

PSO

Aplicado as funções Griewank e Ackley

Adilson Krischanski¹

¹Udesc CCT - Universidade do Estado de Santa Catarina - Campus de Ciencias Tecnologicas
Caixa Postal 89.219-710 – Joinville – SC – Brazil
Departamento de Ciencia da Computacao

{adilson.krischanski}@edu.udesc.br

Abstract. *In this experiment, we applied the Particle Swarm Optimization (PSO) algorithm to the Griewank and Ackley functions, considering unconstrained, inertia-constrained, and fully constrained problems with 10 and 20 variables. We conducted 30 experiments for each case and subsequently evaluated the performance of PSO in terms of convergence and required iterations.*

Resumo. *Neste experimento, aplicamos o algoritmo Particle Swarm Optimization (PSO) nas funções Griewank e Ackley, considerando problemas com 10 e 20 variáveis nas condicoes de sem constricao, constricao inercial e constricao total. realizamos 30 experimentos para cada caso e em seguida avaliamos o desempenho do PSO em termos de convergência e interacoes necessarias.*

1. Problemas de Otimização

Problemas de otimização são situações onde buscamos encontrar uma possível solução dentro de um vasto campo de busca. Esses problemas podem ser classificados como de maximização ou minimização de uma função objetivo, a qual está sujeita a um conjunto de restrições. O objetivo é encontrar possíveis valores para cada uma das variáveis de decisão que satisfaça o valor da função objetivo. Tais problemas podem ser lineares, não lineares, discretos, contínuos ou combinatórios. Resolver problemas de otimização torna-se desafiador devido ao alto custo computacional dado o grande espaço de busca em algumas situações.

1.1. Griewank

A função de Griewank é uma função de otimização comumente utilizada em problemas de teste e benchmarking. Ela é definida matematicamente como:

$$f(x) = 14000 \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i) + 1$$

onde $x=(x_1, x_2, \dots, x_n)$ representa um vetor de n variáveis de decisão. A função de Griewank é não convexa e possui várias mínimas locais, o que a torna um desafio para algoritmos de otimização. A primeira parcela da equação corresponde a um termo quadrático, enquanto a segunda parcela é um produto entre funções coseno que dependem do valor de cada variável dividido pela raiz quadrada do seu índice. A função Griewank tem como objetivo minimizar o seu valor, ou seja, encontrar os valores das variáveis que levam à menor resposta possível.

1.2. Ackley

A função de Ackley é uma função de otimização amplamente utilizada em problemas de teste e benchmarking. Ela foi proposta por David Ackley e possui características interessantes, como um grande número de mínimos locais, o que a torna um desafio para algoritmos de otimização. A equação que a rege é dada por:

$$f(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$$

onde $x=(x_1,x_2,\dots,x_n)$ é um vetor de n variáveis de decisão. A função Ackley combina termos que dependem da norma Euclidiana dos valores das variáveis com termos coseno. O primeiro termo na equação é uma exponencial negativa do termo da raiz quadrada da média dos quadrados das variáveis, enquanto o segundo termo é a média dos cossenos dos valores das variáveis, multiplicada por uma exponencial. A função Ackley visa minimizar seu valor, isto é, encontrar os valores das variáveis que levam ao menor resultado possível.

2. Algoritmo PSO

O algoritmo PSO (Particle Swarm Optimization) é uma técnica de otimização baseada em população que foi inspirada pelo comportamento de um bando de pássaros ou um enxame de insetos. Ele foi desenvolvido por Kennedy e Eberhart em 1995 e se tornou uma abordagem popular para resolver problemas de otimização em diversas áreas. O algoritmo trata de um conjunto de soluções, as quais são chamadas de partículas, estas por sua vez são representadas como pontos em um espaço multidimensional. Cada partícula tem uma posição e uma velocidade, que são atualizadas em cada iteração do algoritmo. A posição de uma partícula representa uma possível solução para o problema, enquanto a velocidade influencia o movimento da partícula no espaço de busca. Durante a execução do algoritmo, as partículas exploram o espaço de busca em busca da melhor solução possível (solução ótima). Elas são influenciadas pelo seu próprio melhor resultado (a melhor posição que a partícula já encontrou) e pelo melhor resultado global encontrado por qualquer partícula na população. Essas informações são usadas para ajustar a velocidade e a direção de movimento das partículas, permitindo que elas busquem regiões promissoras do espaço de busca. Uma de suas principais características é sua simplicidade e alta eficiência computacional. Ele não requer conhecimento prévio do problema ou da sua função objetivo, sendo aplicável a uma ampla gama de problemas de otimização. No entanto, o PSO pode ficar preso em atratores locais.

3. Experimento

O experimento consiste em aplicar o algoritmo PSO para minimizar a solução das funções griewank e ackley, isto é, encontrar um conjunto de valores do domínio da função que de um resultado mais próximo de 0, para a função griewank o domínio será valores $[-600,600]$, já para a função ackley o domínio será $[-32,32]$. Ambas as funções serão modeladas para serem otimizadas com 10 e com 20 dimensões, sendo cada uma das dimensões possuir o mesmo domínio.

3.1. Partícula

Cada uma das partículas armazena as seguintes informações:

- Vetor para armazenar a posição atual em cada uma das dimensões do problema

- Vetor para armazenar a velocidade em cada uma das dimensões do problemas
- Vetor para armazenar a melhor posicao que a particula ja esteve em cada uma das dimensões do problemas
- Valor do Fitness atual
- Valor do Fitness na melhor posição da particula

3.2. Gerando a Solucao Inicial

Para gerar a solucao inicial pra cada particula e sorteado um valor aleatorio para cada dimensao do problema, para garantir que o valor e realmente aleatorio e usado o clock do processador atravez da funcao srand. Isso fez-se necessario para que o valor aleatorio ficasse realmente necessario, antes do seu uso os resultados da geracao eram muito pouco dispersos

3.3. Atualizacao de Solução

A cada nova interacao as particulas sofrem uma alteracao em todas as dimensoes, essa alteracao e dada partindo de tres fatores

- Componente Inercial - Faz referencia a posicao atual da particula
- Componente Cognitiva - Faz referencia a posicao do melhor ponto de que a particula ja passou
- Componente Social - Faz referencia a posicao da melhor particula do sistema, essa componente e atualizada por parametro a cada nova iterao do sistema

Cada uma dessas componentes tem um peso, onde foram realizadas diversas analizes como:

- Manter todos em 1
- usar variacao apenas inercial
- Usar variacao para todos

Para descobrir o valor ideal da variacao a ser utilizada foram testados diversos valores de forma empirica, lembrando que os valores ideais variam de acordo com o problema proposto

3.4. Fitness - Função de Avaliação

A funcao de avalicao nesse caso foi dada com pela propria funcao objeto a ser resolvida ou seja, Funcao Griewank e Ackley.

Griewank:

$$f(x) = 14000 \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i) + 1$$

Ackley:

$$f(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$$

Em ambos os casos o objetivo é minizar a função ou seja obter um valor mais próximo do 0 (zero) absoluto.

4. Tempo e Hardware

O experimento foi conduzido em um processador AMD Ryzen 3 de 3ª geração, com 8 núcleos e 16 GB de memória RAM disponível. Devido à natureza de um código compilado a execução foi rápida, não necessando recursos de paralelismo. Para cada um dos casos foram executados 30 experimentos e os experimentos rodados em paralelo via shell scrip, o tempo medio entre eles foi de 1m38s.

5. Resultados

Dentre todos os experimentos pode-se perceber que a evolução da solução se dava no máximo entre as 10000 primeiras interações, em seguida o experimento ficava estagnado por mais de 100000 interações, para melhor visualização foram plotadas apenas as 1000 primeiras interações. Os gráficos abaixo mostram um dos experimentos de cada um dos casos

Função Griewank

- Sem fator de restrição
- com 0.7 para Inercial
- com 0.7 para Inercial e 1.4 para Cognitivo e social

Função Ackley

- Sem fator de restrição
- com 0.5 para Inercial
- com 0.5 para Inercial e 1.2 para Cognitivo e social

Tais valores foram escolhidos de forma empírica, e durante testes anteriores foram os que apresentaram melhores resultados.

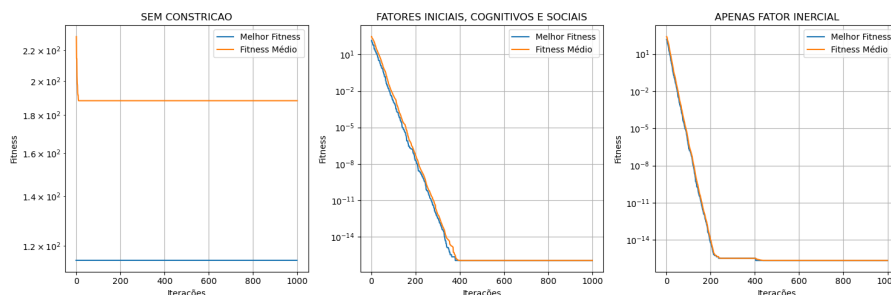


Figura 1. Função Griewank - 10 Dimensões.png

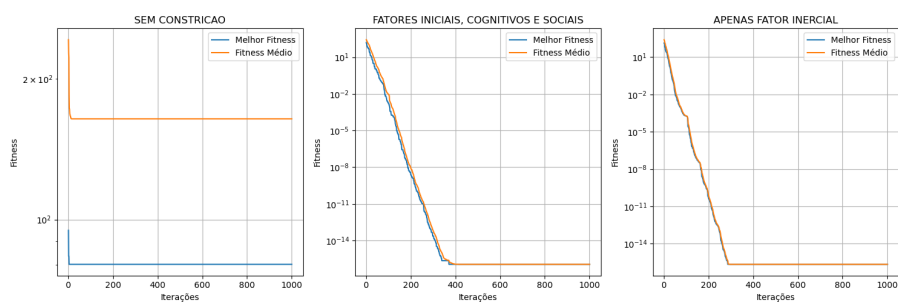


Figura 2. Funcao Griewank - 20 Dimenções.png

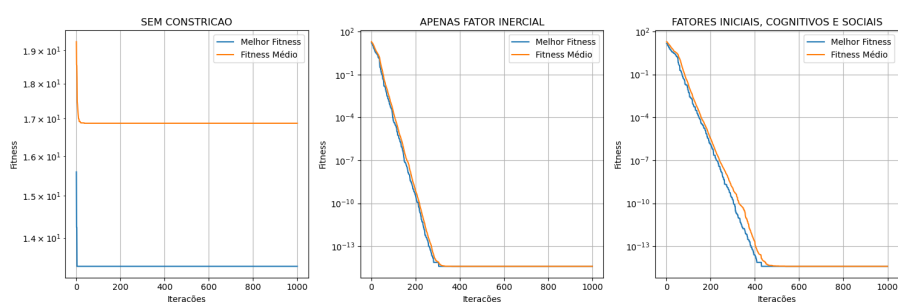


Figura 3. Funcao Ackley - 10 Dimenções.png

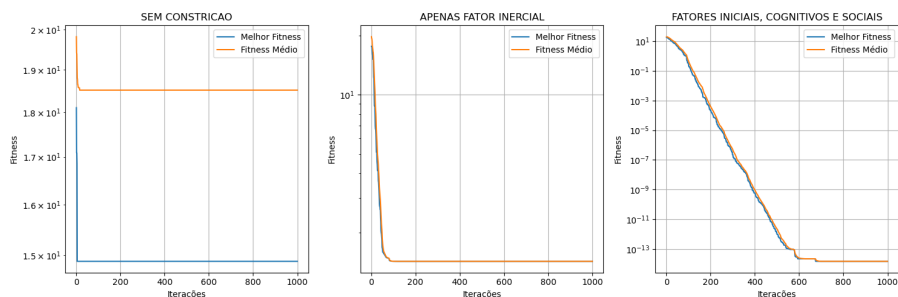


Figura 4. Funcao Ackley - 20 Dimenções.png

Como podemos observar quando nao temos nenhum fator de constricao a solucao tem dificuldade para evoluir de forma significativa, com apenas o fator de contricao inicial a solucao ja consegue evoluir de forma significativa e quando adicionado os fatores sociais e cognitivos a solucao tende a evoluir de forma estavel, todavia demora algumas iterações a mais para atingir o mesmo resultado.

Resultados Coletados

Caso	Sem Construção		Apenas Inercial		Inercial, Cognitiva e Social	
Griewank - 10D	184.4166	15.162081	1.887e-16	9.140e-17	6.661e-17	5.733e-17
Griewank - 20D	179.6589	13.341	2.109e-16	1.771e-16	6.661e-17	5.733e-17
Ackley - 10D	17.38872	0.472650	0.11551	9.140e-17	3.9968e-15	1.696e-14
Ackley - 20D	18.88392	0.21438	1.605	0.8991	0.643	0.8604

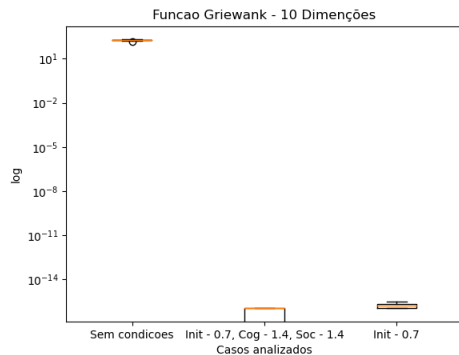


Figura 5. Função Griewank - 10 Dimensões

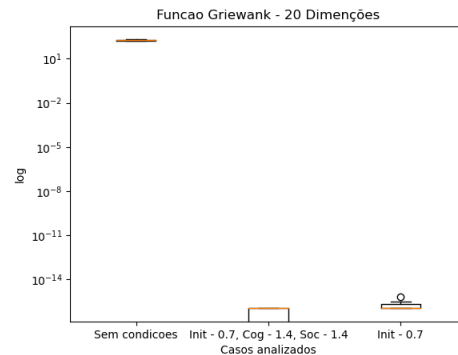


Figura 6. Função Griewank - 20 Dimensões

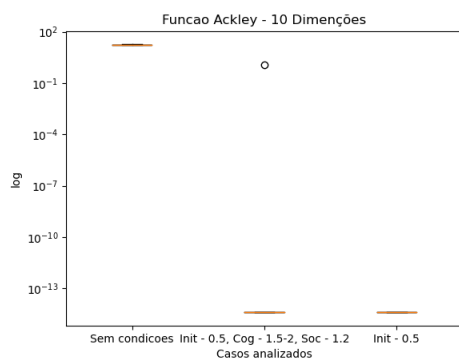


Figura 7. Função Ackley - 10 Dimensões

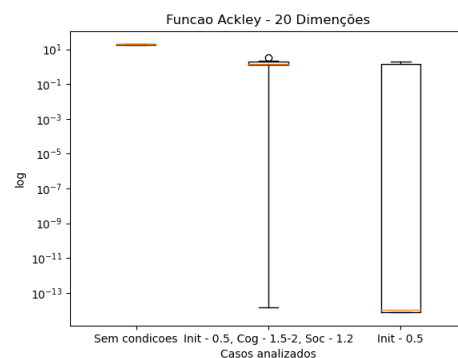


Figura 8. Função Ackley - 20 Dimensões

Como podemos observar se compararmos as funcoes o algortimo tenho um desempenho muito melhor ao buscar uma solucao para a funcao Griewank do que para a funcao Ackley, o principal fator que pode ter impactado nisso foi a escolha dos pesos que podem não ter sido a melhor opção, uma opcao seria uma variacao da funcao fitness onde os pesos possam ser alterados dinamicamente ao executar a simulacao alem de resetar as posicoes das particulas em caso de estagnacao nao ideal.

6. Dificuldades

A maior dificuldade encontrada durante o trabalho estava em conseguir um valor que fosse gerado de forma realmente aleatorio, para isso foi usada uma funcao que usa o tempo de clock do processador, oque melhorou significativamente o problema inicialmente encontrado.

7. Conclusão

Apenas de ter tido uma grande evolução em uma das problemas propostas o PSO é uma boa opção para solucionar problemas de otimização, onde baseado em conceitos de enxame mostrou capaz de divergir e apresentar uma possível solução satisfatoria para um

problema se utiliza com os parametros adequados, ou até mesmo como ja mencionado parametros que possam variar dinamicamente de acordo com o nivel de convergencia do algoritmo.

8. Referências de pesquisa

- Griewank Function. Disponível em: <<https://www.sfu.ca/ssurjano/griewank.html>>.
- Ackley Function. Disponível em: <<https://www.sfu.ca/ssurjano/ackley.html>>.
- TAM, A. A Gentle Introduction to Particle Swarm Optimization. Disponível em: <<https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/>>.