO e DE. E por fim a seção 6 apresenta as conclusões deste trabalho e direcionamentos futuros.

## 2 OTIMIZAÇÃO

Otimização é uma área da matemática que tem como objetivo encontrar a solução (ponto ou pontos) no domínio de uma função (ou várias funções) que produza os valores máximos ou mínimo que esta assume. Desta forma, otimizar consiste em buscar pelo valor máximo ou mínimo de uma função. Uma função em seu ponto que maximiza ou minimiza é chamado de solução ótima.

Para implementarmos o algoritmo ABC existem duas importantes funções que sustentam o algoritmo:

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_i} \quad E.q.(2.1) \quad (1)$$

$$V_{ij} = X_{ij} + \phi_{ij}(X_{ij} - X_{kj}) \quad E.q.(2.2) \quad (2)$$

A solução ótima geralmente estará sujeita a determinadas condições que deverão ser atendidas para que seja considerada realmente viável para resolver problemas.

A variável  $P_i$  é o valor da probabilidade associado com  $i$  fontes de alimentos calculados pela equação 1, posteriormente quando uma abelha espectadora for escolher uma fonte de alimento com base no valor  $P_i$ . Esta equação  $fit_i$  representa a qualidade ou tamanho da fonte de alimento que é medida pelas abelhas trabalhadoras empregadas e  $SN$  é o número de fontes de alimentos que é igual ao número de abelhas empregadas.

Nesta heurística as abelhas artificiais necessitam de fazer buscas para encontrar novas possíveis soluções. A equação 2 representa a estratégia de busca original utilizada pelo algoritmo ABC como otimização.  $V_{ij}$  é o novo candidato a posição de fonte de alimento gerado por esta equação. Onde  $\kappa \in [1, 2, \dots, SN]$  e  $j \in [1, 2, \dots, D]$  são parâmetros escolhidos aleatoriamente, onde  $\kappa$  deve ser diferente de  $j$ . Assim  $X_{ij}$  e  $X_{kj}$  representam as diferentes posições de fontes antigas de alimentos.  $SN$  representa o número de abelhas empregadas e  $D$  é o número de parâmetros de otimização.  $\phi_{ij}$  é o número aleatório entre  $[-1, 1]$  e controla a distância de uma posição que estar na fonte de alimento vizinha em todo  $X_{ij}$ .

Um ponto muito importante na Heurística ABC é o processo de seleção gulosa que determina se uma nova fonte encontrada é melhor do que uma fonte anterior. Caso a fonte nova de alimento seja considerada melhor que a anterior, a mesma será substituída da memória pela nova, caso contrário a informação a fonte anterior é mantida.

Existe vários métodos usado para resolução de problemas de otimização. Toda essa diversidade se deve à distinção entre os problemas, tornando impossível um único método possa ser aplicado de forma generalizada. O Algoritmo 1 apresenta o pseudocódigo do *Colônia Artificial de Abelhas* para problema de otimização.

---

### Algorithm 1 Pseudocódigo do Algoritmo ABC

---

```

Carrega as amostras de treinamento.
Gera a população inicial  $i = 1 \dots SN$ .
Calcula o fitness ( $fit_i$ ) da população.
Seta o ciclo para 1
para cada abelha empregada faça
    Produzir nova solução de  $V_i$  utilizando Eq.2.2
    Calcular valor ( $fit_i$ )
    Aplicar processo de seleção gulosa }
fim para
Calcular o valor de  $P_i$  para soluções ( $X_i$ ) utilizando Eq.2.1
para Para cada abelha espectadora faça
    Selecione uma solução ( $X_i$ ) com base em  $P_i$ 
    Produz nova solução  $V_i$ 
    Calcula o valor de ( $fit_i$ )
    Aplicar processo de seleção gulosa
fim para
se Existir alguma solução não verificada então
    então substitua pela nova função que irá ser aleatoriamente produzida  $\phi_{ij}$ .
    Memorize a melhor solução gerada até agora
    Memorize Ciclo = ciclo + 1
    Até ciclo == número máximo de ciclos.
fim se

```

---

### 2.1 Algoritmos de otimização

Boa parte dos problemas de otimização atuais são ou tornam muita das vezes complexos. Isso dificulta tornando impossível que sejam resolvidos de forma rápida. Para isso o uso de recursos computacionais tornam-se indispensável para a resolução de tais problemas. Os algoritmos de otimização basicamente são classificados de acordo com a característica de serem ou não determinísticos.

A partir da observação de fenômenos da natureza foram propostos algoritmos que os imitam. Os mesmos podem ter inspiração em processos evolutivos, onde se parte de uma condição inicial e evolui-se para uma outra, mas podem também ter inspiração no comportamento inteligente de algumas espécies sociais, como é o caso de alguns insetos. Já o algoritmo Colônia de Abelhas Artificiais teve inspiração natural o comportamento social observado no forrageamento de colônias de abelhas reais. Por se tratar de objeto de estudo nesta atividade iremos abordá-lo com mais profundidade logo em seguida.

**Table 1: Funções de Benchmark**

Function G06	Function G08	Function G11
$f(x) = (x_1 - 10)^3 + (x_2 - 20)^3$ $g_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$ $g_2(x) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0$	$f(x) = \frac{\sin^3(2\pi x_1) \sin(2\pi x_2)}{z_1^3(x_1 + x_2)}$ $g_1(x) = x_1^2 - x_2 + 1 \leq 0$ $g_2(x) = 1 - x_1 + (x_2 - 4)^2 \leq 0$	$f(x) = x_1^2 + (x_2 - 1)^2$ $h(x) = x_2 - x_1^2 = 0$

## 2.2 Funções de Benchmark

As funções de benchmark são funções genéricas onde todas as características delas são previamente conhecidas, onde é possível usar essas funções para testar funcionamento e otimização de nossos algoritmos.

Para avaliação do potencial de busca e otimização do algoritmo ABC foram utilizados três diferentes funções Benchmark conforme apresentado na tabela 1, em 3 dimensões do tipo  $z=f(x,y)$ .

As três funções escolhidas foram funções simples com um ponto mínimo e um ponto máximo onde apenas há uma solução ótima. A primeira função de benchmark utilizada foi a *Function G06*, onde possui seu  $f(x) = -6961.81387558015$ .

A segunda função que foi utilizada neste trabalho para os teste e procedimentos de avaliação foi a *function G08* onde possui seu  $f(x) = -0.0958250414180359$ . E a última função que foi utilizada para testes de avaliação foi a *function G11*, onde possui as restrições de  $-1 < X_1 < 1$  e com o seu valor como ótima solução de  $f(x) = 0.7499$ .

Como forma de satisfazer as restrições, não foram gerados indivíduos que estivessem em conforme as mesmas. Sendo assim ao criar um novo indivíduo caso este não passasse nas restrições, os mesmo não seriam utilizados no algoritmo.

## 3 ALGORITMOS DE ABELHAS

Os Algoritmos de abelhas são algoritmos que buscam sua inspiração no comportamento natural das abelhas [3]. Esses algoritmos estão inseridos na área de Inteligência Artificial conhecida especificamente como Inteligência de Enxames ou Insetos Sociais. Existem outros algoritmos baseados no comportamento das formigas como por exemplo descrito em [2].

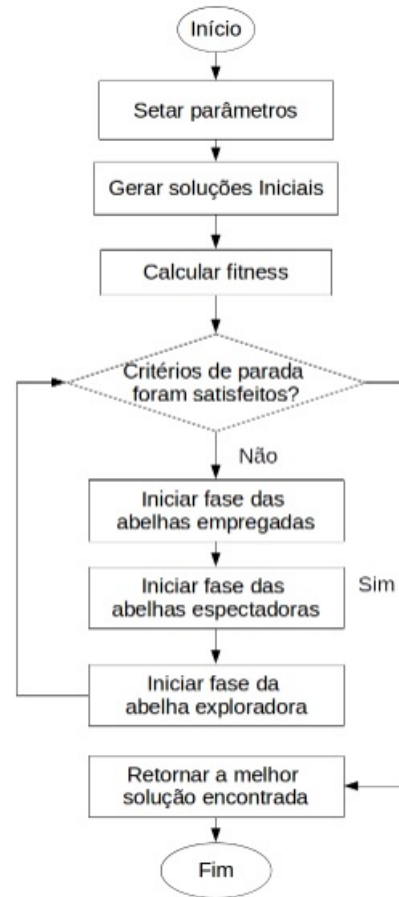
Alguns insetos como: Abelhas, formigas e cupins, são insetos que possuem individualmente características fortemente baixa no aspecto de inteligência e ações não inspecionadas, porém de maneira coletiva possuem grande senso de inteligência coletiva dentro da colônia, onde são capazes de resolver problemas complexos. Por exemplo, no caso das formigas são capazes de encontrar o caminho mais curto para uma fonte de alimento entre muitas possibilidades de caminhos diferentes [13].

As atividades em uma colônia não precisam ser controladas por um agente central ou inspecionadas, de maneira independentes todas elas são realizadas. Em uma colônia cada membro individualmente contribui constantemente deixando informações úteis para as demais abelhas da colônia. No caso das formigas onde deixam feromônios ou as abelhas que realizam danças [5].

A metaheurística Artificial Bee colony (ABC) implementada neste artigo, possui o objetivo no processo de localizar e explorar as melhores posições dentro de um problema de busca [2] é um

algoritmo baseado na inteligência de enxames que teve sua utilização idealizada por [11]. Contudo este algoritmo é muito utilizado em diversas aplicações de otimização como estrutura de proteínas terciárias [1], filtros IRR digital [10], redes neurais artificiais [9] entre outras áreas de aplicação [17].

O fenômeno natural que serviu de inspiração para o desenvolvimento do ABC foi o comportamento social observado em colônias de abelhas reais, em particular a realização das atividades envolvidas no forrageamento..

**Figure 1: Algoritmo ABC - adaptada**

### 3.1 Enxame de abelhas - Forrageamento

Em um enxame de abelhas podem ser observados o seu comportamento coletivo inteligente, com isto, uma colônia de abelhas pode

ser vista como um sistema dinâmico que se ajusta de acordo com suas necessidades, ainda que seus agentes (abelhas), se observados de forma individual possuam capacidade e conhecimento limitados. O funcionamento do sistema não será prejudicado se uma abelha não apresentar o desempenho esperado.

Quando se fala do forrageamento numa colônia de abelhas, basicamente as atividades realizadas estão relacionadas à busca por novas fontes de alimento, exploração das fontes conhecidas, recrutamento de abelhas para intensificação da exploração de boas fontes e abandono de fontes com recursos esgotados. Sendo que para que estas atividades sejam realizadas, três elementos básicos estão presentes no forrageamento das colônias de abelhas.

Geralmente o termo bastante utilizado enxame aplica-se a qualquer forma de coleção de agentes (ou indivíduos) interagindo coletivamente. Os enxames apresentam comportamento coletivo inteligente, onde uma colônia de abelhas consiste de um exemplo clássico de enxame inteligente. Dentre diversas coleções de agentes como formigas, células, vespa, entre outros.

Um enxame para que seja classificado como inteligente, deverão estar presente algumas características, sendo divisão do trabalho e auto-organização:

- I. Divisão do trabalhos: Divisão do trabalho passa a existir quando num enxame diversas tarefas são desenvolvidas simultaneamente por indivíduos especializados. Acredita-se que desta forma há uma maior eficiência na realização das tarefas do que quando executadas sequencialmente por muitos indivíduos não especializados.
- II. Auto-organização: Consiste da capacidade de se estabelecer uma estrutura global de um sistema por meio de interações entre os agentes de níveis específicos, onde um sistema pode ser visto como um conjunto de elementos e as relações entre eles. Para que haja auto-organização, são estabelecidos regras básicas para a interação entre os agentes do sistema.

## 4 EXPERIMENTOS

### 4.1 Configurações

Os testes foram todos realizados no sistema operacional Windows 10, 64 bits, 8 Gb de RAM, 120 GB de SSD, linguagem Python versão 3.8 na plataforma online Colab. Os valores atribuídos para o algoritmo ABC foram: tamanho da colônia  $T = 10$ , dimensionalidade do problema em questão ou seja, quantidade de variáveis de decisão  $D = 2$ , número de ciclos de forrageamento realizados pelas abelhas  $N = 1000$  e o número de execuções realizadas  $E = 30$ . Para o algoritmo DE:  $crossOverRate = 5.0$  e  $scalingVector = 8.0$ . E por fim o Algoritmo PSO: Cognitivo individual  $C_1 = 1.5$ , social  $C_2 = 1.5$  e peso  $w = 0.7$ .

### 4.2 Resultados

De acordo com a Tabela 2 onde demonstra os melhores resultados obtidos nas 30 execuções dos algoritmos ABC, PSO e DE juntamente com os seus respectivos desvios padrão (D.P), na qual observou-se que nas três funções de benchmarks escolhidas apresentou-se ótimos resultados nos teste realizados do algoritmo ABC em comparação ao algoritmo PSO e DE. Em todos os cenários o algoritmo

**Table 2: Melhores resultados e desvio padrão**

	Function G06	Function G08	Function G11
ABC	<b>-6960.33788942</b>	<b>-0,09582504</b>	<b>0,74999999</b>
D.P.	167,9108678	0,012174269	0,001923427
PSO	-6944,481013	<b>-0,09582504</b>	0,75
D.P.	150,3270877	1,62594E-08	0,00444236
DE	-6897,950477	-0,09582401	0,75
D.P.	165,9205605	0,01217009	0,009527374
F.	6961.81387558	-0.095825041418	0.7499

ABC apresentou melhor desempenho em relação os outros algoritmos. Na otimização da função G06 o algoritmo ABC atingiu uma solução de -6960.33788942, demonstrando um desempenho muito próximo da solução ótima. Houve um empate entre os algoritmos ABC e PSO na função G08 atingindo um cenário próximo da solução ótima estabelecida, e por fim na função G11 o algoritmo ABC demonstrou superioridade em seu resultado com solução de 0.74999999 de caráter ótimo em relação ao PSO e DE.

Também foi realizado uma comparação dos resultados obtidos utilizando o método estocástico ANOVA que é conhecido também como Análise de Variância, onde estabelece as diferenças dos resultados de três ou mais grupos. A tabela 3 mostra os resultados obtidos utilizando o teste de ANOVA aplicado para todos os algoritmos e funções G06, G08 e G11. A comparação entre os algoritmos na função G06 utilizando ANOVA, com o valor F de 4,15636546, sendo este maior que o valor F crítico de 3,101295756 observando o valor P igual 0,018880903, chegou-se a conclusão de que não existe diferença significativa considerando a hipótese nula rejeitada em relação aos algoritmos ABC, PSO e DE.

Na função G08, os valores de F igual 0,50587291, sendo menor que o valor de F crítico de 3,10129575 e valor P de 0,60474157 sendo maior que 0,05, determina-se que também não existe diferença significativa porém devido o valor de F crítico ser superior ao valor de F, considera-se então que a hipótese nula não é rejeitada entre os algoritmos usados. Para a função G11, o valor P de 0,07759664 sendo maior que 0,05, significando que não existe diferença significativa também. Sendo o Valor F de 2,63283124 é menor que o de F crítico de 3,10129575, onde a hipótese nula também não é rejeitada. De modo geral, Para as três funções de benchmark os valores de F Crítico apresentado na tabela 3 para as funções utilizadas, verificamos que não existe diferença significativa em relação as soluções das funções do algoritmo ABC em relação ao algoritmo PSO e DE.

Após a realização dos testes em todos os algoritmos utilizando teste ANOVA, foi realizado uma análise através do teste de Tukey apresentado na Tabela 4. Observa-se que para a função G06 o algoritmo ABC em relação ao algoritmo PSO e DE, apresentam uma diferença significativa.

## 5 CONCLUSÃO

Este artigo apresentou uma análise sobre a comparação de otimização de três meta-heurísticas, utilizando-se das funções de benchmark para análise de otimização numérica. De maneira geral, foram realizados experimentos computacionais aplicando as funções de benchmark G06, G08 e G11 e logo após comparando os valores do teste ANOVA obtidos na tabela 3 e posteriormente em seguida



**Table 3: Comparações com teste ANOVA**

ANOVA: Fator único (G06)			
GRUPOS	ABC	PSO	DE
Contagem	30	30	30
Soma	-200264,89	-197031,51	-197260,77
Média	-6675,4965	-6567,7172	-6575,3590
Variância	28194,059	22598,233	27529,632
ANOVA			
Fonte Variação	Entre grupos	Dentre grupos	Total
SQ	217023,030	2271335,83	2488358,86
gl	2	87	89
MQ	108511,515	26107,3084	
F	4,15636546		
Valor P	0,018880903		
F crítico	3,101295756		
ANOVA: Fator único (G08)			
GRUPOS	ABC	PSO	DE
Contagem	30	30	30
Soma	-2,80806936	-2,87475073	-2,807318
Média	-0,09360231	-0,09582502	-0,09357726
Variância	0,00014821	2,64367816	0,00014811
ANOVA			
Fonte Variação	Entre grupos	Dentre grupos	Total
SQ	0,00009993	0,00859339	0,00869332
gl	2	87	89
MQ	0,00004996	0,00009877	
F	0,50587291		
Valor P	0,60474157		
F crítico	3,10129575		
ANOVA: Fator único (G11)			
GRUPOS	ABC	PSO	DE
Contagem	30	30	30
Soma	22,5223288	22,5439352	22,6262467
Média	0,75074429	0,75146450	0,75420822
Variância	3,69957111	0,00001973	0,00009077
ANOVA			
Fonte Variação	Entre grupos	Dentre grupos	Total
SQ	0,00020045	0,00331194	0,00351239
gl	2	87	89
MQ	0,00010022	0,00003806	
F	2,63283124		
Valor P	0,07759664		
F crítico	3,10129575		

realizando o teste de Tukey apresentado na tabela 4 onde o ABC demonstrou um desempenho superior aos demais algoritmos nas 30 execuções.

Todavia, foram encontrado diferenças com relação ao algoritmo G06. Isso também ficou comprovado ao analisar o valor de P que é sempre superior a 0.5 nas funções escolhidas G08, G011 e inferior a 0.5 na função G06. Analisando os algoritmos pelo teste de Tukley na tabela 4 verificamos que realmente não existiu diferença significativa entre os algoritmos G08 e G11 porém na função

**Table 4: Teste de Tukey para as funções G06, G08 e G11**

Function G06			
	ABC	PSO	DE
ABC		<b>0,04478</b>	0,06694
PSO	3,4669		0,9836
DE	3,223	0,2459	
Function G08			
	ABC	PSO	DE
ABC		0,663	1
PSO	1,225		0,6569
DE	0,0138	0,242	
Function G11			
	ABC	PSO	DE
ABC		0,895	0,0868
PSO	0,6353		0,2097
DE	3,0568	2,42	

G06 o algoritmo PSO se destaca por sua diferença entre os outros dois algoritmos. Observa-se que em todas as funções, os resultados demonstraram superioridade do algoritmo ABC na maioria dos resultados e experimentos obtidos.

Conclui-se que os três algoritmos podem ser utilizados na solução de otimização dessas funções quando trabalhado com apenas duas variáveis, e que a diferença do algoritmo PSO na função G06 provavelmente foi devido ajuste em parâmetro de tempo de execução. Todo o código-fonte deste trabalho e os demais algoritmos implementados na linguagem Python encontram-se disponível no link: <https://colab.research.google.com/drive/1Jh43EzMWZ-mAShheZZB89Hfw9GdvAK7?usp=sharin>.

Como direcionamento em trabalhos futuros, determina-se realizar uma análise de maior profundidade e estender os parâmetros extraíndo-se mais otimização e desempenho dos algoritmos utilizados. Também encontrar formas de estabelecer pontos relevantes aplicando funções de Benchmark com valores maiores de modo a produzir soluções com maior precisão.

## REFERENCES

- [1] BAHAMISH, HESHAM AWADH A.; ABDULLAH, R. S. R. A. A. protein tertiary structure prediction using artificial bee colony algorithm. *2009 Third Asia International Conference on Modelling Simulation* (2009). IEEE, 2009. p. 258-263.
- [2] BROWNLEE, J. *Clever algorithms nature inspired programming recipes*. Jason Brownlee.
- [3] CRESTANI, V. Crestani, v. (2017). classificação de mensagens de e-mail utilizando o algoritmo artificial bee colony. 1.4 :293-311.
- [4] DANCHIN, ETIENNE; GIRALDEAU, L.-A. C. F. Behavioural ecology.
- [5] DUARTE, G. R. Comparação de técnicas de penalização adaptativas no algoritmo colônia de abelhas artificiais aplicado a problemas de otimização em engenharia. *VETOR-Revista de Ciências Exatas e Engenharias* (2015). v. 25, n. 2, p. 2-24, 2015.
- [6] FISCHER, I. Using genetic algorithms for simulation of social dilemmas. *New Issues and Paradigms in Research on Social Dilemmas* (2008).
- [7] GOLDBERG, D. E. Genetic algorithms in search, optimization, and machine learning. Addison Wesley Publishing Company, Inc. Universidade do Alabama, Reading, MA, 412.
- [8] JONG, K. A. D. An analysis of the behavior of a class of genetic adaptive systems.
- [9] KARABOGA, DERSIS; AKAY, B. O. Artificial bee colony (abc) optimization algorithm for training feed-forward neural networks. In: International conference on modeling decisions for artificial intelligence. Springer, Berlin, Heidelberg. p. 318-329.
- [10] KARABOGA, DERSIS; OZTURK, C. A novel clustering approach: Artificial bee colony (abc) algorithm. Elsevier B.V.

- [11] KARABOGA, D. An idea based on honey bee swarm for numerical optimization. *IJCSNS International Journal of Computer Science and Network Security* (2005). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
- [12] KENNEDY, JAMES; EBERHART, R. Particle swarm optimization. *Proceedings of ICNN'95-International Conference on Neural Networks*. IEEE, 1995. p. 1942-1948.
- [13] LIANYING, ZHOU; FENGYU, L. A swarm intelligence based intrusion detection technique. *IJCSNS International Journal of Computer Science and Network Security* (2006). v. 6, n. 7, p. 146e50, 2006.
- [14] LIENIG, J., AND THULASIRAMAN, K. A genetic algorithm for channel routing in vlsi circuits. *evolutionary computation*. 1.4 :293-311.
- [15] LINDEN, R. Algoritmos genéticos. 3ª ed. Ciência Moderna.
- [16] MOLLINETTI, M. A. F. E. A. Análise de performance do imperialist competitive algorithm através de funções de benchmark. In: Encontro Anual de Tecnologia da Informação e Semana Acadêmica de Tecnologia da Informação – EATI, ano 3, n. 1. Frederico Westphalen: EATI, p. 13-21.
- [17] SHUKRAN, MOHD AFIZI MOHD; CHUNG, Y. Y. Y. W. C. W. N. Z. A. M. A. Artificial bee colony based data mining algorithms for classification tasks. Published by Canadian Center of Science and Education, 2011.
- [18] STORN, R. Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical report, International Computer Science Institute* (1995). IEEE, 1995. p. 1942-1948.
- [19] TOSATTI, M. A., G. F. J. C. F. F. R. F. A. B. K. J. P. S. E. D. . . H. F. Algoritmo híbrido genético-fuzzy aplicado em câncer de próstata. *Hifen*, 32(62).