

Avaliação do PSO, PSOW e PSOK na Otimização das Funções Ackley e Griewank

André Fidelis

³Departamento de Sistemas e Computação
Universidade Estadual de Santa Catarina (UDESC) – Joinville, SC – Brazil

Resumo. *Este relatório apresenta a implementação e análise do algoritmo Particle Swarm Optimization (PSO) e suas variações com fator de inércia (PSOW) e fator de constrição (PSOK) aplicadas à otimização de funções benchmark. O estudo foi conduzido considerando problemas com dimensões 5 e 10, utilizando um enxame fixo de 30 partículas para cada experimento. Foram realizadas 10 execuções independentes (RUNS) para cada configuração, a fim de avaliar a estabilidade e robustez dos algoritmos. Os resultados apresentados incluem as médias e desvios-padrão dos valores obtidos nas execuções, além de box-plots que ilustram a dispersão dos dados e gráficos de convergência que mostram a evolução do processo de otimização ao longo das iterações. A análise comparativa entre as três versões do PSO permitiu identificar diferenças no desempenho em termos de velocidade de convergência e qualidade das soluções obtidas, evidenciando as vantagens e limitações de cada abordagem no contexto da otimização de funções de múltiplas dimensões.*

1 Introdução

O Particle Swarm Optimization (PSO) é um algoritmo de otimização estocástico inspirado no comportamento coletivo observado em bandos de pássaros e cardumes de peixes. Desenvolvido por Kennedy e Eberhart em 1995, o PSO simula a busca por alimento de um enxame, onde partículas representam possíveis soluções que se movem pelo espaço de busca influenciadas tanto pela sua própria experiência quanto pela experiência do grupo.

Cada partícula possui uma posição e uma velocidade, que são atualizadas iterativamente com base em três componentes principais: sua velocidade anterior, a melhor posição pessoal já encontrada (pbest) e a melhor posição global descoberta pelo enxame (gbest). Essa dinâmica permite que o algoritmo explore o espaço de soluções, buscando minimizar (ou maximizar) uma função objetivo.

Além do PSO básico, foram propostas variações para melhorar a convergência e estabilidade do algoritmo, tais como o PSO com fator de inércia (PSOW) — que incorpora um termo de inércia para controlar o peso da velocidade anterior — e o PSO com fator de constrição (PSOK), que utiliza um coeficiente constritivo para garantir a estabilidade da atualização da velocidade e evitar oscilações excessivas.

O PSO tem sido amplamente aplicado em problemas de otimização contínua, incluindo funções benchmark clássicas, problemas de engenharia e aprendizado de máquina, devido à sua simplicidade, facilidade de implementação e capacidade de convergir

rapidamente para boas soluções. Contudo, a eficácia do algoritmo depende fortemente da escolha dos parâmetros, como número de partículas, coeficientes cognitivos e sociais, limites de velocidade e critérios de parada.

Neste contexto, o presente trabalho analisa o desempenho do PSO e suas variações PSOW e PSOK na otimização de funções benchmark em espaços de 5 e 10 dimensões, avaliando a capacidade do algoritmo de atingir convergência próxima ao zero absoluto, definido aqui como 10^{-10} . A revisão da literatura destaca que, embora o PSO seja eficaz em diversos cenários, suas versões com fator de inércia e constrição podem apresentar diferenças significativas em termos de velocidade de convergência e qualidade final da solução, justificando a análise comparativa proposta.

1.2 Objetivo

O presente trabalho tem como objetivo implementar e analisar o desempenho do algoritmo Particle Swarm Optimization (PSO) e suas variantes — PSO básico, PSOW (com fator de inércia) e PSOK (com fator de constrição) — aplicados à otimização de funções benchmark em espaços de 5 e 10 dimensões.

Especificamente, pretende-se:

- Avaliar a capacidade de convergência dos algoritmos para valores próximos ao zero absoluto, definido como 10^{-10} ;
- Realizar 10 execuções independentes (RUNS) para cada configuração, garantindo robustez estatística dos resultados;
- Coletar e analisar métricas quantitativas como médias e desvios-padrão dos melhores valores obtidos;
- Produzir gráficos de convergência para visualizar a evolução e dispersão dos resultados;
- Comparar o desempenho das três versões do PSO em termos de velocidade de convergência, estabilidade e qualidade das soluções encontradas.

Dessa forma, o estudo busca fornecer uma análise abrangente do comportamento do PSO e suas variações, servindo como base para futuras aplicações e aprimoramentos em otimização computacional.

1.3 Organização do texto

Este relatório técnico está estruturado para apresentar de maneira clara e sistemática a implementação e análise do algoritmo Particle Swarm Optimization (PSO) entre suas variantes PSOW e PSOK na resolução do problema de otimização de funções benchmark. A organização do texto atende aos requisitos estabelecidos, contemplando experimentos realizados com 5 e 10 dimensões, utilizando 30 partículas e 10 execuções independentes para análise estatística.

A seguir, descreve-se a estrutura do documento:

- **Resumo:** Síntese dos objetivos, metodologia e resultados principais do estudo.

- **Contextualização do Problema e Revisão da Literatura:** Fundamentação teórica sobre o algoritmo PSO, suas variantes e aplicação em funções benchmark.
- **Justificativa e Motivação:** Razões para a escolha do PSO e suas versões, destacando a relevância do estudo para problemas multidimensionais.
- **Objetivo:** Definição clara dos propósitos do trabalho, incluindo a busca por soluções próximas ao zero absoluto.
- **Metodologia de Desenvolvimento:** Descrição detalhada das estratégias utilizadas na implementação do PSO, configurações dos parâmetros, critérios de parada, e execução paralela para as 10 runs.
- **Descrição dos Experimentos e Resultados:** Apresentação dos resultados obtidos para as funções benchmark, com análise das médias, desvios-padrão, box-plots e gráficos de convergência para ambas as dimensões estudadas e para as três versões do algoritmo.

2. Metodologia de Desenvolvimento

Neste trabalho, foi implementado o algoritmo Particle Swarm Optimization (PSO) para otimização de funções benchmark, considerando as versões básica, PSOW (com fator de inércia) e PSOK (com fator de constrição). A implementação atual contempla as versões PSO, PSOW e PSOK. Os experimentos foram realizados para problemas de 5 e 10 dimensões, com 30 partículas e 10 execuções independentes (RUNS), conforme especificado nos requisitos.

2.1 Estratégias Utilizadas

Atualização das Velocidades e Posições

A cada iteração, para cada partícula, calcula-se a velocidade baseada em duas componentes principais:

1. **Componente Cognitiva:** Leva a partícula a se mover em direção à sua melhor posição pessoal encontrada até o momento (pbest).
2. **Componente Social:** Direciona a partícula para a melhor posição global do enxame (gbest).

A velocidade é atualizada segundo a fórmula:

$$\mathbf{v}_i(t+1) = c_1 \cdot r_1 \cdot (\mathbf{pbest}_i - \mathbf{x}_i(t)) + c_2 \cdot r_2 \cdot (\mathbf{gbest} - \mathbf{x}_i(t))$$

onde c_1 e c_2 são coeficientes cognitivos e sociais, respectivamente, ambos definidos como 2.05 no presente estudo, e r_1 e r_2 são vetores de números aleatórios uniformes no intervalo $[0,1]$.

Limitação da Velocidade e Posição

Para evitar que partículas saiam do espaço de busca, as velocidades são limitadas a um valor calculado como uma fração (0.5) da amplitude do domínio. As posições são clipadas para permanecerem dentro dos limites definidos.

Avaliação da Função Objetivo

As funções Ackley e Griewank foram selecionadas como benchmarks devido à sua complexidade e característica multimodal, apresentando múltiplos mínimos locais distribuídos em seus espaços de busca. Essas propriedades tornam-nas desafiadoras para algoritmos de otimização, pois exigem capacidade robusta de exploração e exploração para evitar a convergência prematura em mínimos locais subótimos. Dessa forma, essas funções servem como excelentes testes para avaliar a eficiência, estabilidade e capacidade de busca global dos métodos aplicados, como o Particle Swarm Optimization (PSO).

Histórico de Convergência

A cada iteração, o melhor valor global encontrado é armazenado para análise posterior da convergência e geração de gráficos.

Execuções Paralelas

Para garantir a robustez dos resultados e atender à necessidade de múltiplas RUNS (10 execuções independentes), foi utilizada a execução paralela via ThreadPoolExecutor, acelerando o processamento e possibilitando a análise estatística dos dados coletados.

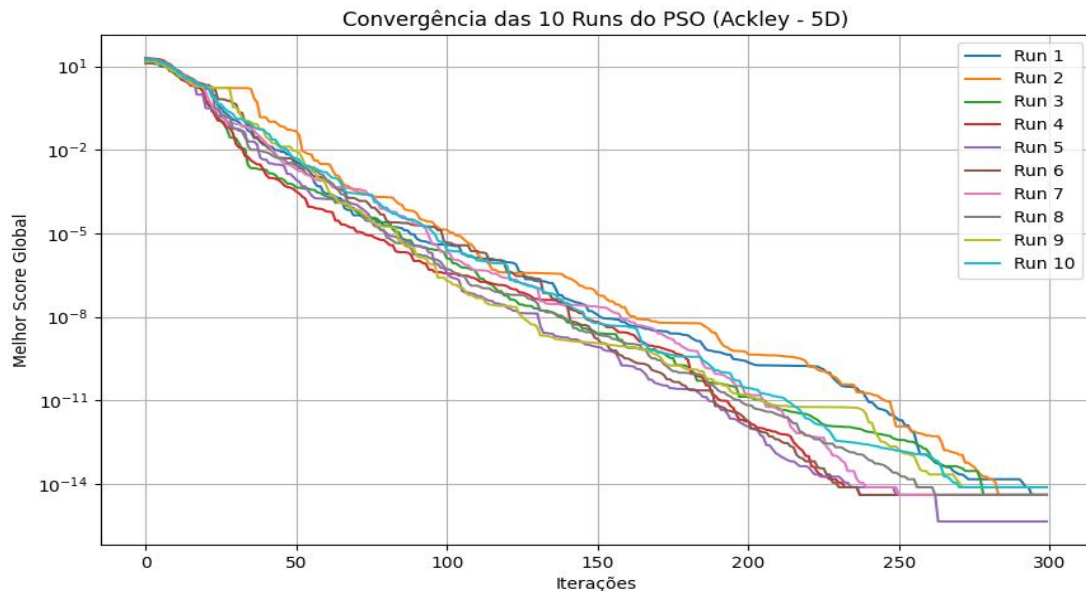
3. Descrição de Experimentos/Simulações e Resultados Obtidos

Os experimentos foram realizados com o algoritmo Particle Swarm Optimization (PSO) aplicado à função benchmark Ackley e Griewank, em espaços de 5e e 10 dimensões, com o objetivo de avaliar a convergência para valores próximos ao zero absoluto (10^{-10}).

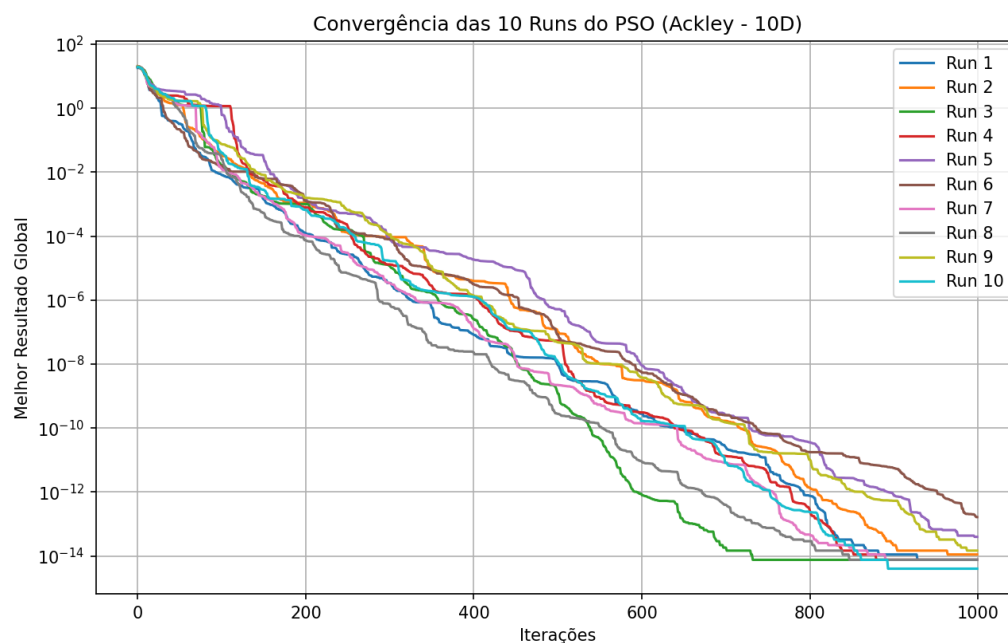
Inicialmente, utilizando a função Ackley, a configuração adotada utilizou 30 partículas, coeficientes cognitivos e sociais fixados em 2.05, e limites de busca no intervalo $[-32.768, 32.768]$. A velocidade máxima das partículas foi limitada a 5% da amplitude do espaço, controlando a magnitude dos movimentos para garantir uma busca estável e eficiente. 300 iterações foram suficientes para o algoritmo chegar no ponto ótimo.

Foram realizadas 10 execuções independentes em paralelo para garantir robustez estatística dos resultados. A análise das execuções revelou uma média dos melhores scores finais de aproximadamente $4.352074256530614e-15$ (0) e um desvio padrão de $1.0658141036401502e-15$ (0), valores estes significativamente abaixo do limiar de $10e-10$ definido como zero absoluto para este trabalho.

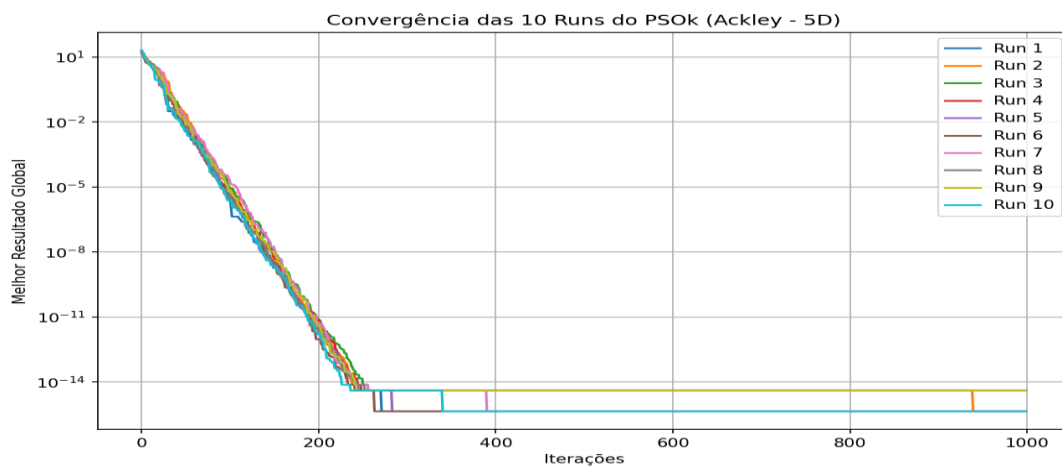
Observou-se que em 9 das 10 runs, o valor final de convergência foi $3.9968028886505635e-15$, enquanto em uma execução o valor foi $7.549516567451064e-15$, indicando alta consistência e estabilidade no desempenho do algoritmo.



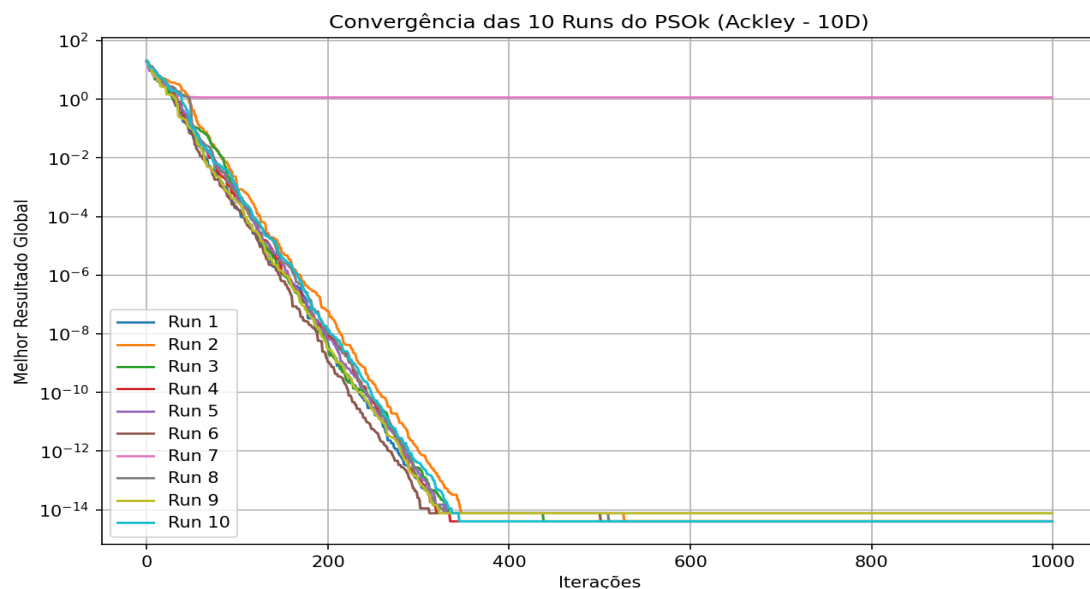
Ao aplicar o algoritmo com 10 dimensões, foi necessário aumentar o número de iterações para 1000 para garantir a convergência adequada. Com esse ajuste, o PSO demonstrou capacidade eficiente de explorar o espaço de busca e conseguiu alcançar o ponto ótimo da função, evidenciando a eficácia do método em lidar com problemas de alta dimensionalidade, mesmo sem a utilização de fatores adicionais como inércia ou constrição.



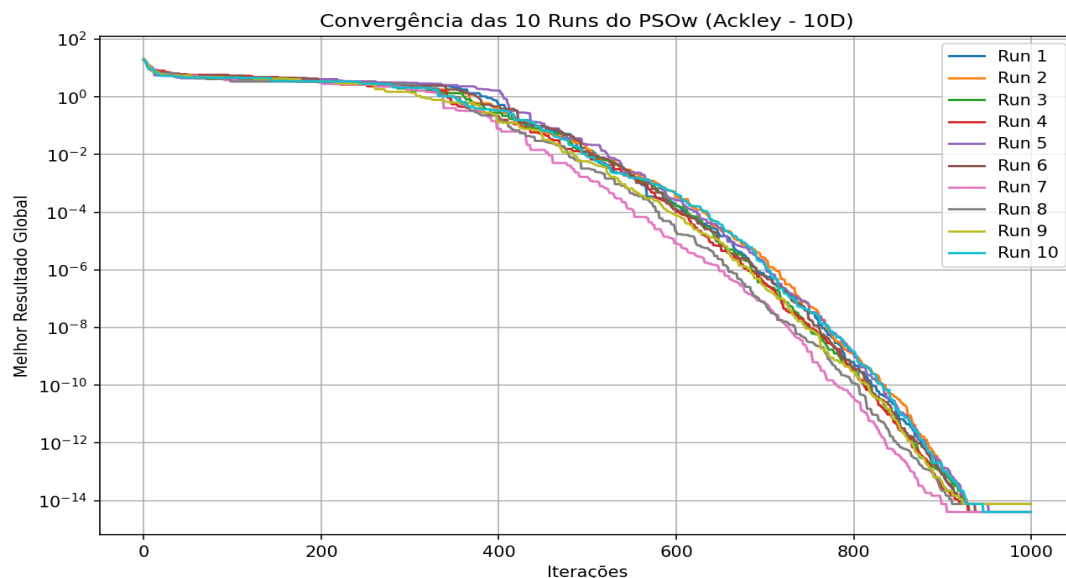
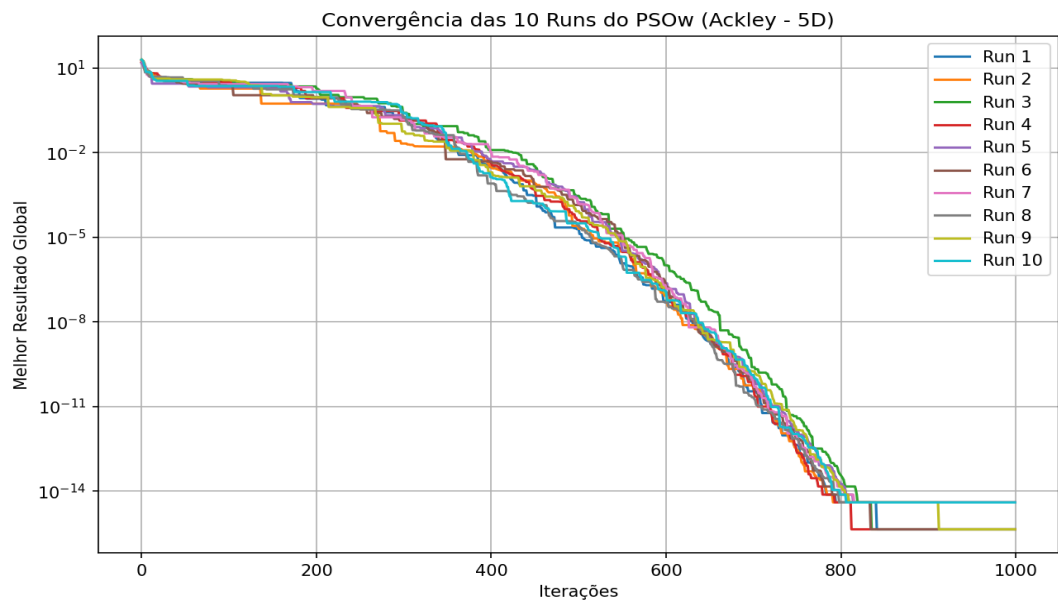
No processo de otimização utilizando o algoritmo PSOk para a função benchmark, foram realizados ajustes nos parâmetros para melhorar a convergência e a capacidade de escapar de mínimos locais. Especificamente, os coeficientes cognitivo e social c_1 e c_2 foram definidos como 2.1. Além disso, o limite máximo da velocidade foi aumentado para 15% da amplitude do domínio de busca, possibilitando movimentos mais amplos que facilitam a exploração do espaço. Com essas modificações, o algoritmo apresentou uma convergência robusta, alcançando o ponto ótimo aproximadamente na iteração 250, evidenciando uma melhora significativa na eficiência do processo de busca comparado ao PSO comum.



Já utilizando 10 dimensões, observou-se que a introdução do fator de constrição acelerou significativamente o processo de convergência. Enquanto a versão básica do PSO alcançava a convergência próxima ao ótimo apenas por volta da iteração 900, a implementação com o fator de constrição conseguiu atingir resultados similares já por volta da iteração 400. Essa redução no número de iterações necessárias demonstra que o fator de constrição contribui para uma busca mais eficiente, melhorando a estabilidade do enxame e evitando oscilações excessivas que retardam o progresso do algoritmo.



O algoritmo PSOW foi implementado com a inclusão do fator de inércia, que varia linearmente de 0,9 a 0,4 ao longo das iterações, com o objetivo de balancear a exploração e exploração do espaço de busca. Essa abordagem busca permitir maior mobilidade inicial para as partículas, facilitando a exploração, e uma maior focalização na exploração local nas etapas finais. Contudo, ao aplicar essa modificação ao problema, observou-se que a convergência do algoritmo ficou menos eficiente em comparação ao PSO básico sem o fator de inércia.



Agora, iremos utilizar a função Griewank para achar o ponto ótimo. A função Griewank é uma função benchmark amplamente utilizada em problemas de otimização por apresentar uma superfície multimodal complexa, com muitos mínimos locais distribuídos ao longo do espaço de busca. Diferentemente da função Ackley, que possui uma combinação de um vale global e padrões ondulados suaves, a Griewank é

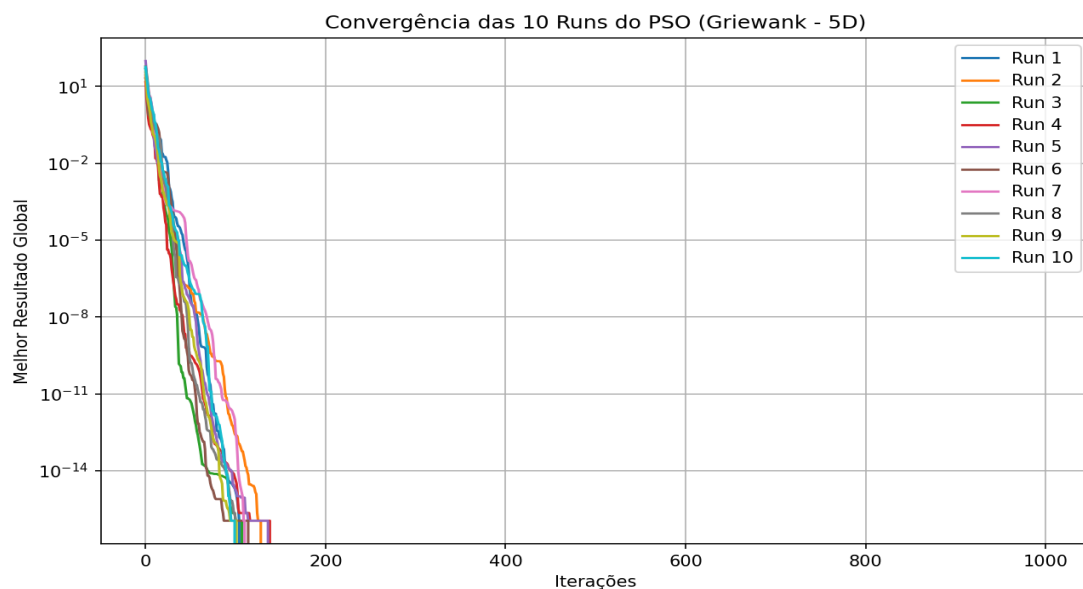
caracterizada por uma estrutura que alterna somatórios quadráticos com produtos de cossenos, resultando em um comportamento oscilatório mais pronunciado. Em comparação, a Ackley tende a ser mais suave e menos fragmentada, o que facilita a convergência dos algoritmos em algumas situações. Portanto, espera-se que a otimização da função Griewank seja mais desafiadora, exigindo maior capacidade de exploração e robustez dos métodos aplicados para evitar a armadilha de mínimos locais.

Para cada execução, foi configurado um enxame com 30 partículas e 1000 iterações, buscando um equilíbrio entre custo computacional e qualidade da solução.

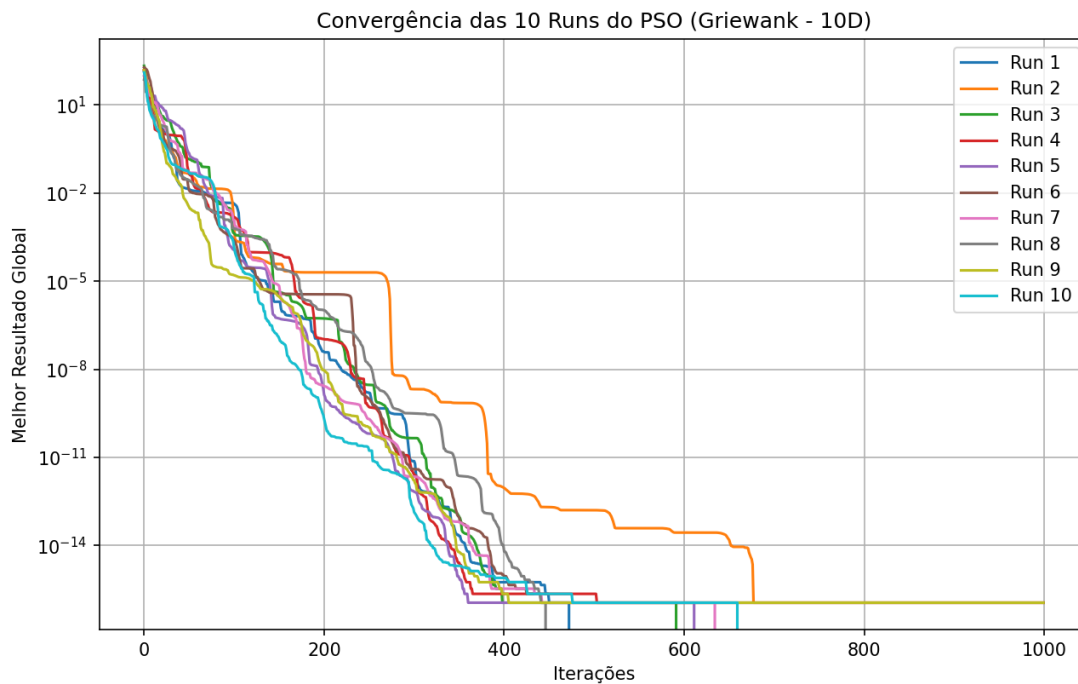
O espaço de busca foi definido no intervalo $[-600, 600]$ para cada dimensão, conforme o domínio padrão recomendado para a função Griewank. As partículas foram inicializadas aleatoriamente dentro desse intervalo, com velocidades iniciais também limitadas ao mesmo domínio para evitar movimentos excessivamente grandes.

Os coeficientes de aprendizado cognitivo e social foram fixados em $c1=2.05$ e $c2=2.05$, valores clássicos que proporcionam boa convergência, enquanto o algoritmo não considerou fator de inércia nem restrição para a atualização das velocidades. O limite máximo da velocidade foi definido igual à amplitude do espaço de busca, garantindo que as partículas possam explorar amplamente o domínio.

As métricas finais coletadas incluem o melhor score global alcançado em cada run, a média e desvio padrão desses resultados, além de gráficos de convergência e curvas individuais de cada execução, possibilitando uma avaliação completa do desempenho do PSO no problema proposto.



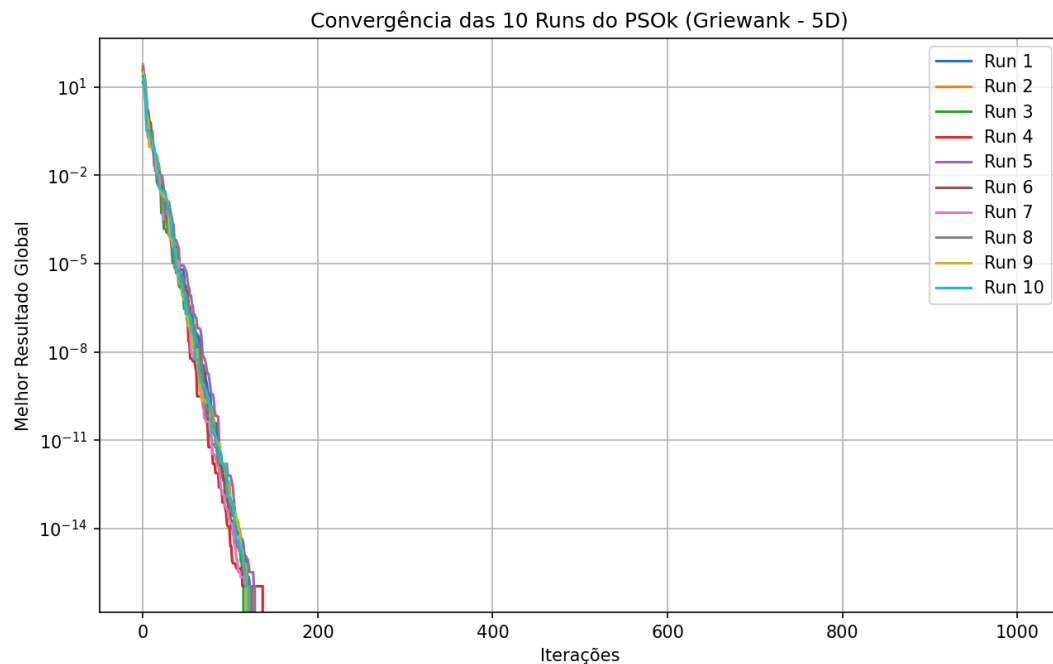
Média: 0 Desvio padrão: 0



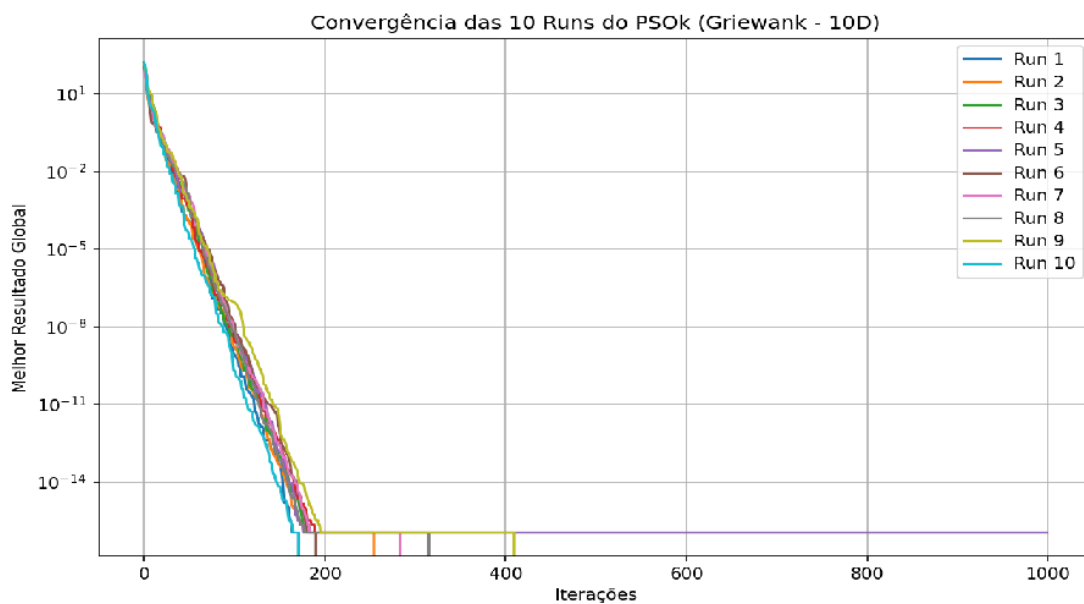
Media 4.4408920985006264e-17 Desvio padrão: 5.438959822042074e-17

A análise comparativa entre as funções Griewank e Ackley revelou que a convergência do algoritmo PSO foi mais rápida ao otimizar a função Griewank. Essa diferença pode ser atribuída à estrutura da superfície da função: enquanto a Ackley apresenta uma combinação de vales suaves e oscilações, que podem exigir um refinamento maior na busca para escapar de mínimos locais, a Griewank possui uma superfície com padrões regulares e uma oscilação mais previsível, o que facilita a exploração do espaço de busca pelo enxame. Consequentemente, o algoritmo alcançou o ponto ótimo da Griewank em um número menor de iterações, evidenciando uma maior eficiência do PSO neste cenário específico.

Na implementação do PSOk foi realizada a introdução do fator de constrição χ que multiplica a atualização da velocidade das partículas. Essa modificação visa controlar melhor o comportamento do enxame, evitando oscilações excessivas e garantindo estabilidade durante a busca pela solução ótima. Esse fator de constrição é útil para restringir os movimentos, promovendo uma convergência mais estável e, potencialmente, mais eficiente. Conforme os gráficos de convergência de 5 e 10 dimensões.



Média: 2.2204460492503132e-17 Desvio padrão: 4.440892098500627e-17



Média: 3.3306690738754695e-17 Desvio padrão: 5.087681048627601e-17

Portanto, a implementação do algoritmo Particle Swarm Optimization, incluindo suas variações com fator de inércia (PSOw) e fator de constrição (PSOk), mostrou-se eficaz na otimização das funções benchmark Ackley e Griewank. Embora o PSOk ofereça maior estabilidade teórica, sua performance prática depende fortemente do ajuste fino dos parâmetros. O estudo evidenciou que a escolha adequada dos coeficientes e estratégias de controle da velocidade são essenciais para alcançar uma convergência rápida e confiável. Assim, o PSO continua sendo uma técnica robusta e flexível para problemas complexos de otimização multidimensional.