

# Inteligência de Enxames para Funções Complexas

Gustavo Silveira Dias  
Bacharelado em Engenharia de Computação  
Instituto Federal de Minas Gerais  
Inteligência Artificial  
Prof. Dr. Cíniro Nametala

## SUMÁRIO

<b>I</b>	<b>Introdução</b>	1
<b>II</b>	<b>Definição do problema</b>	1
<b>III</b>	<b>Metodologia</b>	1
III-A	Principais modificações . . . . .	1
III-B	Parametrização . . . . .	1
<b>IV</b>	<b>Resultados</b>	2
IV-A	Análise do melhor caso . . . . .	2
IV-B	Análise de experimentos . . . . .	2
<b>V</b>	<b>Conclusões</b>	2
	<b>Referências</b>	2

## LISTA DE FIGURAS

1	Fitness do Gbest a cada iteração (melhor caso) . . . . .	2
2	Fitness do Média Gbest a cada iteração (melhor caso) . . . . .	2

# Inteligência de Enxames para Funções Complexas

**Resumo**—Neste relatório técnico são discutidos detalhes de implementação, testes e resultados obtidos com o algoritmo Particle Swarm Optimization (PSO) aplicado a Funções Complexas. Buscando-se balancear parâmetros foram realizadas avaliações empíricas utilizando uma função rastrigin. Após definição de parâmetros, o algoritmo balanceado foi executado 20 vezes.

## I. INTRODUÇÃO

O algoritmo Particle Swarm Optimization (PSO), em português, Otimização por Enxame de Partículas, é uma técnica e otimização computacional inspirada no comportamento biológico de pássaros e peixes. Desenvolvido por Dr. James Kennedy e Dr. Russel Eberhart em 1995, o PSO é uma abordagem heurística que visa encontrar soluções eficientes para problemas complexos de otimização.

A inspiração para o PSO provém da observação do comportamento de enxames na natureza, onde os indivíduos do grupo, chamados de partículas, colaboram de forma cooperativa para atingir um objetivo comum. Cada partícula no PSO representa uma possível solução para o problema abordado e move-se através do espaço de busca, ajustando sua posição com base no desempenho das soluções.

O funcionamento do PSO é guiado por dois principais componentes matemáticos: a posição atual da partícula e sua velocidade. A posição representa a solução candidata no espaço de busca, enquanto a velocidade indica a direção e a magnitude do movimento da partícula. A atualização da posição e velocidade é realizada a cada iteração, com base em fatores que incluem o desempenho passado da partícula e do seu grupo.

O PSO demonstra eficácia em várias aplicações, como otimização de funções matemáticas que é o caso desse trabalho. Sua abordagem intuitiva e capacidade de explorar espaços de busca complexos fazem do PSO uma ferramenta valiosa para lidar com problemas de otimização em várias áreas.

## II. DEFINIÇÃO DO PROBLEMA

O problema de otimização proposto envolve a minimização da função Rastrigin, sujeita a restrições de igualdade e desigualdade, com variáveis de decisão restritas ao intervalo  $[-5.12, 5.12]$ . A função objetivo  $f(x)$  combina termos quadráticos e cossenos, tornando o espaço de busca desafiador. As restrições  $g(x)$  e  $h(x)$  introduzem complexidades adicionais, com  $g(x)$  sendo uma restrição de desigualdade relacionada a um mapeamento no círculo trigonométrico. A solução ótima, representada por  $x^* = -1/3, -1/3, -1/3, \dots, -1/3$ , requer a satisfação simultânea de ambas restrições. A abordagem do Otimização por Enxame de Partículas será crucial para explorar eficientemente esse espaço multidimensional, ajustando as posições das partículas a cada iteração. O PSO buscará soluções válidas dentro do intervalo especificado, causando penalidades para

valores fora desse intervalo. A complexidade desse problema e a necessidade de considerar simultaneamente restrições de igualdade e desigualdade destacam a importância do PSO na busca por uma solução ótima.

## III. METODOLOGIA

O Otimização por Enxame de Partículas implementado para a avaliação nesse relatório seguiu os mesmos passos propostos e já comentados na seção de Introdução, contudo, algumas estratégias foram adicionadas buscando-se atingir melhorias na resolução do problema proposto. As principais são descritas a seguir.

### A. Principais modificações

Após a atualização da posição de cada partícula, um passo adicional foi implementado para garantir que as soluções mantenham proximidade com o valor alvo de 0.33 ou -0.33. Para cada dimensão de cada partícula, verificamos se a solução está suficientemente próxima de 0.33 ou -0.33. Se a solução não atender a essa condição de proximidade, a posição da partícula para essa dimensão é aleatoriamente redefinida dentro do intervalo permitido. Esse procedimento ajuda a manter as soluções próximas ao valor desejado ao longo das iterações do algoritmo PSO.

A variável limite proximidade controla a proximidade aceitável em relação aos valores alvo. Ajustar esse valor permite definir a tolerância desejada para aceitar ou descartar soluções com base na proximidade com 0.33 ou -0.33. Essa lógica visa direcionar o algoritmo para soluções que atendam aos critérios específicos do problema, reforçando a convergência para o resultado desejado.

### B. Parametrização

Levando-se em conta que a versão do algoritmo utilizada foi exatamente a clássica, foram realizados testes empíricos variando-se, vez a vez, cada um dos parâmetros para se observar comportamentos como o vetor  $X^*$  e a qualidade da Fitness encontrada. Após diversos testes, a parametrização utilizada foi a seguinte:

- Constante de Inércia (W): 0.7
- Ponderação do Indivíduo (C1): 1.3
- Ponderação Global (C2): 1.7
- Dimensões (N): 10
- Iterações: 250
- Partículas: 4000
- Limite Inferior: -5.12
- Limite Superior: 5.12

Foram realizados, com estas configurações, um total de 20 execuções. Vale lembrar que os testes foram realizados com 10 dimensões. Os resultados mostrados na seção a seguir dizem a respeito a um teste específico (melhor caso).

## IV. RESULTADOS

### A. Análise do melhor caso

Tornando-se por base um dos experimentos realizados (melhor caso) pode-se observar nas Figuras 1 e 2, respectivamente, a convergência da melhor solução e média, iteração a iteração.

O vetor que apresentou os melhores valores é composto da seguinte sequência  $x^* = [0.3236 \ -0.2717 \ 0.3041 \ -0.3161 \ -0.3447 \ -0.2868 \ 0.3871 \ 0.3308 \ -0.3207 \ 0.2833]$ . Como pode-se perceber o algoritmo encontrou valores próximo ao ótimo.

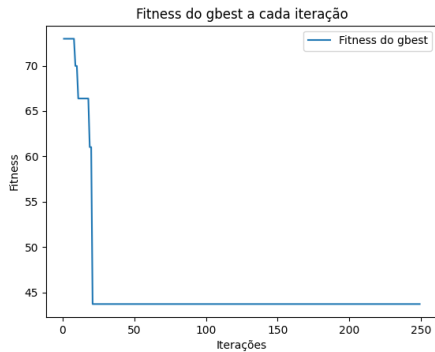


Figura 1: Fitness do Gbest a cada iteração (melhor caso)

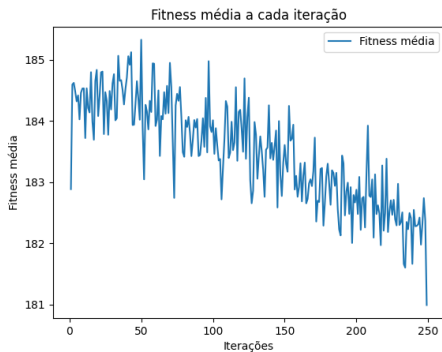


Figura 2: Fitness do Média Gbest a cada iteração (melhor caso)

### B. Análise de experimentos

Vale ressaltar que na apresentação em sala de aula o código apresentava valores distantes ao ótimo. Após a apresentação me dediquei em resolver esse problema para entregar da melhor forma que conseguiria. A decisão que resultou em uma performance melhor do algoritmo foi descrita na seção de Principais Modificações.

Em resumo, o aumento no número de dimensões em problemas de otimização implica em desafios consideráveis, incluindo a complexidade computacional, dificuldades em exploração e exploração, e a necessidade de estratégias específicas para lidar com o espaço de busca expandido. A seleção cuidadosa de métodos e abordagens torna-se crucial

para o sucesso na resolução de problemas em espaços de alta dimensão.

Por fim, utilizando-se a linguagem de programação Python3 e realizando-se experimentos sob um processador Intel i5 de 10ª geração com 2.1GHz.

## V. CONCLUSÕES

Neste relatório foi apresentado uma análise do algoritmo Otimização por Enxame de Partículas quando submetido a Funções Complexas.

Os principais resultados observados deixaram claro que o algoritmo é capaz de identificar pontos ótimos, no entanto, na sua versão clássica o algoritmo demonstrou grande dificuldade em explorar soluções de qualidade.

## REFERÊNCIAS

- [1] E. Bonabeau, M. Dorigo et G. Theraulaz (1999). *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press. ISBN 0-19-513159-2
- [2] Kennedy, J. e Eberhart, R. C. (1995). *Particle Swarm Optimization*. IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942-1948.