



**UEPB**

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

**YGOR ALCÂNTARA DE MEDEIROS**

**ANÁLISE E ESTUDO COMPARATIVO SOBRE TOPOLOGIAS DE  
COMUNICAÇÃO ENTRE PARTÍCULAS NO ALGORITMO VAGALUME**

**PATOS  
2017**

**YGOR ALCÂNTARA DE MEDEIROS**

**ANÁLISE E ESTUDO COMPARATIVO SOBRE TOPOLOGIAS DE  
COMUNICAÇÃO ENTRE PARTÍCULAS NO ALGORITMO VAGALUME**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Graduado em Bacharelado em Ciência da Computação.

Área de concentração: Algoritmos Evolutivos.

Orientador: Prof. Dr. Wellington Candeia de Araujo.

**PATOS  
2017**

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

M488a Medeiros, Ygor Alcântara de

Análise e estudo comparativo sobre topologias de comunicação entre partículas no Algoritmo Vagalume [manuscrito] / Ygor Alcantara de Medeiros. - 2017.  
24 p. : il. color.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas, 2017.

"Orientação: Prof. Dr. Wellington Candeia de Araujo, CCEA".

1. Algoritmos. 2. Algoritmo Vagalume. 3. Topologias. 4. Metodologia Benchmark. I. Título.

21. ed. CDD 004


Ygor Alcântara de Medeiros

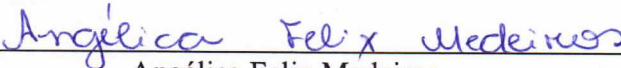
**ANÁLISE E ESTUDO COMPARATIVO SOBRE TOPOLOGIAS DE  
COMUNICAÇÃO ENTRE PARTÍCULAS NO ALGORITMO VAGALUME**


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

Aprovado em 01 de fevereiro de 2017

**BANCA EXAMINADORA**

  
Wellington Candeia de Araujo  
(Orientador)

  
Angélica Felix Medeiros  
(Examinadora)

  
Jefferson Felipe Silva de Lima  
(Examinador)

# ANÁLISE E ESTUDO COMPARATIVO SOBRE TOPOLOGIAS DE COMUNICAÇÃO ENTRE PARTÍCULAS NO ALGORITMO VAGALUME

Ygor Alcântara de Medeiros\*

## RESUMO

A busca pela sobrevivência e comida dos vagalumes em comunicação mútua através de luzes brilhantes que emitem de seus corpos, efeito conhecido como bioluminescência, deu os fundamentos para criar um algoritmo inspirado neste comportamento social. Apesar de o algoritmo ser relativamente ótimo no seu desempenho em funções de otimização e problemas do mundo real, o algoritmo acaba muitas vezes ficando preso em uma solução local no seu espaço de busca. Com base neste contexto, este trabalho tem como objetivo melhorar essa questão inserindo as topologias de comunicação entre partículas do algoritmo PSO no algoritmo vagalume, realizando assim uma análise empírica em cima dos resultados obtidos. Com os resultados obtidos durante o experimento, foi possível observar que as topologias Von Neumann e Star se sobressaíram no quesito desempenho e custo computacional do que as demais topologias envolvidas neste trabalho.

**Palavras-Chave:** Algoritmos. Algoritmo Vagalume. Topologias. Metodologia Benchmark.

## 1 INTRODUÇÃO

Os problemas de otimização estão intrinsecamente presentes na humanidade, as pessoas sempre buscam aperfeiçoar e buscar uma solução ótima para um problema definido que tentam resolver com base nos seus objetivos, conhecimento e suas limitações a respeito do mesmo. Este tipo de problema se aplica a várias áreas do conhecimento humano, como: matemática computacional, física, arquitetura, economia, química e biologia. Os problemas de otimização sempre mudam com o decorrer do tempo, assim como o conhecimento humano sobre diversas áreas de estudos, tornando cada vez mais difícil em encontrar a melhor solução para determinado problema.

Na matemática este tipo de problema está relacionado a problemas de maximização e minimização de funções matemáticas relacionadas com múltiplas variáveis e um espaço de busca limitado, onde se precisa utilizar métodos de otimização para se encontrar a solução ótima de uma função escolhida. Pensando nessa lógica o matemático acaba levando várias horas e esforço para se conseguir o melhor resultado desejado e inevitavelmente propenso ao erro. Nesse sentido os algoritmos computacionais acabam ajudando a diminuir o tempo e o

---

\* Aluno de Graduação em Ciência da Computação na Universidade Estadual da Paraíba – Campus VII.  
Email: ygoralcantara5@gmail.com

esforço para buscar o melhor resultado, transformando assim o processo de otimização mais automatizado, necessitando apenas em transferir a linguagem matemática para a linguagem computacional.

Uma das formas de se usufruir da computação em funções matemáticas é utilizando algoritmos meta-heurísticos baseados na natureza, ou seja, algoritmos bio-inspirados. Os algoritmos populacionais possuem o mesmo objetivo, representar de forma computacional o comportamento social de uma população de animais ou insetos da natureza, como pássaros, enxames de abelhas, colônia de formigas, morcegos, vagalumes e entre outros. Entre esses algoritmos meta-heurísticos, os algoritmos baseados em população, tem obtido grande sucesso em pesquisas e estudos em solucionar problemas de otimização com diversas variáveis e soluções locais.

Um algoritmo criado recentemente por Yang (2010) que vem chamando atenção na comunidade acadêmica foi o algoritmo fundamentado no comportamento social de uma população de vagalumes. A busca pela sobrevivência leva os vagalumes a se comunicarem entre si através de luzes brilhantes emitidas de seus corpos. Através desta interação Yang teve a ideia de modelar este comportamento específico em um mecanismo de otimização de busca avançada. O algoritmo vagalume é baseado em outro algoritmo bio-inspirado, o PSO (Particle Swarm Optimization) ou enxame de partículas.

Pesquisas e estudos acerca do PSO (LI e GUO, 2014) sugerem melhorias significativas para o algoritmo em questão de desempenho e tempo de processamento. Uma delas são as topologias. Uma topologia é uma estrutura não visual que representa como as partículas se comunicam entre si em um espaço de busca limitado em procura da melhor solução.

Seguindo esse pressuposto, o objetivo deste trabalho é inserir as topologias existentes do algoritmo PSO no algoritmo vagalume, posteriormente realizar uma análise empírica entre as topologias no algoritmo vagalume otimizando a minimização de funções matemáticas com múltiplas dimensões, mostrando assim os resultados obtidos e provando que as topologias do PSO funcionam no algoritmo vagalume.

## **2 ALGORITMO VAGALUME**

Os vagalumes são um grupo de insetos da natureza também conhecidos pelo nome pirilampo, são famosos pela sua capacidade de produção e emissão de luzes fluorescentes a partir da região inferior do abdômen. Esse fenômeno é chamado de bioluminescência e serve

para os vagalumes se atraírem entre si para o acasalamento ou alertar sobre o perigo de predadores.

A partir deste comportamento específico, Yang (2010) propôs um algoritmo meta-heurístico baseado na bioluminescência dos vagalumes. Para que o algoritmo funcione de forma correta algumas características em particular devem ser levadas em consideração: 1. Todos os vagalumes possuem apenas um sexo, ou seja, qualquer vagalume pode ser atraído um pelo outro; 2. A luminosidade de cada vagalume serve de base para o objetivo a ser alcançado, quanto maior for a emissão de luzes, então maior será a probabilidade de se encontrar a solução ótima do problema; 3. O vagalume com maior luminosidade atrai o de menor taxa de emissão para sua localidade atual.

Existem duas características importantes no funcionamento do algoritmo que devem ser observadas: A variação da luminosidade e a formulação da atratividade. Podemos observar que a atratividade de um vagalume sempre será determinada pelo seu brilho, e que o mesmo está associado ao valor de uma função objetivo que se quer alcançar. Em um caso mais simples como o de otimizar uma função matemática para maximizá-la ou minimizá-la, pode-se levar em consideração que a taxa de luminosidade de um vagalume  $i$ , que se encontra em uma posição  $X_i = (x_1, x_2, \dots, x_d)^T$ , pode ser selecionado como  $I(x) \rightarrow f(x)$ . Porém a atratividade é totalmente relativa, pois a mesma varia de acordo com o valor  $r_{ij}$ , ou seja, a distância entre o vagalume  $i$  e o vagalume  $j$ , pois a intensidade de luz dos vagalumes diminui em relação com a distância entre os dois pontos, afetando assim a atratividade que acaba variando relativamente com o coeficiente de absorção  $Y$  (YANG, 2009).

A luminosidade do vagalume varia de acordo com a distância  $r_{ij}$  exponencialmente, dessa maneira podemos tratar essa expressão como:

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

onde temos  $I_0$  sendo a luminosidade inicial e  $Y$  o coeficiente de absorção de luz.

A atratividade é relativamente proporcional à luminosidade do vagalume na posição  $X_i$  observado pelos seus vizinhos adjacentes, então podemos definir a atratividade  $\beta$  de um vagalume como:

$$\beta(r) = \beta_0 e^{-Yr^2} \quad (2),$$

onde  $\beta$  é a atratividade em  $r = 0$ . Geralmente o parâmetro  $Y$  é usado com o valor padrão  $Y = 1$ , podendo ser alterado, dependendo do tipo do problema. Porém vale salientar que se o

parâmetro receber o valor  $Y = 0$ , o algoritmo não funcionará, pois os vagalumes não se atrairão por ninguém.

A distância entre dois vagalumes pode ser medida através da fórmula matemática que calcula a distância entre dois pontos no plano cartesiano:

$$r_{ij} = \sqrt{(X_i - X_j)^2 - (Y_i - Y_j)^2} \quad (3).$$

Quando um vagalume  $i$  é atraído por outro vagalume  $j$ , a função matemática para calcular o movimento do vagalume  $i$  em direção ao vagalume  $j$ , se dá como:

$$X_i = X_i + \beta_0 e^{-Yr^2} (X_j - X_i) + \alpha \left( \text{rand} - \frac{1}{2} \right) \quad (4),$$

onde temos  $X_i$  sendo a posição atual do vagalume  $i$ , na segunda parte da equação temos a fórmula para calcular atratividade entre dois vagalumes (2), o terceiro componente  $\alpha \in [0,1]$  é um parâmetro de randomização do algoritmo vagalume; logo após temos  $\text{rand}$  que é um gerador aleatório que distribui valores uniformemente no intervalo 0 e 1.

A Figura 1 ilustra o pseudocódigo do algoritmo vagalume:

**Figura 1:** Pseudocódigo do algoritmo vagalume

---

Função Objetivo  $f(x)$ , com  $X_i = (x_1, x_2, \dots, x_d)^T$   
 Gere a população inicial dos vagalumes em posições aleatórias  $X_i (i = 1, 2, \dots, n)$   
 Defina a Intensidade de absorção de luz  $I_i$  em  $X_i$  que é determinada por  $f(X_i)$   
 Defina o coeficiente de absorção de luz  $Y$   
**while**( $t <$  quantidade máxima de iterações do algoritmo)  
   **for**  $i = 1 : n$  quantidade de vagalumes  
     **for**  $j = 1 : n$  quantidade de vagalumes  
       **if**( $I_i < I_j$ )  
         Calcule a distância  $r_{ij} = \sqrt{(X_i - X_j)^2 - (Y_i - Y_j)^2}$   
         Calcule a atratividade  $\beta(r) = \beta_0 e^{-Yr^2}$   
         Movimente  $I_i$  em direção a  $I_j$   
       **end if**  
     **end for**  
   **end for**  
 Classifique os vagalumes e encontre o melhor de todos  
**end while**

---

**Fonte:** Elaborado pelo autor

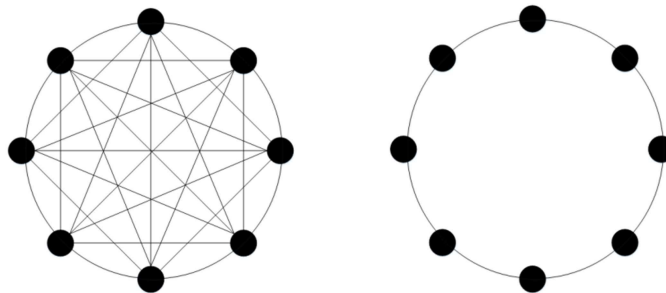


### 3 TIPOS DE TOPOLOGIAS

Algoritmos meta-heurísticos baseados em um comportamento social de uma população da natureza possuem uma forma das partículas se comunicarem entre si para coletar ou transferir informações sobre o mundo que os cercam, isso é chamado de topologia. Sem ela seria impossível das partículas se comunicarem com outras, causando assim uma quebra na estrutura social de uma população e os vagalumes passariam a procurar pela solução ótima de um problema independentemente da influência de seus vizinhos. As topologias promovem uma série de regras de como os vagalumes tem que se comunicar entre si, no algoritmo PSO existem várias delas, veremos com detalhes as mais utilizadas na comunidade acadêmica.

#### 3.1 Topologia Global e Local

**Figura 2:** (a) Topologia Global (b) Topologia Local



**Fonte:** Elaborado pelo Autor

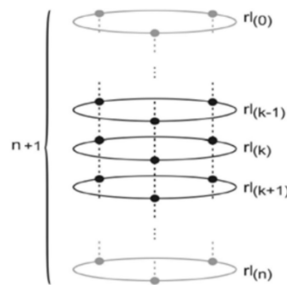
As topologias global e local (KENNEDY e EBERHART, 1995) são as topologias mais usadas e conhecidas na comunidade acadêmica. A topologia global, conforme ilustrado na Figura 2.a. Foi a primeira topologia a ser criada para o algoritmo PSO, possui uma rápida velocidade de convergência para a solução ótima. Porém o tempo de processamento de dados faz com que seja a mais lenta de todas as topologias, pois todas as partículas estão coletando ou transferindo informações entre si, isto acaba sendo vantajoso em problemas unimodais (problemas com apenas uma solução ótima) onde assim que uma partícula encontra uma solução ótima transfere esses dados para todas as outras partículas.

A topologia local ou simplesmente chamada de anel, conforme ilustrado na Figura 2.b, foi a primeira modificação em quesito de topologia para o algoritmo PSO. Possui uma lenta velocidade de convergência para a solução ótima, pois as partículas se comunicam apenas com seus vizinhos diretos. Em uma população  $N$  de partículas, a partícula  $i$  pode ter que

esperar  $N / 2$  iterações para coletar ou transferir dados sobre a possível solução encontrada. Porém é uma das topologias mais rápidas em tempo de processamento de dados e ótima em problemas multimodais (problemas com mais de uma solução ótima) pois acaba dividindo os vagalumes no espaço de busca, criando subgrupos, possibilitando assim uma busca mais ampliada pelas possíveis soluções.

### 3.2 Topologia Multi-Ring

**Figura 3:** Topologia Multi-Ring



**Fonte:** Elaborado por BASTOS-FILHO et al, 2008.

A Topologia Multi-Ring (BASTOS-FILHO et al, 2008), como ilustrado na Figura 3, foi baseada em outra topologia muito conhecida, a topologia local, como visto na Figura 2.b. Essa topologia é formada por múltiplos anéis onde as partículas além de se comunicarem diretamente com seus vizinhos como na topologia local, também se comunicam coletando e transferindo informações com seus vizinhos diretos superiores e inferiores, ou seja, um vagalume  $i$  que se encontra no anel  $rl_{(k)}$ , troca informações com:

$$\{rl_{(k)(i-1)}, rl_{(k)(i+1)}, rl_{(k-1)(i)}, rl_{(k+1)(i)}\} \quad (5),$$

com algumas exceções como a de um vagalume  $i$  que se encontra no primeiro anel  $rl_{(0)}$ , onde tem sua troca de informações com:

$$\{rl_{(1)(i+1)}, rl_{(0)(i-1)}, rl_{(0)(i+1)}\} \quad (6),$$

ou no último anel  $rl_{(n)}$ , onde possui as seguintes regras de comunicação:

$$\{rl_{(n-1)(i-1)}, rl_{(n)(i-1)}, rl_{(n)(i+1)}\} \quad (7).$$

Essa topologia também possui uma habilidade de girar um anel, mudando assim as posições de todos os vagalumes no anel caso o desempenho dos mesmos se mantenha

estático, ou seja, não mudem em certa quantidade de iterações. Para isso temos uma fórmula matemática que define o grau de rotação a ser inserido para o anel  $rl_{(k)}$ :

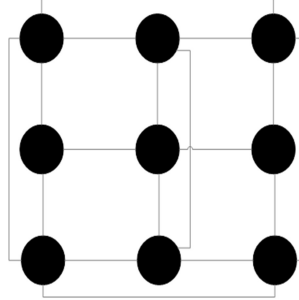
$$i = (i + d) \bmod (nl) \quad (8),$$

onde temos  $i$  que é o índice do vagalume,  $d$  que é a distância para se realizar a rotação e  $nl$  que é o número de vagalumes distribuídos entre cada anel da topologia. Depois de feita a rotação do anel  $rl_{(k)}$ , a nova vizinhança da partícula  $i$  será distribuída como:

$$\{rl_{(k)(i-1)}, rl_{(k)(i+1)}, rl_{(k+1)(i+d)}, rl_{(k-1)(i+d)}\} \quad (9).$$

### 3.3 Topologia Von Neumann

**Figura 5:** Topologia Von Neumann

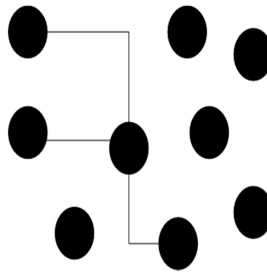


Fonte: Elaborado pelo Autor

A topologia Von Neumann (KENNEDY e MENDES, 2002), conforme ilustrada na Figura 5, possui uma estrutura gráfica que se parece com uma matriz onde todos os vagalumes são distribuídos igualmente entre si, formando uma espécie de malha. Toda a vizinhança das partículas é estática, ou seja, não sofre rotações ou se comunicam com outras partículas que não sejam seus vizinhos diretos. Um vagalume  $i$  coleta ou transfere informações apenas para seus vizinhos superiores, inferiores, direita e esquerda.

### 3.4 Topologia Random

**Figura 6:** Topologia Random

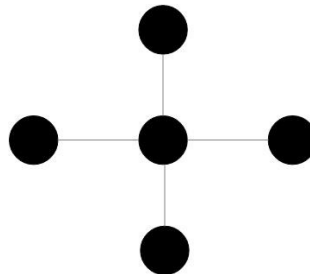


Fonte: Elaborado pelo Autor

A topologia random (CLERC, 2007), conforme ilustrado na Figura 6, é uma topologia onde são distribuídas todas as suas partículas de forma aleatória, ou seja, não possui uma estrutura gráfica onde todas as partículas devem obedecer à comunicação da mesma. Em uma população de vagalumes  $S$ , declara-se uma vizinhança  $K = N$ , onde  $N$  é a quantidade de vizinhos que uma partícula  $i$  terá, logo em toda iteração são sorteados aleatoriamente  $N$  vizinhos para coletarem ou transferirem informações para o vagalume  $i$ .

### 3.5 Topologia Star

Figura 7: Topologia Star

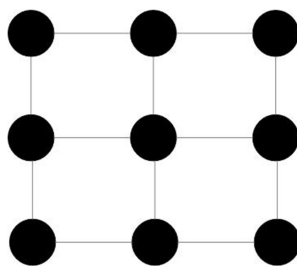


Fonte: Elaborado pelo Autor

A topologia Star (ENGELBRECHT, 2007), conforme ilustrado na Figura 7, é uma topologia onde sua principal característica é a partícula do meio de sua estrutura. Esta mesma partícula possui conexão com todos os vagalumes do enxame, funcionando como uma espécie de mediadora, trocando informações com outros vagalumes e atualizando sua posição apenas se melhorar a sua solução local.

### 3.6 Topologia Mesh

Figura 8: Topologia Mesh



**Fonte:** Elaborado pelo Autor

A topologia Mesh (KENNEDY e MENDES, 2002), conforme ilustrado na Figura 8, é uma topologia onde as partículas trocam informações apenas com seus vizinhos diretos. O vagalume no meio da estrutura da topologia se comunica apenas com quatro de seus vizinhos, enquanto que os vagalumes nas posições laterais se comunicam com três outros vagalumes, e por último os vagalumes nas diagonais se comunicam com apenas dois vagalumes.

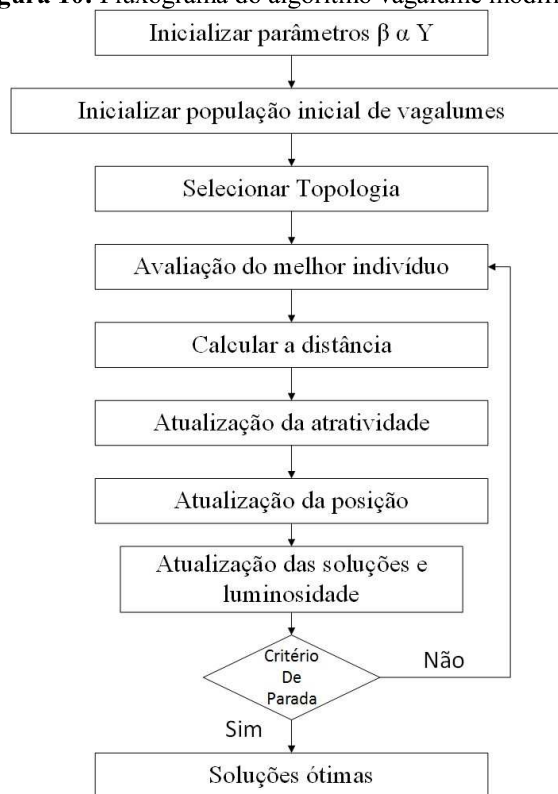
#### **4 MODIFICAÇÃO NO ALGORITMO VAGALUME**

O algoritmo vagalume utiliza a topologia global (Figura 2.a) como forma de troca de informações entre todos os vagalumes. A convergência da solução ótima é rápida, pois assim que um vagalume encontra a melhor solução, a mesma repassa esta informação para todos os outros vagalumes, assim todos os vagalumes acabam convergindo e indo rapidamente para o local da melhor solução. Porém isto acaba tendo um fator negativo em relação à rápida convergência, pois todos os vagalumes podem acabar ficando presos em um mínimo local sem conseguirem sair desta localidade por várias iterações.

Isto acaba sendo ruim para problemas multimodais, em que os vagalumes têm que procurar em vários espaços de busca ao mesmo tempo por várias soluções ótimas (FISTER, YANG e BREST, 2013). Além deste fator negativo, temos o tempo de processamento dos dados que acaba tendo um custo computacional elevado em comparação a outras topologias, pois a forma dos vagalumes se comunicarem e trocarem informações entre si é através da luminosidade de cada um, ou seja, o melhor vagalume acaba brilhando mais forte e os vagalumes adjacentes tendem a ir em direção a essa luz.

Com base neste contexto, a modificação e adaptação do algoritmo vagalume se encontram de acordo com o fluxograma ilustrado na Figura 9:

**Figura 10:** Fluxograma do algoritmo vagalume modificado



**Fonte:** Elaborado pelo autor

A modificação realizada em relação ao pseudocódigo (Figura 1) apresentado antes inicia nos laços de repetição, onde nesta parte é inserida a estrutura da topologia escolhida e logo após é comparada a luminosidade do vagalume  $i$  com a do vagalume  $j$ , obedecendo às regras de comunicação da estrutura gráfica da topologia escolhida.

## 5 EXPERIMENTOS

Para podermos realizar uma análise empírica com base nas topologias do algoritmo de enxame de partículas inseridas no algoritmo vagalume, foram selecionadas algumas funções matemáticas (SURJANOVIC e BINGHAM, 2013) de otimização para minimizar e encontrar a melhor solução ótima em ambas as funções. Foram definidas duas funções matemáticas unimodais e quatro funções multimodais. As funções de benchmark utilizadas para realizar o teste de desempenho estão listadas na Tabela 1 com os respectivos parâmetros definidos para cada função.

A importância do teste de benchmark em encontrar valores como a média e o desvio padrão, se dá pelo fato de que valores estatísticos obtidos com os testes possuem mais credibilidade em analisarmos se outras topologias inseridas no algoritmo vagalume retornam

um resultado satisfatório com base na média dos valores finais retornados a cada execução, ou seja, a solução ótima de determinada função benchmark retornada a cada execução do algoritmo e a variação da média desses valores.

**Tabela 1:** Funções Benchmark

<b>Função</b>	<b>Fórmula</b>	<b>Dimensões</b>	<b>Limites</b>
Sphere $f(x1)$	$f(x) = \sum_{i=1}^d Xi^2$	30	[-5,5]
Rosenbrock $f(x2)$	$f(x) = \sum_{i=1}^{d-1} [100(Xi + 1 - Xi^2) + (Xi - 1)^2]$	5	[-5,5]
Ackley $f(x3)$	$f(x) = -a \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i)\right) + a + \exp(1)$	30	[-5, 5]
Michalewicz $f(x4)$	$f(x) = -\sum_{i=1}^d \text{seno}(Xi) \text{seno}^{2m}\left(\frac{iXi^2}{\pi}\right)$	5	[0,π]
Hartman $f(x5)$	$f(x) = -\sum_{i=1}^4 \alpha_i \exp\left(-\sum_{j=1}^3 A_{ij}(X_j - P_{ij})^2\right)$	3	[0, 1]
Goldstein $f(x6)$	$f(x) = [1 + (X_1 + X_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	3	[-1, 0]

**Fonte:** Elaborado pelo autor

Os parâmetros do algoritmo vagalume como o coeficiente de absorção de luz  $\Upsilon$ , atratividade individual de cada vagalume  $\beta$  e parâmetro de randomização  $\alpha$ , assim como algumas configurações particulares de cada topologia estão definidos na Tabela 2.

**Tabela 2:** Configuração do Algoritmo Vagalume

<b>Parâmetros</b>	<b>Valores</b>
Coefficiente de absorção de luz $\Upsilon$	1.0
Atratividade individual $\beta$	1.0
Parâmetro de randomização $\alpha$	1.0
Qº de anéis da Topologia Multi-Ring	5 anéis
Qº de vizinhos da Topologia Random	50 vizinhos
Critério de parada	5.000 iterações
Tamanho da população	100 vagalumes

**Fonte:** Elaborado pelo autor

Como ilustrado na Tabela 2, foram definidos 100 vagalumes e 5.000 iterações para cada execução do algoritmo, pelo fato de algumas funções serem multimodais e possuírem uma complexidade elevada em encontrar o mínimo global.

Na topologia Multi-Ring foram definidos 5 anéis, pelo fato dos resultados obtidos em teste de benchmark no trabalho de Caraciolo (2008) terem saído melhores com 5 anéis na topologia.

Na topologia Random foram definidos 50 vizinhos, metade da quantidade de vagalumes utilizados para o teste de benchmark, desta forma a topologia acaba convergindo de forma mais rápida do que as demais, por ter vizinhos totalmente aleatórios.

A quantidade de execuções do algoritmo por função e topologia foi definida em 100 vezes, desta forma fica melhor de observamos a média obtida das melhores soluções encontradas a cada execução e o desvio padrão destes valores.

Na Tabela 3 podemos visualizar a média dos resultados obtidos em forma tabular para as funções benchmark listadas na Tabela 1:

**Tabela 3:** Média do desempenho das funções benchmark

Função	Topologia Global	Topologia Local	Topologia Multi-Ring	Topologia Von Neumann	Topologia Random	Topologia Star	Topologia Mesh
$f(x1)$	2.1631e-05	0.083993	0.015516	0.0090859	0.23975	0.63959	0.052899
$f(x2)$	0.0008547	0.016611	0.028951	0.025623	0.91725	0.14847	0.019144
$f(x3)$	0.0041373	0.36426	0.11847	0.08635	0.72736	0.27094	0.096268
$f(x4)$	-4.6779	-4.6021	-4.6269	-4.6297	-4.3889	-4.4787	-4.6272
$f(x5)$	-3.8628	-3.8624	-3.8625	-3.8626	-3.8619	-3.8614	-3.8625
$f(x6)$	3.0001	3.0129	3.0048	3.0039	3.002	3.0139	3.0051

Fonte: Elaborado pelo autor

Na Tabela 4 podemos visualizar o desvio padrão dos resultados obtidos em forma tabular para as funções benchmark listadas na Tabela 1:

**Tabela 4:** Desvio padrão do desempenho das funções benchmark

Função	Topologia Global	Topologia Local	Topologia Multi-Ring	Topologia Von Neumann	Topologia Random	Topologia Star	Topologia Mesh
$f(x1)$	6.1466e-06	0.018032	0.0044578	0.0029132	0.048704	0.88491	0.052011
$f(x2)$	0.0008675	0.016004	0.048815	0.044276	0.60986	0.64438	0.037956
$f(x3)$	0.0005839	0.057664	0.021189	0.014891	0.11533	0.064706	0.014473
$f(x4)$	0.01653	0.052775	0.046307	0.039447	0.097674	0.14723	0.04091
$f(x5)$	5.4411e-05	0.0003840	0.0002734	0.0002117	0.0007067	0.0016522	0.0003605
$f(x6)$	0.0001317	0.013822	0.0040595	0.0041013	0.0021597	0.015733	0.0048921

Fonte: Elaborado pelo autor

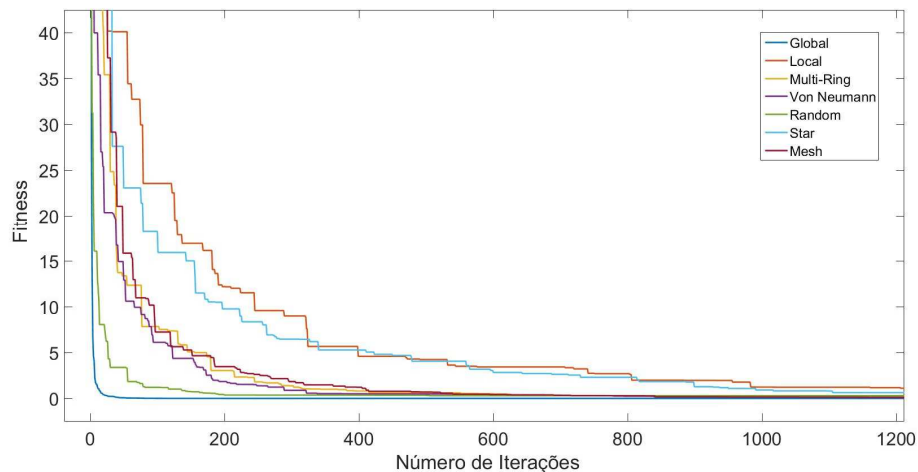


Na Tabela 3 e 4 podemos observar que a topologia global obteve melhor desempenho em relação às outras topologias inseridas no algoritmo vagalume. Isso se dá pelo fato de os vagalumes nesta topologia se comunicarem todos com todos, dessa maneira acaba convergindo de forma mais rápida. Porém o mínimo global das funções benchmark listado na Tabela I foi alcançado pelas demais topologias apresentadas neste trabalho com um menor custo computacional em relação à topologia global.

Para analisarmos o custo computacional de cada topologia em relação às funções benchmark, foram realizados testes de convergência com o seu devido tempo de execução para cada função benchmark e topologias apresentadas neste artigo, seguindo as configurações ilustradas na Tabela 2, com exceção do número de execuções.

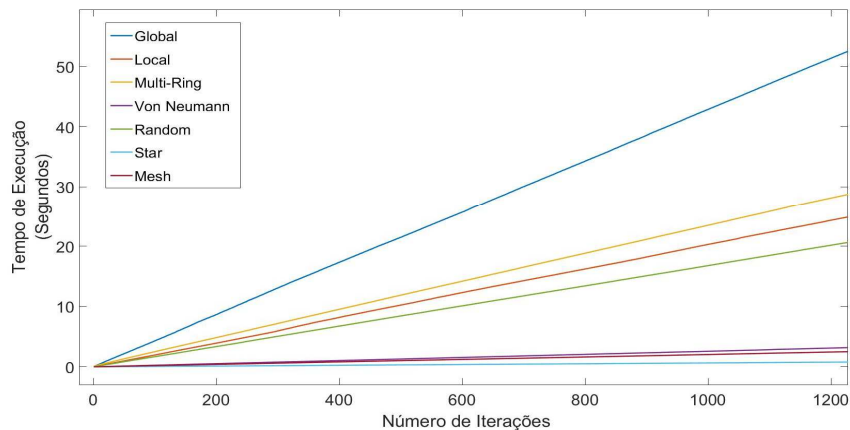
Na Figura 10 podemos observar a curva de convergência da função *Sphere* em relação ao número de iterações definidas na Tabela 2, enquanto que na Figura 11 mostra-se o tempo de execução para a mesma:

**Figura 10:** Gráfico de Convergência da Função *Sphere*



**Fonte:** Elaborado pelo autor

**Figura 11:** Gráfico do Custo Computacional da Função *Sphere*



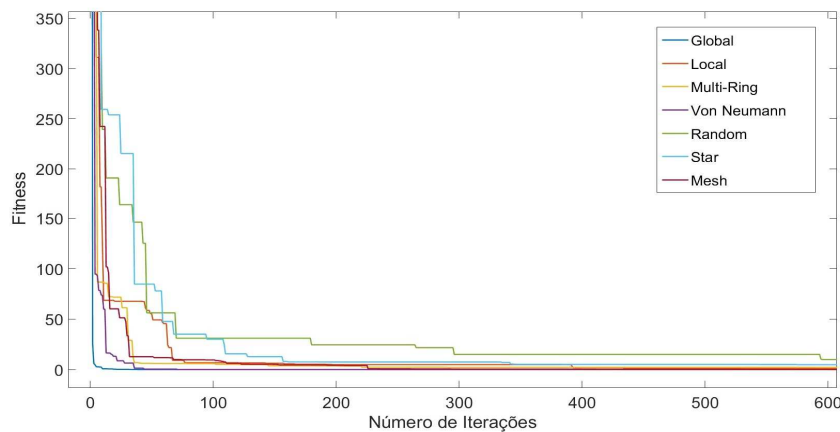
**Fonte:** Elaborado pelo autor

Em relação as Figuras 10 e 11, os dados foram ilustrados até a iteração 1200 para melhor visualização das informações obtidas, pois depois da iteração 1200, todas as topologias se estabilizaram nos seus respectivos resultados.

Podemos observar que todas as topologias apresentadas conseguiram alcançar o mínimo global da função *Sphere*, que é  $f(x_1) = 0$ , porém a topologia global se saiu melhor em quesito desempenho, pois na iteração 5.000 alcançou a melhor solução de  $f(x_1) = 1.5946e - 05$ . Mas em quesito custo computacional foi a pior de todas as topologias, pois demorou 210 segundos para que o algoritmo fosse executado, enquanto que a topologia Star custou apenas 3 segundos e alcançou a melhor solução de  $f(x_1) = 0.0516$ .

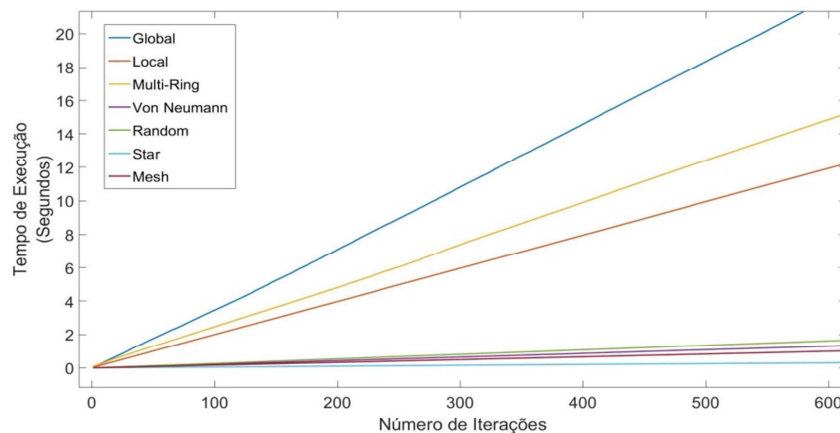
Na Figura 12 é ilustrada a curva de convergência da função *Rosenbrock* com o seu custo computacional ilustrado na Figura 13:

**Figura 12:** Gráfico de Convergência da Função *Rosenbrock*



Fonte: Elaborado pelo autor

**Figura 13:** Gráfico do Custo Computacional da Função *Rosenbrock*



Fonte: Elaborado pelo autor

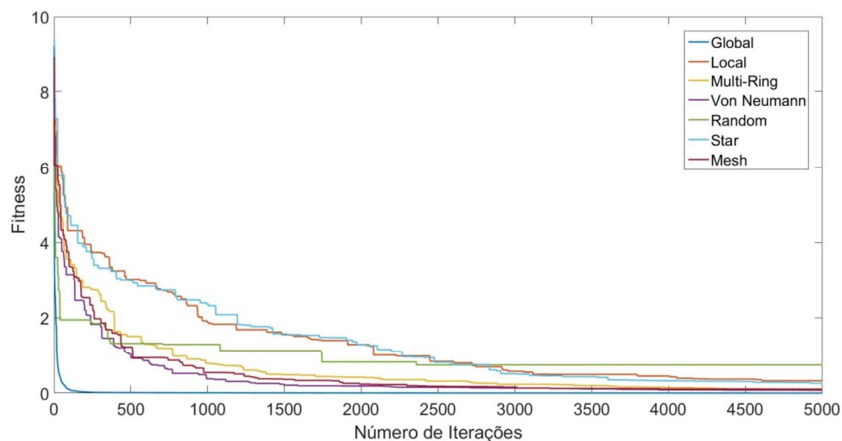
Em relação as Figuras 12 e 13, os dados foram ilustrados até a iteração 600 para melhor visualização das informações obtidas.

Apesar da função Rosenbrock ser uma função unimodal, a mesma é uma função bastante complexa de se encontrar a melhor solução de  $f(x_2) = 0$ . Quanto maior for o número de variáveis de entrada para a função, mais complexo a mesma se torna, por isso neste artigo foram definidos 5 dimensões para esta função, como demonstrado na Tabela I.

Todas as topologias alcançaram o mínimo global desta função. A topologia Star foi a mais eficiente de todas as topologias em quesito desempenho e custo computacional, além de ter executado o algoritmo com apenas 2,5 segundos, a mesma conseguiu alcançar  $f(x_2) = 0.0036$ , ficando apenas um pouco atrás da topologia global com  $f(x_2) = 0.0015$ .

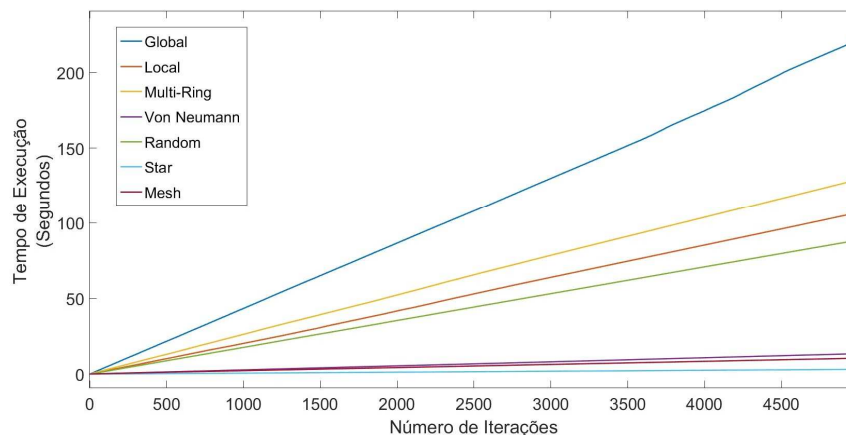
Na Figura 14 é ilustrada a curva de convergência da função Ackley com o seu custo computacional ilustrado na Figura 15:

**Figura 14:** Gráfico de Convergência da Função Ackley



Fonte: Elaborado pelo autor

**Figura 15:** Gráfico do Custo Computacional da Função Ackley

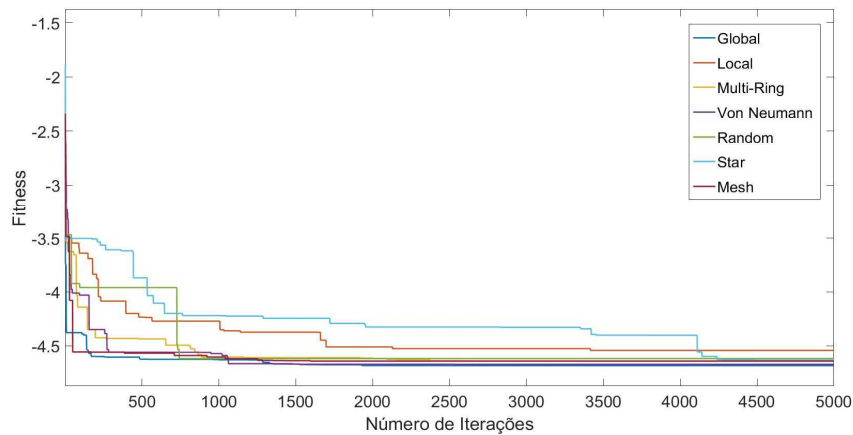


Fonte: Elaborado pelo autor

Na função Ackley, a topologia global demorou 221 segundos para a execução do algoritmo, enquanto que na topologia Von Neumann foram apenas 13 segundos, apesar da alta complexidade desta função ser multimodal e de que foram testados 30 variáveis de entrada, a topologia Von Neumann conseguiu alcançar a melhor solução de  $f(x_3) = 0.0767$ .

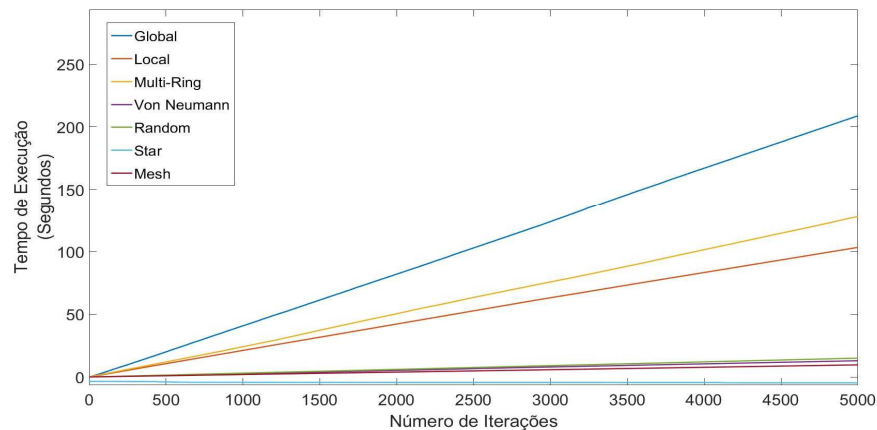
Na Figura 16 é ilustrada a curva de convergência da função Michalewicz com o seu custo computacional ilustrado na Figura 17:

**Figura 16:** Gráfico de Convergência da Função Michalewicz



Fonte: Elaborado pelo autor

**Figura 17:** Gráfico do Custo Computacional da Função Michalewicz



Fonte: Elaborado pelo autor

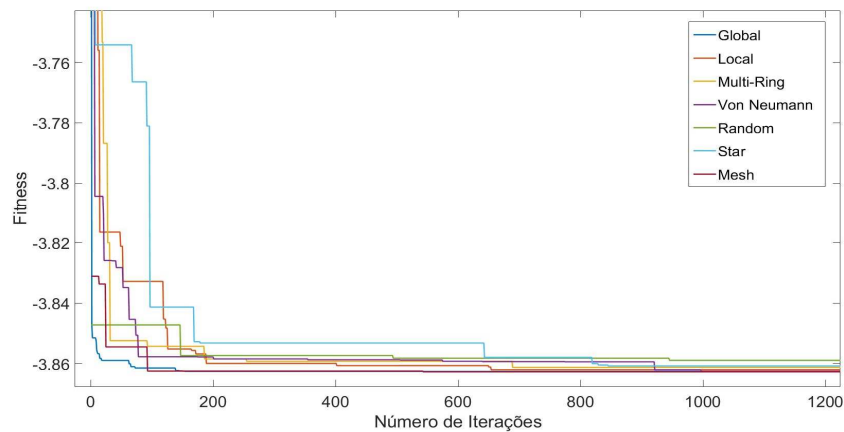
Na função Michalewicz, apesar de poucas variáveis de entrada, possui uma complexidade desafiador para muitos algoritmos meta-heurísticos, porém o algoritmo vagalume é conhecido por se sobressair em relação a esta complexidade desta função multimodal.

A topologia Von Neumann foi a que mais se igualou com a topologia global em quesito desempenho, pois alcançou a melhor solução de  $f(x_4) = -4.6729$ , enquanto a

topologia global conseguiu alcançar o mínimo global da função com  $f(x_4) = -4.6856$ , porém demorou 218 segundos para a execução do algoritmo, enquanto a topologia Von Neumann conseguiu em 14 segundos. A topologia Random também teve um resultado satisfatório em relação ao seu custo computacional com apenas 15 segundos de execução, porém os vagalumes acabaram ficando presos em uma solução local de  $f(x_4) = -4.6196$ , este tipo de problema é bem conhecido no algoritmo vagalume, como foi citado antes na seção IV.

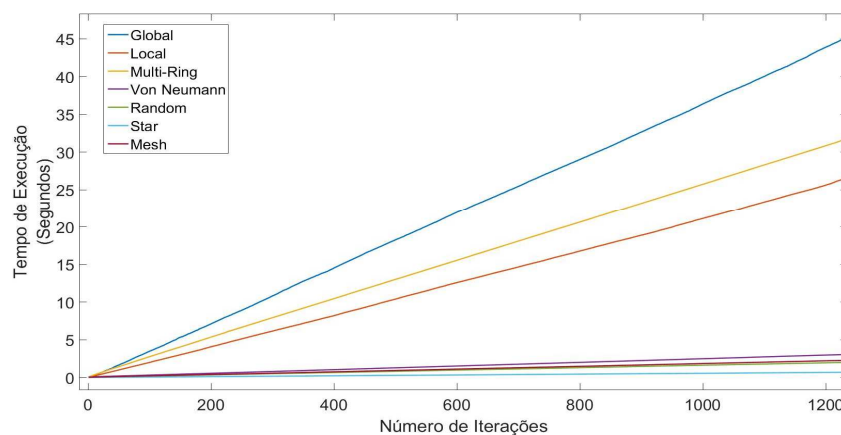
Na Figura 18 é ilustrada a curva de convergência da função Hartman com o seu custo computacional ilustrado na Figura 19:

**Figura 18:** Gráfico de Convergência da Função Hartman



Fonte: Elaborado pelo autor

**Figura 19:** Gráfico do Custo Computacional da Função Hartman



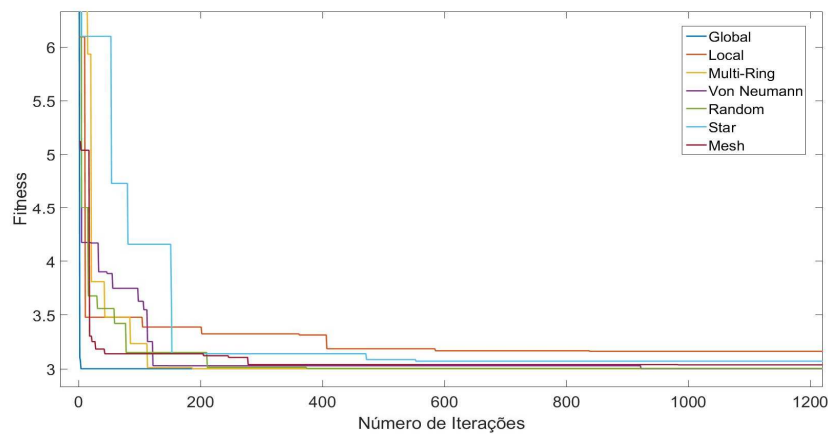
Fonte: Elaborado pelo autor

Apesar da função Hartman ser uma função multimodal, a mesma se provou ser uma função pouco complexa pra se encontrar o mínimo global por possuir apenas 3 variáveis de entrada. Com exceção da topologia random, as demais topologias conseguiram alcançar o

mínimo global da função de  $f(x_5) = -3.86$ , porém a topologia random não ficou muito atrás deste valor, conseguindo alcançar  $f(x_5) = -3.8599$  com um custo computacional de apenas 8 segundos, melhor até mesmo da topologia Von Neumann que conseguiu executar o algoritmo em apenas 12 segundos, que vinha se provando até então, a melhor topologia em custo computacional.

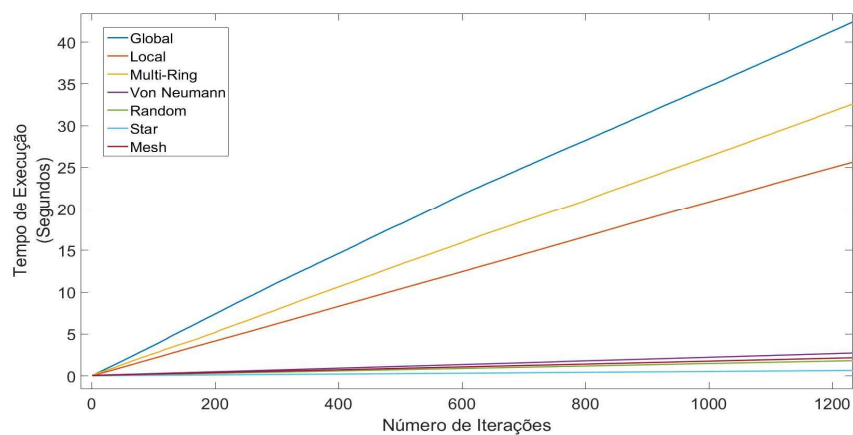
Na Figura 20 é ilustrada a curva de convergência da função Goldstein com o seu custo computacional ilustrado na Figura 21:

**Figura 20:** Gráfico de Convergência da Função Goldstein



**Fonte:** Elaborado pelo autor

**Figura 21:** Gráfico do Custo Computacional da Função Goldstein



**Fonte:** Elaborado pelo autor

Em relação as Figuras 20 e 21, os dados foram ilustrados até a iteração 1200 para melhor visualização das informações obtidas.

Na função Goldstein, todas as topologias conseguiram alcançar o mínimo global de  $f(x_5) = 3$ , destaque para a topologia star que conseguiu executar o algoritmo em apenas 2,5 segundos.

Diante de todos os testes realizados neste artigo, para problemas que envolvem multiobjetivos, a topologia global provou ser a mais eficiente em desempenho e péssima em custo computacional.

Por outro lado, as topologias Von Neumann e star provaram ser as topologias mais eficientes em custo computacional, não ficando muito atrás da topologia global em desempenho e inclusive alcançando o mínimo global da maioria das funções apresentadas neste artigo. As únicas topologias que não obtiveram resultados satisfatórios nos testes de custo computacional foram as topologias local e multi-ring, pelo fato da multi-ring ter sido baseada na topologia local, e os vagalumes apenas se comunicarem com seus vizinhos diretos, isto influencia significativamente no custo computacional do algoritmo, ocasionando em uma necessidade de mais iterações para convergir e encontrar a melhor solução das funções benchmark do que as outras topologias.

## **6 CONSIDERAÇÕES FINAIS**

Podemos observar que os resultados de todas as topologias são bem próximos da topologia global usada no algoritmo original do vagalume. Sugerindo assim que qualquer topologia existente que seja desenvolvida originalmente para o algoritmo de enxame de partículas pode ser inserida no algoritmo vagalume sem perder qualidade no desempenho, ganhando uma melhoria significativa em custo computacional. Isso pode ajudar a futuros pesquisadores a buscar outros tipos de topologias para melhorar o desempenho do algoritmo vagalume em funções de otimizações ou problemas relacionados além da pesquisa científica no ambiente acadêmico e no ambiente industrial.

### **ANALYSIS AND COMPARATIVE STUDY ON COMMUNICATION TOPOLOGIES BETWEEN PARTICLES IN FIREFLY ALGORITHM**

#### **ABSTRACT**

The search for survival and food of fireflies in mutual communication through glowing lights emitting from their bodies, effect known as bioluminescence, led the basis to create an inspired algorithm in this social behavior. Despite the operation of the algorithm be relatively fine in performance optimization functions and real-world problems, the algorithm often ends up getting stuck in an local solution in the search space. Based on this context, this work aims to improve this issue placing the communication topologies between PSO algorithm particles in the firefly algorithm, thus performing an empirical analysis on the obtained results. The results obtained during the experiment, it was possible to observe that the topologies Von

Neumann and Star stand out in the performance and computational cost than the other topologies involved in this paper.

**Keywords:** Algorithm. Firefly. Topology. Benchmark.

## REFERÊNCIAS

YANG, Xin-She. Nature-Inspired Metaheuristic Algorithms. Luniver Press. **Beckington, U.K.**, n. 2, p 81-89, 2010;

LI, Fenglin; GUO, Jian. Topology Optimization of Particle Swarm Optimization. In: **International Conference in Swarm Intelligence**. Springer International Publishing, 2014. p. 142-149;

YANG, Xin-She. Firefly algorithms for multimodal optimization. In: **International Symposium on Stochastic Algorithms**. Springer Berlin Heidelberg, 2009. p. 169-178;

KENNEDY, J. e EBERHART, R. Particle Swarm Optimization. In: **Proceedings of IEEE International Conference on Neural Networks**. p. 1942-1948, 1995;

BASTOS-FILHO, Carmelo JA et al. Multi-ring particle swarm optimization. In: **2008 10th Brazilian Symposium on Neural Networks**. IEEE, 2008. p. 111-116;

KENNEDY, J. e MENDES, R. Population Structure and Particle Swarm Performance. In: **Proceedings of the IEEE Conference on Evolutionary Computation**. p.1671-1676, 2002;

CLERC, Maurice. Back to random topology. **Relatório Técnico, mar, 2007**;

ENGELBRECHT, Andries P. **Computational intelligence: an introduction**. John Wiley & Sons, 2007. p. 628;

FISTER, Iztok; YANG, Xin-She; BREST, Janez. A comprehensive review of firefly algorithms. **Swarm and Evolutionary Computation**, v. 13, p. 34-46, 2013;

SURJANOVIC, S., BINGHAM, D. Virtual Library of Simulation Experiments: Test Functions and Datasets. 2013. Disponível em: <<http://www.sfu.ca/~ssurjano>>. Acesso em: 05 de Janeiro de 2016;

CARACIOLO, Marcel P. Multi-Ring: Uma nova topologia para otimização por enxame de partículas (PSO). **Departamento de Sistemas e Comunicação**, Escola Politécnica de Pernambuco. Recife-PE. 2008.